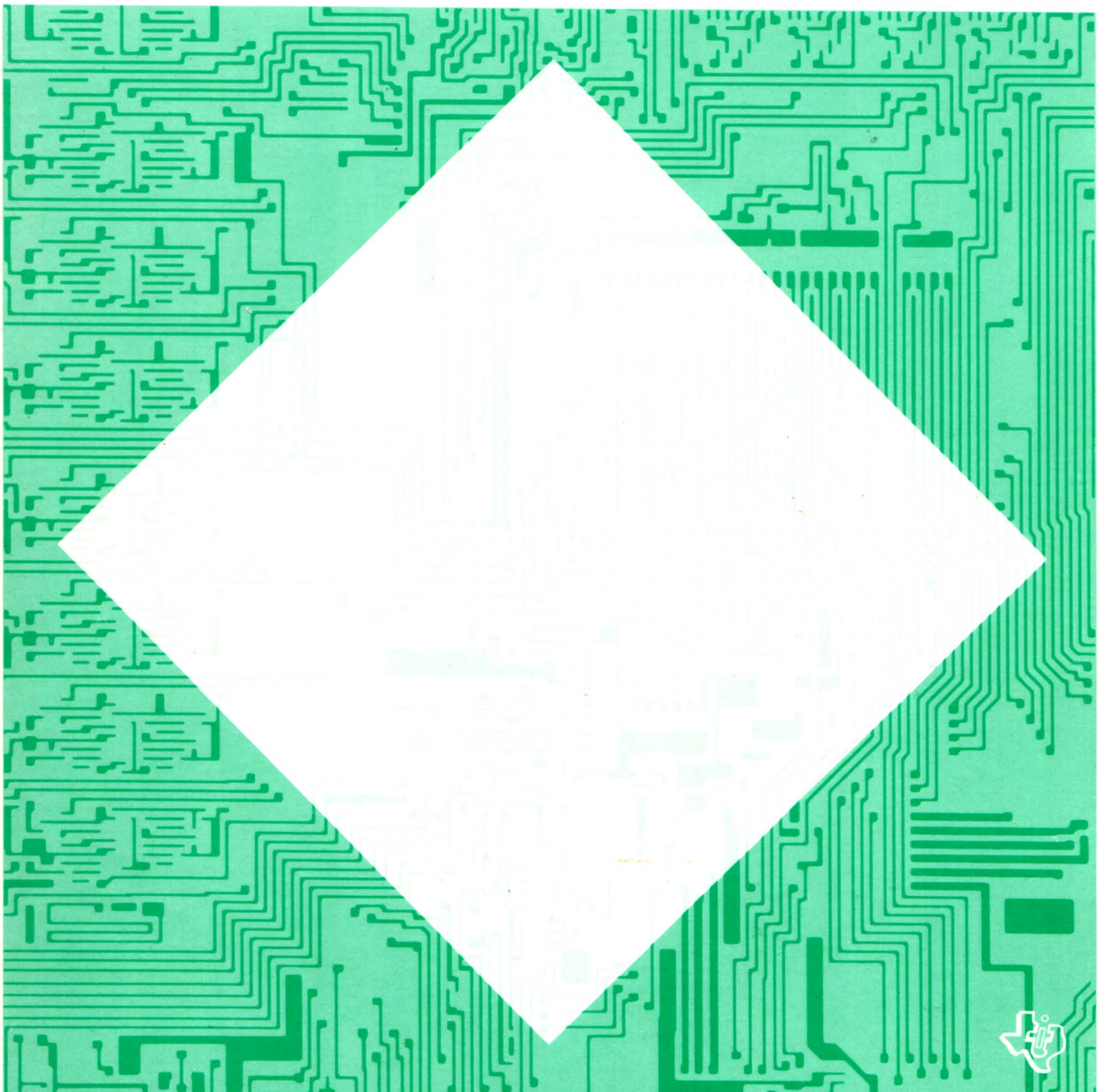
TI Programmable 58/59

Electrical Engineering

Using the power of your *Solid State Software*™ module



Copyright © 1979, Texas Instruments Incorporated

TABLE OF CONTENTS

*Printer required.
**Printer and TI-59 required.

INTRODUCTION

The Electrical Engineering Library contains programming aids for the design engineer, systems engineer, and engineering technical assistant. From computing component values of filters and other electrical circuits to numerical aids for analysis and design of feedback control systems, parameter conversions to transmission line calculations, signal detection to time series transformations, the programs cover a broad range of electrical engineering and systems engineering applications.

USING THIS MANUAL

Following this brief introduction, you will find the description, user instructions, and example problems for each of the 19 programs in the Electrical Engineering Library. Each program is easily identified by the "EE" number in the upper corner of the page. This number corresponds with the call number you use to tell the calculator which program in the *Solid State Software** module you wish to use.

The primary reference point in this manual for each program is the User Instructions. These user instructions are also available for you in the handy pocket guide furnished with the library. The program description and sample problems should be used when you first run a program, to help you understand its full capabilities and limitations. Nonmagnetic label cards to identify the user-defined keys are also included in the library. Carefully remove the cards from the sheet and insert them in the card carrying case for convenient storage. Note that a special holder has been built into the case for storage of the library module.

When using the *Solid State Software* programs as subroutines to your own programs, you will also want to check Register Contents for the program and check Program Reference Data provided in Appendix A.

USING THE OPTIONAL PRINTER

If you have the optional PC-100A or PC-100C printer†, a printed record of entries and results is automatic. The User Instructions and example problems are marked to show exactly which values are printed in addition to being displayed.

Use the Calculator Mounting procedure in the PC-100A or PC-100C Owner's Manual to mount your calculator on the printer. The switch called out in Step 2 of the PC-100A manual should be set to "OTHER" for your calculator. Always turn the calculator and printer off before mounting or removing the calculator.

TIPS FOR RUNNING PROGRAMS

Before you begin using the *Solid State Software* programs on your own, here are a few things to keep clearly in mind until you become familiar with your calculator.

*Trademark of Texas Instruments

†Note: The TI Programmable 58 and TI Programmable 59 will not operate on the PC-100 print cradle.

INTRODUCTION

1. Press [CLR] before running a program if you are not sure of the status of the calculator. (To be completely sure of calculator status, turn it off and on again — but remember that this will clear the program memory.)

2. Some programs may leave the calculator in fix-decimal format (see Appendix A). In that event, you should press [INV] [2nd] [fix] before running another program if this format is not desired.

3. There is no visual indication of which *Solid State Software* program has been called. If you have any doubts, the safest method is to call the desired program with [2nd] [Pgm] mm, where mm is the two-digit program number. The calculator will remain at this program number until another program is called, [RST] is pressed or the calculator is turned off.

4. A flashing display normally indicates an improper key sequence or that a numerical limit has been exceeded. When this occurs, always repeat the program sequence and check that each step is performed as directed by the User Instructions. Any unusual limits of a program are given in the User Instructions or related notes. The In Case of Difficulty portion of Appendix A in *Personal Programming* may be helpful in isolating a problem.

5. Some of the *Solid State Software* programs may run for several minutes depending on input data. If you desire to halt a running program, press the [RST] key. This is considered as an emergency halt operation which returns control to the main memory. A program must be recalled to be run again.

USING *SOLID STATE SOFTWARE* PROGRAMS AS SUBROUTINES

Any of the *Solid State Software* programs may be called as a subroutine to your own program in the main memory. Either of two program sequences may be used: 1) [2nd] [Pgm] mm (User-Defined Key) or 2) [2nd] [Pgm] mm [SBR] (Common Label). Both will send the program control to program mm, run the subroutine sequence, and then automatically return to the main program without interruption. Following [2nd] [Pgm] mm with anything other than [SBR] or a user-defined key is not a valid key sequence and can cause unwanted results.

It is very important to consider the Program Reference Data in Appendix A for any program called as a subroutine. You must plan and write your own program such that the data registers, flags, subroutine levels, parentheses levels, T-register, angular mode, etc., used by the called subroutine are allowed for in your program. In addition, a Register Contents section of each program description provides a guide to determine where data is or must be located to run the program. A sample program that calls a *Solid State Software* program as a subroutine is provided in the PROGRAMMING CONSIDERATIONS section of *Personal Programming.*

If you need to examine and study the content of a *Solid State Software* program, you can download as described in the following paragraphs.

DOWNLOADING *SOLID STATE SOFTWARE* PROGRAMS

If you need to examine a *Solid State Software* program, it can be downloaded into the main program memory.* This will allow you to single step through a program in or out of the learn mode. It also allows using the program list or trace features of the optional printer. The only

*Unless the library is a protected special-purpose library.

requirement for downloading a *Solid State Software* program is that the memory partition be set so there is sufficient space in the main program memory to receive the downloaded program. The key sequence to download a program is [2nd] [CP] [2nd] [Pgm] mm [2nd] [Op] 09 where mm is the program number to be downloaded. This procedure places the requested program into program memory beginning at program location 000. The downloaded program writes over any instructions previously stored in that part of program memory. Remember to press [RST] before running or tracing the downloaded program.

All of the EE programs can be downloaded in the TI Programmable 59 using normal power-up partitioning.

The partition must be changed from the power-up condition in the TI Programmable 58 for programs EE-02, 03, 06, 08, 09, 13, 14, 16, 17, and 19. The key sequence to repartition the main memory for EE-03, 06, 13, and 17 is **2** [2nd] [Op] **17**. The sequence for EE-02, 14, 16, and 19 is **1** [2nd] [Op] **17**. The sequence for EE-08 and 09 is **0** [2nd] [Op] **17**.

REMOVING AND INSTALLING MODULES

The Electrical Engineering Module can easily be installed in the calculator or replaced with another. It is a good idea to leave the module in place in the calculator except when replacing it with another module. Be sure to follow these instructions when you need to remove or replace a module.

CAUTION

Be sure to touch some metal object before handling a module to prevent possible damage by static electricity.

1. **Turn the calculator OFF.** Loading or unloading the module with the calculator ON may cause the keyboard or display to lock out. Also, shorting the contacts can damage the module or calculator.

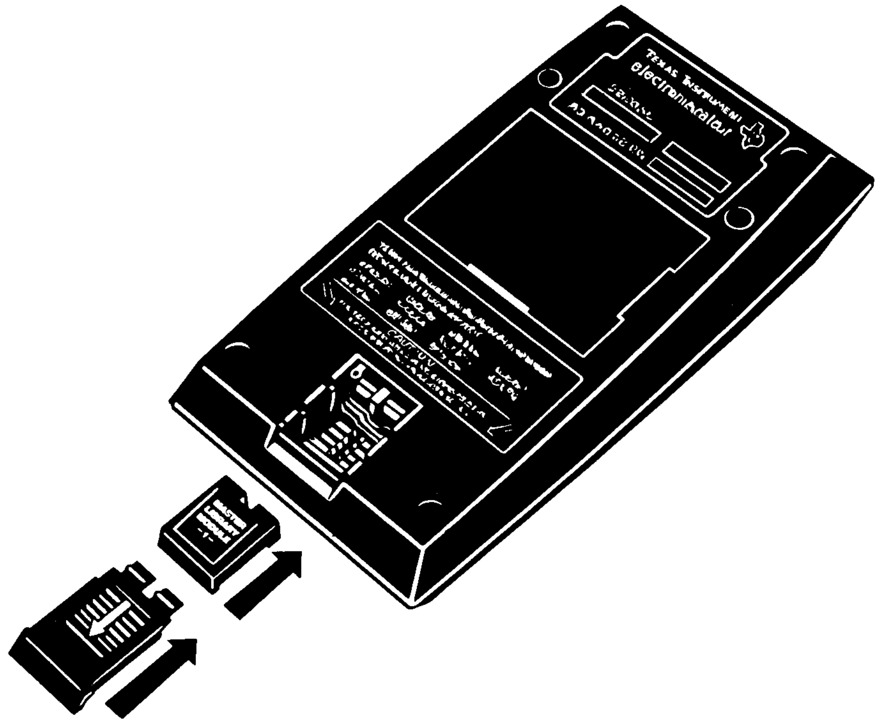
2. Slide out the small panel covering the module compartment at the bottom of the back of the calculator. (See Diagram on following page.)

3. Remove the module. You may turn the calculator over and let the module fall out into your hand.

4. Insert the module, notched end first with the labeled side up into the compartment. The module should slip into place effortlessly.

5. Replace the cover panel, securing the module against the contacts.

INTRODUCTION



Don't touch the contacts inside the module compartment as damage can result.

ELECTRICAL ENGINEERING MODULE CHECK

This program performs the following functions separately.

1. Library Module Check
2. Linear Regression Initialization

Library Module Check

When you want to know which of your *Solid State Software* modules is in the calculator without physically looking at it, you can call the Library Module check portion of the routine. If the Electrical Engineering Module is in the calculator, the number 11. will be displayed. This number is unique to the Electrical Engineering Library (other optional libraries use other identifying digits). If the calculator is attached to a PC-100A or PC-100C print cradle, the following will be printed:

```
EE
 11.
```

If these results are not obtained, there is probably a malfunction in the calculator or the *Solid State Software* module. First, install the Master Library module and try running the diagnostic program, ML-01. If this program does not run properly, refer to Appendix A of *Personal Programming* for an explanation of the various procedures to be followed when you have difficulties.

Linear Regression Initialization

This routine initializes the calculator for linear regression by clearing data registers R_{01} through R_{06} and the T-register. It should be used whenever linear regression or other built-in statistics functions are to be started. You can also use the routine at any time to clear these registers selectively without disturbing any other registers.

Solid State Software TI © 1979

ELECTRICAL ENGINEERING MODULE CHECK EE-01

MODULE CHECK: [SBR] [2nd] [R/S]

LINEAR REGRESSION INITIALIZATION: [SBR] [CLR]

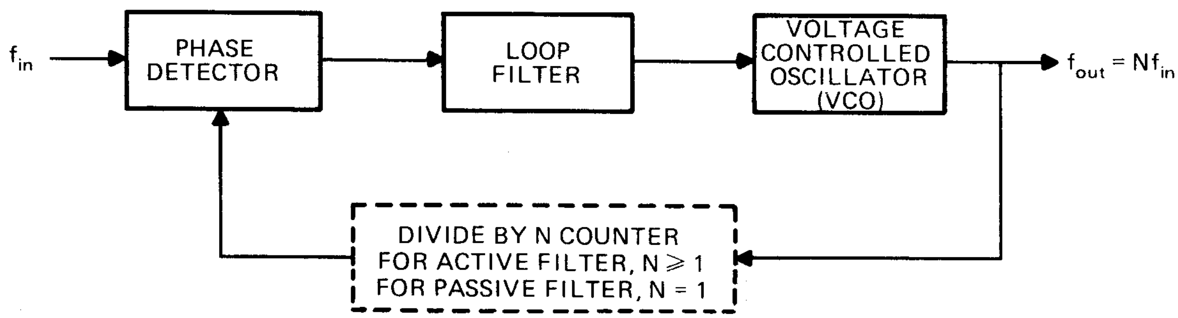
USER INSTRUCTIONS

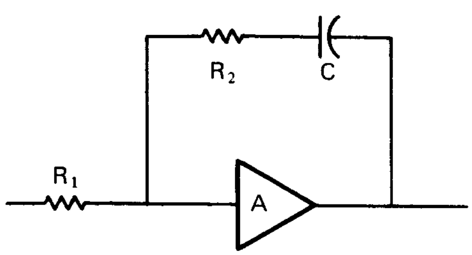
STEP	PROCEDURE	ENTER	PRESS	DISPLAY
	Library Module Check			
A1	Select Program		[2nd] [Pgm] 01	
A2	Run Module Check		[SBR] [2nd] [R/S]	11.[1]
	Initialize Linear Regression			
B1	Select Program		[2nd] [Pgm] 01	
B2	Initialize Linear Regression		[SBR] [CLR]	0.

NOTE: 1. The number 11. indicates the Electrical Engineering Library.

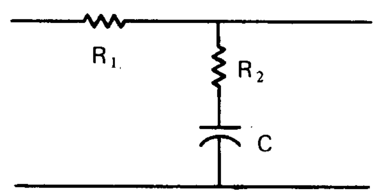
PHASE-LOCKED LOOP

Given the choice of an appropriate loop filter (active or passive), this program computes the resulting design parameters for a basic phase-locked loop (PLL) as illustrated below. A type-2, second-order loop is realized with an active filter while a type-1, second-order loop results with a passive filter.





ACTIVE FILTER



PASSIVE FILTER

The basic PLL transfer function is $\frac{\theta_o}{\theta_i}(S) = \frac{G\,F(S)}{S + G\,F(S)}$

where

- θ_i is the input phase
- θ_o is the output phase
- G is the loop gain
- $F(S)$ is the transfer function of the loop filter.

$$G = K_p K_v$$

where

- K_p is the gain of the phase detector in volts/radian
- K_v is the gain of the VCO in radians/(second volt)

If the loop filter is active, a (÷N) counter can be added to the loop, making the output frequency, f_{out}, an integral multiple of the input or reference frequency, f_{in}. In this case, N is an integer divisor ⩾ 1 and $f_{out} = N\,f_{in}$. This technique is used for multiple frequency generation in frequency synthesizers. If the loop filter is passive, the program ignores the (÷N) counter and in effect N = 1 making $f_{out} = f_{in}$.

In this program, the user must input the loop gain G and a value for C in farads. If the loop filter chosen is active, then N is also required. At this point, the user has the option of either inputting ω_n, the natural angular frequency in radians/second and ζ, the damping factor and solving for the resistances R_1 and R_2 in ohms, or vice versa. With a passive loop filter, a test is made before computing R_1 and R_2 to insure that the resistance values are greater than zero. Finally, after completing the above computations, the loop-noise bandwidth can be computed.

Solid State Software				TI © 1979
PHASE LOCKED LOOP				EE-02
N	R_1	R_2	→ B	Execute (PF)
G	ω_n	C	ζ	Execute (AF)

USER INSTRUCTIONS

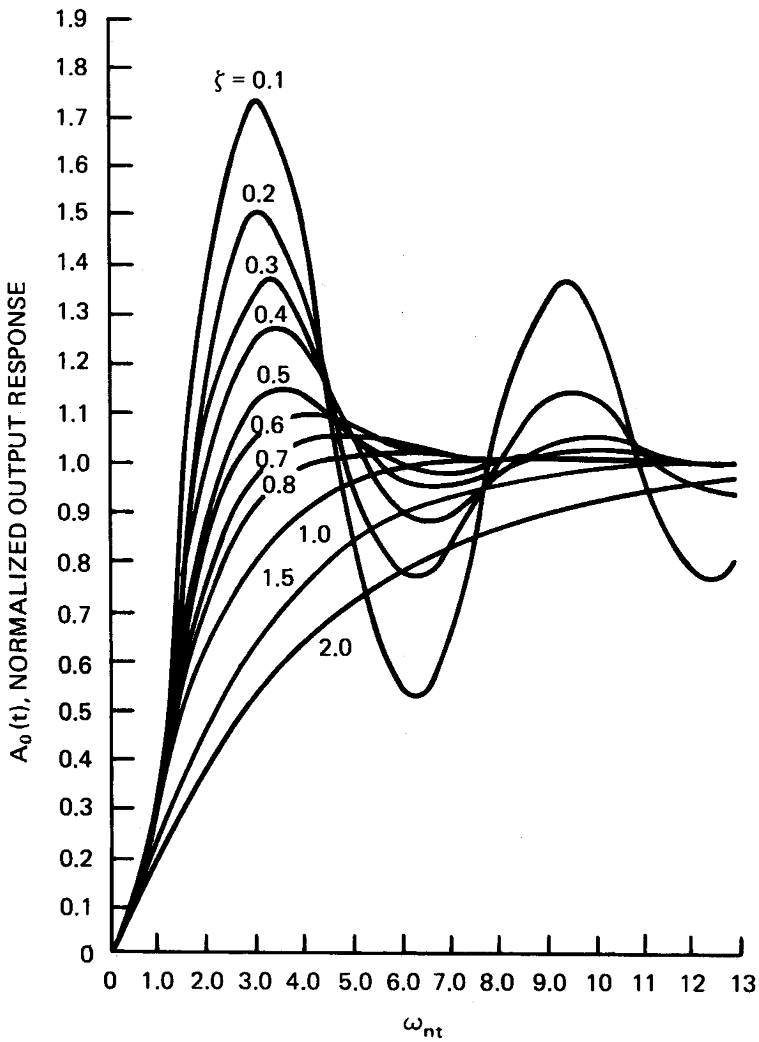
STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select program		[2nd] [Pgm] 02	
2	Initialize		[SBR] [CLR]	0.
3	Enter loop gain[1,5]	G	[A]	G
4	Enter C for PLL filter in farads	C	[C]	C
5	Enter integer divisor[2] (omit this step for passive PLL filter)	N	[2nd] [A′]	N
6	Enter remaining PLL parameters as required[5]			
	a. Natural frequency in radians/second	ω_n	[B]	ω_n
	b. Damping factor	ζ	[D]	ζ
	c. R_1 in ohms	R_1	[2nd] [B′]	R_1
	d. R_2 in ohms	R_2	[2nd] [C′]	R_2
7	Compute desired PLL (active filter) parameters			
	a. Natural frequency in radians/second		[E] [B]	ω_n
	b. Damping factor		[E] [D]	ζ
	c. R_1 in ohms		[E] [2nd] [B′]	R_1
	d. R_2 in ohms		[E] [2nd] [C′]	R_2
8	Compute desired PLL (passive filter) parameters			
	a. Natural frequency in radians/second		[2nd] [E′] [B]	ω_n
	b. Damping factor		[2nd] [E′] [D]	ζ
	c. R_1 in ohms[3]		[2nd] [E′] [2nd] [B′]	R_1
	d. R_2 in ohms[3]		[2nd] [E′] [2nd] [C′]	R_2
9	Compute loop-noise bandwidth in		[2nd] [D′]	B

NOTES:

1. The dimension for the loop gain is 1/second since $G = K_p K_v$ where K_p = phase detector gain in volts/radian and K_v = VCO gain in radians/second volt.
2. Input must be an integer ≥1. Enter 1 for N when working with basic PLL design ($f_{in} = f_{out}$).
3. Flashing 9's in the display indicates that the input quantities G, ω_n, or ζ need to be modified so that
$$0 < \frac{2\zeta\omega_n G - \omega_n^2}{G^2} < 1.$$
This insures that the resistances will be greater than zero.
4. Loop-noise bandwidth is one-sided.
5. Inputs must be greater than zero.

Example 1: Determine the active filter component values for a phase-locked loop with the following specifications. The loop must achieve a lock so that $A_0(t)$ is within 10% of its final value after 500 milliseconds with a damping factor (ζ) of 0.5. From the curves below, where the normalized output response, $A_0(t)$, is plotted as a function of the normalized time ($\omega_n t$), $\zeta = 0.5$ is within 10% at $\omega_n t = 4.5$. Thus, $\omega_n = 4.5/.5 = 9$ radians/second.

Type 2 Second Order Step Response



Data sheet specifications for the phase detector give a value of 0.111 volts/radian for the gain constant. The measured value for the gain of the VCO is 9.5×10^6 radians/second volt. The output frequency is 10 megahertz with a channel spacing of 250 hertz. Thus, N = 10 MHz/ 250 Hz = 40000 and loop gain = (0.111) (9.5×10^6) = 1054500. The value for C is chosen to be 10 microfarads.

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 02		Select program
	[SBR] [CLR]	0.	Initialize
1054500	[A]	1054500.	Enter loop gain
.00001	[C]	0.00001	Enter C in farads
40000	[2nd] [A']	40000.	Enter integer divisor
9	[B]	9.	Enter natural frequency
.5	[D]	0.5	Enter damping factor
	[E] [2nd] [B']	32546.2963	R_1
	[E] [2nd] [C']	11111.11111	R_2
	[2nd] [D']	4.5	Loop-noise bandwidth

Example 2: A phase-locked loop has an overall loop gain of 20000. If the passive filter component values are: $R_1 = 5\,k\Omega$, $R_2 = 500\Omega$, and $C = 50\mu F$, what is the natural frequency damping factor, and noise bandwidth for the loop?

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 02		Select program
	[SBR] [CLR]	0.	Initialize
20000	[A]	20000.	Enter loop gain
.00005	[C]	0.00005	Enter C in farads
5000	[2nd] [B']	5000.	Enter R_1 in ohms
500	[2nd] [C']	500.	Enter R_2 in ohms
	[2nd] [E'] [B]	269.679945	Compute natural frequency
	[2nd] [E'] [D]	3.377741311	Compute damping factor
	[2nd] [D']	465.4345854	Compute loop-noise bandwidth

Register Contents

R_{00}	R_{05}	R_{10} G	R_{15}	R_{20} N	R_{25}
R_{01}	R_{06}	R_{11} R_1	R_{16}	R_{21}	R_{26}
R_{02}	R_{07}	R_{12} R_2	R_{17} ω_n	R_{22}	R_{27}
R_{03}	R_{08}	R_{13} C	R_{18} ζ	R_{23}	R_{28}
R_{04}	R_{09}	R_{14}	R_{19}	R_{24}	R_{29}

Method Used

$F(S) = \dfrac{S\tau_2 + 1}{S\tau_1}$ for an active filter provided that the amplifier gain is very large and

$F(S) = \dfrac{S\tau_2 + 1}{S(\tau_1 + \tau_2) + 1}$ for a passive filter where $\tau_1 = R_1C$, $\tau_2 = R_2C$.

Using the transfer function representative F(S) for the passive filter, the basic PLL transfer function becomes

$$\frac{\theta_o}{\theta_i}(S) = \frac{G(S\tau_2 + 1)/(\tau_1 + \tau_2)}{S^2 + S(1 + G\tau_2)/(\tau_1 + \tau_2) + G/(\tau_1 + \tau_2)}$$

In order to preserve the analogy with second-order servo controls, one can substitute for the time constants τ_1 and τ_2, the parameters of ω_n, the natural angular frequency, and ζ, the damping factor. Making these substitutions,

$$\frac{\theta_o}{\theta_i}(S) = \frac{S\omega_n(2\zeta - \omega_n/G) + \omega_n^2}{S^2 + 2\zeta\omega_n S + \omega_n^2}$$

where, by definition

$$\omega_n = \sqrt{\frac{G}{\tau_1 + \tau_2}} \quad \text{and} \quad \zeta = \frac{\omega_n}{2}\left(\tau_2 + \frac{1}{G}\right)$$

A similar argument applies in the case where the loop filter is active except that the (÷N) counter changes the form of the PLL transfer function,

$$\frac{\theta_o}{\theta_i}(S) = \frac{G\,F(S)/N}{S + G\,F(S)/N}$$

Using the transfer function representation F(S) for the active filter, the PLL transfer function becomes

$$\frac{\theta_o}{\theta_i}(S) = \frac{G(S\tau_2 + 1)/N\tau_1}{S^2 + S(G\tau_2/N\tau_1) + G/N\tau_1} \quad \text{or} \quad \frac{\theta_o}{\theta_i}(S) = \frac{2\zeta\omega_n S + \omega_n^2}{S^2 + 2\zeta\omega_n S + \omega_n^2}$$

where, by definition

$$\omega_n = \sqrt{\frac{G}{N\tau_1}} \quad \text{and} \quad \zeta = \frac{\omega_n\tau_2}{2}$$

The loop filter components R_1 and R_2 are computed using the defining equations for ω_n and ζ.

Active filter calculations for R_1 and R_2

$$\omega_n = \sqrt{\frac{G}{N\tau_1}} \quad \Rightarrow \quad R_1 = \frac{G}{N\omega_n^2 C}$$

$$\zeta = \frac{\omega_n}{2}\tau_2 \quad \Rightarrow \quad R_2 = \frac{2\zeta}{\omega_n C}$$

Passive filter calcualtions for R_1 and R_2

$$\omega_n = \sqrt{\frac{G}{\tau_1 + \tau_2}} \quad \Rightarrow \quad R_1 = \frac{G}{\omega_n^2 C} - R_2$$

$$\zeta = \frac{\omega_n}{2}(\tau_2 + \frac{1}{G}) \quad \Rightarrow \quad R_2 = \frac{2\zeta}{\omega_n C} - \frac{1}{GC}$$

NOTE: In order to insure R_1, $R_2 > 0$ in the passive case,

$$0 < \frac{2\zeta\omega_n G - \omega_n^2}{G^2} < 1$$

Finally, for both active and passive loops, the one-sided loop-noise bandwidth B in hertz is computed

$$B = \frac{\omega_n}{2}\ (\zeta + \frac{1}{4\zeta})$$

Reference: *Phaselock Techniques,* Floyd M. Gardner, John Wiley & Sons 1966.

S⇌Y PARAMETER CONVERSIONS

Small-signal, two-port "black boxes" are often characterized in terms of complex scattering (S) parameters normalized to a common input and output transmission line impedance Z_0. The translation of these parameters to admittance (Y) parameters is necessary to create equivalent circuit models. This program converts S parameters to Y parameters or vice versa.

The matrix equation relating these parameters is

$$\begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} = \begin{bmatrix} \dfrac{1 - S_{11} + S_{22} - \Delta S}{1 + S_{11} + S_{22} + \Delta S} & \dfrac{-2S_{12}}{1 + S_{11} + S_{22} + \Delta S} \\[2ex] \dfrac{-2S_{21}}{1 + S_{11} + S_{22} + \Delta S} & \dfrac{1 + S_{11} - S_{22} - \Delta S}{1 + S_{11} + S_{22} + \Delta S} \end{bmatrix}$$

where $\Delta S = S_{11}\ S_{22} - S_{12}\ S_{21}$ and the S parameters are input in polar form $(R, < \theta)$.

The relationship for converting from Y parameters to S parameters may be obtained by interchanging Y and S in the above matrix equation. The angle θ representation for the respective parameters as output from this program is in degrees where $-90° \leqslant \theta < 270°$.

When converting from S to Y parameters, once the matrix elements are calculated, the Y matrix is multiplied by the real scalar Z_0^{-1} to give the actual Y parameter.

When converting from Y to S parameters, the Y parameters are multiplied by Z_0 before the matrix elements are calculated to give the actual S parameters.

Solid State Software TI © 1979

S ⇄ Y PARAMETER CONVERSIONS				EE-03
$\rightarrow|S|_{11};\angle_{11}$	$\rightarrow|S|_{12};\angle_{12}$	$\rightarrow|S|_{21};\angle_{21}$	$\rightarrow|S|_{22};\angle_{22}$	S ← Y
$|S|_{11};\angle_{11}$	$|S|_{12};\angle_{12}$	$|S|_{21};\angle_{21}$	$|S|_{22};\angle_{22}$	S → Y

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select program		[2nd] [Pgm] 03	
2	Initialize[1]		[SBR] [CLR]	0.
3	Enter polar form of S (or Y) matrix[2]			
	a. Magnitude S_{11} (or Y_{11})	$|S_{11}(Y_{11})|$	[x⪆t]	0.
	b. Angle S_{11} (or Y_{11})	$\angle S_{11}(Y_{11})$	[A]	Real S_{11} (or Y_{11})
	c. Magnitude S_{12} (or Y_{12})	$|S_{12}(Y_{12})|$	[x⪆t]	Imag S_{11} (or Y_{11})
	d. Angle S_{12} (or Y_{12})	$\angle S_{12}(Y_{12})$	[B]	Real S_{12} (or Y_{12})
	e. Magnitude S_{21} (or Y_{21})	$|S_{21}(Y_{21})|$	[x⪆t]	Imag S_{12} (or Y_{12})
	f. Angle S_{21} (or Y_{21})	$\angle S_{21}(Y_{21})$	[C]	Real S_{21} (or Y_{21})
	g. Magnitude S_{22} (or Y_{22})	$|S_{22}(Y_{22})|$	[x⪆t]	Imag S_{21} (or Y_{21})
	h. Angle S_{22} (or Y_{22})	$\angle S_{22}(Y_{22})$	[D]	Real S_{22} (or Y_{22})
	i. Display imag S_{22} (or Y_{22}), optional		[x⪆t]	Imag S_{22} (or Y_{22})
4	Enter real characteristic impedance in ohms[3] and press one of the following	Z_0		Z_0
	Convert S→Y		[E]	0.
	Convert Y→S		[2nd] [E′]	0.
5.	Display conversion results			
	a. Magnitude Y_{11} (or S_{11})		[2nd] [A′]	$|Y_{11}(S_{11})|$†
	b. Angle Y_{11} (or S_{11})[4]		[x⪆t]	$\angle Y_{11}(S_{11})$†
	c. Magnitude Y_{12} (or S_{12})		[2nd] [B′]	$|Y_{12}(S_{12})|$†
	d. Angle Y_{12} (or S_{12})		[x⪆t]	$\angle Y_{12}(S_{12})$†
	e. Magnitude Y_{21} (or S_{21})		[2nd] [C′]	$|Y_{21}(S_{21})|$†
	f. Angle Y_{21} (or S_{21})		[x⪆t]	$\angle Y_{21}(S_{21})$†
	g. Magnitude Y_{22} (or S_{22})		[2nd] [D′]	$|Y_{22}(S_{22})|$†
	h. Angle Y_{22} (or S_{22})		[x⪆t]	$\angle Y_{22}(S_{22})$†

NOTES:

1. Default value for Z_0 set at 50 ohms.
2. Angle input is in degrees.
3. Enter 0 if default value of 50 ohms or last-entered value for Z_0 is desired.
4. Angle output θ is in degrees where $-90^\circ \leqslant \theta < 270^\circ$.

†These values are printed if printer is connected.

Example: Convert the following S parameters to Y parameters:

$S_{11} = .1\angle 50°$, $S_{12} = .2\angle 55°$, $S_{21} = .3\angle 60°$, and $S_{22} = .4\angle 65°$.

$Z_0 = 50$ ohms

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 03		Select program
	[SBR] [CLR]	0.	Initialize
.1	[x⋛t]	0.	Enter $\vert S_{11}\vert$
50	[A]	0.064278761	Enter $\angle S_{11}$
.2	[x⋛t]	.0766044443	Enter $\vert S_{12}\vert$
55	[B]	.1147152873	Enter $\angle S_{12}$
.3	[x⋛t]	.1638304089	Enter $\vert S_{21}\vert$
60	[C]	0.15	Enter $\angle S_{21}$
.4	[x⋛t]	.2598076211	Enter $\vert S_{22}\vert$
65	[D]	.1690473047	Enter $\angle S_{22}$
	[x⋛t]	.3625231148	Display imag S_{22}
0		0.	Use default value for Z_0
	[E]	0.	Convert S→Y
Display conversion results			
	[2nd] [A']	.0173534572†	$\vert Y_{11}\vert$
	[x⋛t]	−3.227794565†	$\angle Y_{11}$
	[2nd] [B']	.0061012628†	$\vert Y_{12}\vert$
	[x⋛t]	216.2717472†	$\angle Y_{12}$
	[2nd] [C']	.0091518942†	$\vert Y_{21}\vert$
	[x⋛t]	221.2717472†	$\angle Y_{21}$
	[2nd] [D']	.0141294793†	$\vert Y_{22}\vert$
	[x⋛t]	−35.53169433†	$\angle Y_{22}$

†These values are printed if printer is connected.

Register Contents

R_{00}	R_{05} $\vert\ \vert_{11}$	R_{10} $\angle_{21}$	R_{15} ΔS, ΔY (real)	R_{20}	R_{25}
R_{01} Used	R_{06} $\angle_{11}$	R_{11} $\vert\ \vert_{22}$	R_{16} ΔS, ΔY (imag)	R_{21}	R_{26}
R_{02} Used	R_{07} $\vert\ \vert_{12}$	R_{12} $\angle_{22}$	R_{17} Z_0	R_{22}	R_{27}
R_{03} Used	R_{08} $\angle_{12}$	R_{13} S_{22}-S_{11} Y_{22}-Y_{11} (real)	R_{18}	R_{23}	R_{28}
R_{04} Used	R_{09} $\vert\ \vert_{21}$	R_{14} S_{22}-S_{11} Y_{22}-Y_{11} (imag)	R_{19}	R_{24}	R_{29}

COMPLEX ARITHMETIC

For two given complex numbers $X = a + bi$ or $r_1 \angle \theta_1$ and $Y = c + di$ or $r_2 \angle \theta_2$, this program calculates the following:

$X + Y$

$X - Y$

$X \times Y$

$X \div Y$

Y^X

$\sqrt[X]{Y}$

$\log_Y X$

These operations may be chained in the following manner. Initially, two complex numbers are entered with the first number being X and the second number Y. After a function has been performed, the result becomes the new X, and a new Y may be entered. X and Y may be interchanged when necessary.

Also, a result from this program is stored in R_{01} and R_{02} and may be used (without reentering it) as the input for this program or for programs EE-05 and EE-06. In other words, Steps 2a and 2b or 2c, 2d and 2e may be omitted in sequential calculations using these programs.

Solid State Software			TI © 1979	
COMPLEX ARITHMETIC				EE-04
Y	→ X−Y	→ X+Y	→ $\log_Y X$	X ⇄ Y
X	→ X+Y	→ X×Y	→ Y^x	→ $\sqrt[x]{Y}$

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select program.		[2nd] [Pgm] 04	
	If rectangular form:			
2a	Enter real part of X	a	[A]	a
2b	Enter imaginary part of X	b	[A]	b
	(2a and 2b must be performed in sequence) OR			
	If polar form:			
2c	Enter modulus of X	r_1	[A]	r_1
2d	Enter argument of X in degrees	θ_1	[A]	θ_1
2e	Convert polar to rectangular		[SBR] [2nd] [P→R]	0.
	(2c, 2d, and 2e must be performed in sequence)			
	If rectangular form:			
3a	Enter real part of Y	c	[2nd] [A′]	c
3b	Enter imaginary part of Y	d	[2nd] [A′]	d
	(3a and 3b must be performed in sequence) OR			
	If polar form:			
3c	Enter modulus of Y	r_2	[2nd] [A′]	r_2
3d	Enter argument of Y in degrees	θ_2	[2nd] [A′]	θ_2
3e	Convert polar to rectangular		[SBR] [2nd] [P→R]	0.
	(3c, 3d, and 3e must be performed in sequence)			
	Perform either Step 4, 5, 6, 7, 8, 9, or 10.			
4	Calculate X + Y		[B] [x⋛t]	real part imaginary part
5	Calculate X − Y		[2nd] [B′] [x⋛t]	real part imaginary part
6	Calculate X × Y		[C] [x⋛t]	real part imaginary part
7	Calculate X ÷ Y		[2nd] [C′] [x⋛t]	real part imaginary part
8	Calculate Y^x		[D] [x⋛t]	real part imaginary part
9	Calculate $\log_Y X$		[2nd] [D′] [x⋛t]	real part imaginary part
10	Calculate $\sqrt[x]{Y}$		[E] [x⋛t]	real part imaginary part
	After a calculation, the result becomes the new X.			
11	To swap X and Y		[2nd] [E′]	0.

Example 1: X = 2+3i, Y = 2∠30°

Find X + Y

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 04		Select program
2	[A]	2.	a
3	[A]	3.	b
2	[2nd] [A']	2.	r
30	[2nd] [A']	30.	θ
	[SBR] [2nd] [P→R]	0.	Convert polar to rectangular
	[B]	3.732050808	Real (X + Y)
	[x⪆t]	4.	Imaginary (X + Y)

Calculate polar form of X + Y

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 05		Select program
	[B]	5.470667531	r
	[x⪆t]	46.98475372	θ

Example 2: Solve the following expression.

$$[(2 + 3i)(1 - i)]^{(1+i)}$$

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 04		Select program
2	[A]	2.	a
3	[A]	3.	b
1	[2nd] [A']	1.	c
1	[+/−] [2nd] [A']	−1.	d
	[C]	5.	Real (X × Y)
	[x⪆t]	1.	Imaginary (X × Y)
	[2nd] [E']	0.	X ⇌ Y
1	[A]	1.	a
1	[A]	1.	b
	[D]	−1.058423508	Real (Y^X)
	[x⪆t]	4.049577726	Imaginary (Y^X)

Register Contents

R_{00}	R_{05}	R_{10}	R_{15}	R_{20}	R_{25}
R_{01} a	R_{06}	R_{11}	R_{16}	R_{21}	R_{26}
R_{02} b	R_{07}	R_{12}	R_{17}	R_{22}	R_{27}
R_{03} Used	R_{08}	R_{13}	R_{18}	R_{23}	R_{28}
R_{04} Used	R_{09}	R_{14}	R_{19}	R_{24}	R_{29}

Method Used

$$X = a + bi \qquad Y = c + di$$

$$X \div Y = \frac{ac + bd}{c^2 + d^2} + \frac{bc - ad}{c^2 + d^2} i$$

$$X + Y = (a + c) + (b + d)i$$

$$Y^X = e^{X \ln Y}, \; Y \neq 0$$

$$X - Y = (a - c) + (b - d)i$$

$$\sqrt[X]{Y} = e^{\frac{\ln Y}{X}}, \; X \neq 0, \; Y \neq 0$$

$$X \times Y = (ac - bd) + (ad + bc)i$$

$$\log_Y X = \frac{\ln X}{\ln Y}, \; X \neq 0, \; Y \neq 0$$

For calculation of e^X, ln X, and 1/X for complex numbers, see program EE-05.

COMPLEX FUNCTIONS

The following functions are calculated for the complex number $X = a + bi$ or $r\angle\theta$.

- Polar representation (r,θ) of X, or
- Rectangular representation $(a + bi)$ of X
- X^2
- $\sqrt{X}$
- $1/X$
- e^X
- $\ln X$

After a function has been performed, the result is stored in R_{01} and R_{02} and becomes the new X. Therefore, a result may be used without reentering it in this program and in programs EE-04 and EE-06.

Solid State Software			TI © 1979	
COMPLEX FUNCTIONS				EE-05
→ ln X	→ e^x		→ a+bi	
X	→ r, θ	→ X^2	→ $\sqrt{X}$	→ 1/X

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select program.		[2nd] [Pgm] 05	
	If rectangular form:			
2a	Enter real part of X	a	[A]	a
2b	Enter imaginary part of X	b	[A]	b
	(2a and 2b must be performed in sequence)			
	OR			
	If polar form:			
2c	Enter modulus of X	r	[A]	r
2d	Enter argument of X in degrees	θ	[A]	θ
2e	Convert polar to rectangular		[SBR] [2nd] [P→R]	0.
	(2c, 2d, and 2e must be performed in sequence)			
3	Calculate polar form of X, if desired		[B] [x≷t]	r θ
4	Calculate rectangular form of X, if desired		[2nd] [D′] [x≷t]	a b
	Perform either **Step 5, 6, 7, 8, or 9**			
5	Calculate X^2		[C] [x≷t]	real part imaginary part
6	Calculate $\sqrt{X}$		[D] [x≷t]	real part imaginary part
7	Calculate 1/X		[E] [x≷t]	real part imaginary part
8	Calculate ln X		[2nd] [A′] [x≷t]	real part imaginary part
9	Calculate e^X		[2nd] [B′] [x≷t]	real part imaginary part
	After a calculation, the result becomes the new X.			

Example 1: Find X^2 if $X = 2\angle 60°$

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 05		Select program
2	[A]	2.	r
60	[A]	60.	θ
	[SBR] [2nd] [P→R]	0.	Convert polar to rectangular
	[C]	−2.	Real (X^2)
	[x≷t]	3.464101615	Imaginary (X^2)

Calculate polar form of X^2

ENTER	PRESS	DISPLAY	COMMENTS
	[B]	4.	r
	[x≷t]	120.	θ

Example 2: Find ln X^2 if $X = 2 + 3i$

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 05		Select program
2	[A]	2.	a
3	[A]	3.	b
	[C]	−5.	Real (X^2)
	[x≷t]	12.	Imaginary (X^2)
	[2nd] [A']	2.564949357	Real (ln X^2)
	[x≷t]	1.965587446	Imaginary (ln X^2)

Register Contents

R_{00}	R_{05}	R_{10}	R_{15}	R_{20}	R_{25}
R_{01} a	R_{06}	R_{11}	R_{16}	R_{21}	R_{26}
R_{02} b	R_{07}	R_{12}	R_{17}	R_{22}	R_{27}
R_{03} Used	R_{08}	R_{13}	R_{18}	R_{23}	R_{28}
R_{04} Used	R_{09}	R_{14}	R_{19}	R_{24}	R_{29}

Method Used

$$X = a + bi$$

$$\text{Magnitude of } X = r = \sqrt{a^2 + b^2}$$

Angle of X (degrees) = θ, where $-90° \leqslant \theta < 270°$**

$$\theta = \begin{cases} \tan^{-1} b/a & \text{if } a \neq 0 \\ 90° & \text{if } a = 0,\ b > 0 \\ -90° & \text{if } a = 0,\ b < 0 \end{cases}$$

$$X^2 = r^2 (\cos 2\theta + i \sin 2\theta)$$

$$\sqrt{X} = \sqrt{r}\ (\cos \frac{\theta}{2} + i \sin \frac{\theta}{2})$$

$$\frac{1}{X} = \frac{1}{a + bi}$$

$$e^X = e^a \cos b + i\, e^a \sin b$$

$$\ln X = \ln r + i\theta,\ X \neq 0$$

**See page V-31 of *Personal Programming.*

COMPLEX TRIGONOMETRIC FUNCTIONS

This program calculates the value of trigonometric functions for a complex number $X = a + bi$ or $r\angle\theta$. The following functions are evaluated:

sin X

cos X

tan X

$\sin^{-1}$ X

$\cos^{-1}$ X

$\tan^{-1}$ X

The result of any function is stored in rectangular form in R_{01} and R_{02} and becomes the new value of X. Therefore a result may be used without reentering it in this program and in programs EE-04 and EE-05.

Solid State Software TI © 1979

COMPLEX TRIGONOMETRIC FUNCTIONS				EE-06
	→ $\sin^{-1}$X	→ $\cos^{-1}$X	→ $\tan^{-1}$X	
X	→ sin X	→ cos X	→ tan X	

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select program.		[2nd] [Pgm] 06	
	If rectangular form:			
2a	Enter real part of X	a	[A]	a
2b	Enter imaginary part of X	b	[A]	b
	(2a and 2b must be performed in sequence) OR			
	If polar form:			
2c	Enter modulus of X	r	[A]	r
2d	Enter argument of X in degrees	θ	[A]	θ
2e	Convert polar to rectangular		[SBR] [2nd] [P→R]	0.
	(2c, 2d, and 2e must be performed in sequence)			
	Perform either Step 3, 4, 5, 6, 7, or 8			
3	Calculate sin X		[B] [x≷t]	real part imaginary part
4	Calculate cos X		[C] [x≷t]	real part imaginary part
5	Calculate tan X		[D] [x≷t]	real part imaginary part
6	Calculate $\sin^{-1}$ X		[2nd] [B′] [x≷t]	real part imaginary part
7	Calculate $\cos^{-1}$ X		[2nd] [C′] [x≷t]	real part imaginary part
8	Calculate $\tan^{-1}$ X		[2nd] [D′] [x≷t]	real part imaginary part

NOTES: 1. After a calculation, the result becomes the new X.

2. X is expressed in radians. Program leaves calculator in radian mode.

Example 1: Find tan X if X = 3∠56°

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 06		Select program
3	[A]	3.	r
56	[A]	56.	θ
	[SBR] [2nd] [P→R]	0.	Convert polar to rectangular
	[D]	−.0029707247	Real (tan X)
	[x≷t]	1.013601147	Imaginary (tan X)

Use Pgm 05 to convert to polar form of tan X if desired.

	[2nd] [Pgm] 05		Select program
	[B]	1.0136055	r
	[x≷t]	90.16792552	θ

Example 2: Find sin X if X = 2 + 3i

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 06		Select program
2	[A]	2.	a
3	[A]	3.	b
	[B]	9.154499147	Real (sin X)
	[x≷t]	−4.16890696	Imaginary (sin X)

Register Contents

R_{00} Used	R_{05}	R_{10}	R_{15}	R_{20}	R_{25}
R_{01} a	R_{06}	R_{11}	R_{16}	R_{21}	R_{26}
R_{02} b	R_{07}	R_{12}	R_{17}	R_{22}	R_{27}
R_{03} Used	R_{08}	R_{13}	R_{18}	R_{23}	R_{28}
R_{04} Used	R_{09}	R_{14}	R_{19}	R_{24}	R_{29}

Method Used

$$\sin X = \frac{e^{iX} - e^{-iX}}{2i}$$

$$\cos X = \frac{e^{iX} + e^{-iX}}{2}$$

$$\tan X = \sin X/\cos X$$

$$\sin^{-1} X = \sin^{-1} (a \pm bi) = \sin^{-1} B + i \ln [A \pm (A^2 - 1)^{½}]$$

$$\cos^{-1} X = \cos^{-1} (a \pm bi) = \pm\cos^{-1} B - i \ln [A + (A^2 - 1)^{½}]$$

where

$$A = \frac{1}{2} [(a + 1)^2 + b^2]^{½} + \frac{1}{2} [(a - 1)^2 + b^2]^{½}$$

$$B = \frac{1}{2} [(a + 1)^2 + b^2]^{½} - \frac{1}{2} [(a - 1)^2 + b^2]^{½}$$

$$\tan^{-1} X = \frac{1}{2} \tan^{-1} \left[\frac{2a}{1 - a^2 - b^2}\right] + \frac{i}{4} \ln \left[\frac{a^2 + (b + 1)^2}{a^2 + (b - 1)^2}\right]$$

Note: X is in radians.

DECIBELS, NEPERS, POWER, VOLTAGE CURRENT RATIO CONVERSIONS

Power, voltage, and current ratios are often expressed in decibels (dB) or nepers (Np). This program converts between any of these when one is given. Input and output voltages and currents are assumed to be measured at points having the same impedance. If this is not the case, then decibel and neper values may be calculated from the output-to-input voltage or current ratio if V_2/V_1 is replaced by $V_2\sqrt{Z_1}/V_1\sqrt{Z_2}$ and I_2/I_1 with $I_2\sqrt{Z_2}/I_1\sqrt{Z_1}$ where Z_1 is the input impedance and Z_2 is the output impedance.

Solid State Software				TI © 1979
dB,Np,P,V,I RATIO CONVERSIONS				EE-07
P_2/P_1	$V_2/V_1=I_2/I_1$	Np	dB	→ Execute

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select program.		[2nd] [Pgm] 07	
2	Initialize		[SBR] [CLR]	0.
3	Input any one of the following:			
	Power ratio[1]	P_2/P_1	[A]	P_2/P_1
	Voltage or Current ratio[1]	V_2/V_1 or I_2/I_1	[B]	V_2/V_1 or I_2/I_1
	Nepers	Np	[C]	Np
	Decibels	dB	[D]	dB
4	Calculate any of the following:			
	Power ratio		[E] [A]	P_2/P_1
	Voltage and Current ratio		[E] [B]	$V_2/V_1 = I_2/I_1$
	Nepers[2]		[E] [C]	Np
	Decibels[2]		[E] [D]	dB

Notes: 1. P_2/P_1, V_2/V_1, I_2/I_1 must be greater than zero.

2. Negative values denote an Np or dB loss.

Example 1: For a circuit with a voltage ratio of 3.1623, calculate the power ratio, dB and Np values. Assume equal impedances at measuring points.

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 07		Select program
	[SBR] [CLR]	0.	Initialization
3.1623	[B]	3.1623	Voltage ratio
	[E] [A]	10.00014129	Power ratio
	[E] [C]	1.151299611	Nepers
	[E] [D]	10.00006136	Decibels

Example 2: What is the loss in decibels for an attenuator which has 8 volts at the input and 2 volts at the output? The input and output impedances are equal and terminated properly.

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 07		Select program
	[SBR] [CLR]	0.	Initialization
.25	[B]	0.25	Voltage ratio
	[E] [D]	−12.04119983	Decibels

Register Contents

R_{00}	R_{05}	R_{10} P_2/P_1	R_{15}	R_{20}	R_{25}
R_{01}	R_{06}	R_{11} $V_2/V_1 = I_2/I_1$	R_{16}	R_{21}	R_{26}
R_{02}	R_{07}	R_{12} Np	R_{17}	R_{22}	R_{27}
R_{03}	R_{08}	R_{13} dB	R_{18}	R_{23}	R_{28}
R_{04}	R_{09}	R_{14}	R_{19}	R_{24}	R_{29}

Method Used

1. Given the power ratio P_2/P_1:

$$dB = 10 \log (P_2/P_1) \qquad Np = .5 \ln (P_2/P_1) \qquad V_2/V_1 = e^{Np}$$

2. Given the voltage or current ratio $V_2/V_1 = I_2/I_1$:

$$dB = 20 \log (V_2/V_1) \qquad P_2/P_1 = 10^{dB/10} \qquad Np = \ln (V_2/V_1)$$

3. Given the decibels dB:

$$P_2/P_1 = 10^{dB/10} \qquad Np = .5 \ln (P_2/P_1) \qquad V_2/V_1 = e^{Np}$$

4. Given the nepers Np:

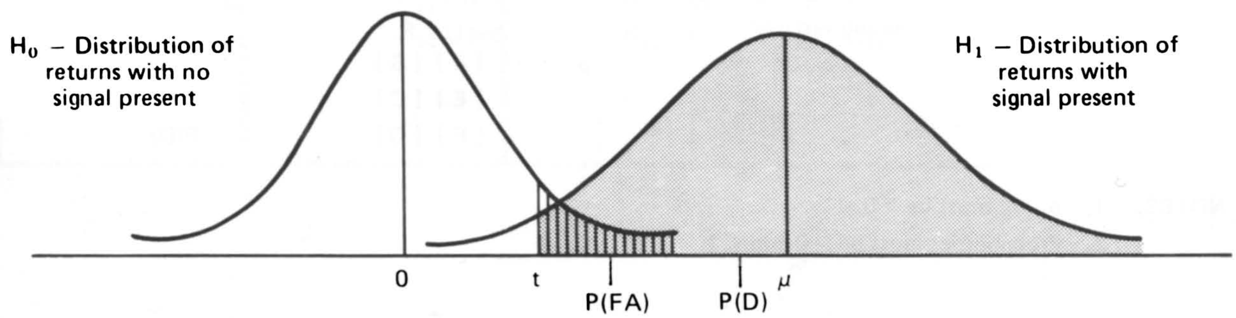
$$P_2/P_1 = e^{2Np} \qquad dB = 10 \log (P_2/P_1) \qquad V_2/V_1 = e^{Np}$$

SIGNAL DETECTION

There are many circumstances in which one must decide if a signal is present, even though noise and other variations cloud the issue. This program relates areas under two overlapping normal curves and determines the probability of falsely declaring a signal present when none exists (probability of a false alarm), as well as the probability of properly declaring a signal present (probability of detection).

This program is stated in terms of radar detection; however, this procedure applies to all problems of this type. The radar return to a given range cell is subject to both internally and externally generated noise. Under many conditions this return is approximately normally distributed about a mean we may take to be 0 with a standard deviation σ_0. We'll call this normal curve H_0.

The presence of a point target in this range cell adds energy to the distribution and alters the return to a normal distribution with mean μ and standard deviation σ_1. We'll call this normal curve H_1.



Let t represent a threshold value which one may vary. When the signal strength is greater than t, a signal is declared present. When the signal strength is less than t, no signal is declared. The area under H_0 to the right of t represents the probability of a false alarm P(FA), and the area under H_1 to the right of t represents the probability of detection, P(D).

The signal-to-noise ratio (in dB) may be defined as

$$SNR = 20 \log_{10} (\mu/\sigma_0)$$

Once σ_1/σ_0 has been entered, this program calculates

1) SNR given P(FA) and the P(D)
2) P(FA) given P(D) and SNR
3) P(D) given P(FA) and SNR

Solid State Software			TI © 1979	
SIGNAL DETECTION				EE-08
σ_1/σ_0	SNR	P(FA)	P(D)	→ Execute

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select program		[2nd] [Pgm] 08	
2	Initialize		[SBR] [CLR]	0.
3	Enter ratio of standard deviations	σ_1/σ_0[1]	[A]	σ_1/σ_0
4	Enter two of the following in any order		[B]	
	a. Signal-to-noise ratio (dB)	SNR	[B]	$\mu/\sigma_0 = 10^{SNR/20}$
	b. Probability of declaring a signal present when none exists	P(FA)[2]	[C]	P(FA)
	c. Probability of detecting a signal	P(D)[2]	[D]	P(D)
5	Calculate the remaining variable			
	SNR		[E] [B]	SNR
	P(FA)		[E] [C]	P(FA)
	P(D)		[E] [D]	P(D)

NOTES: 1. σ_1/σ_0 must be >0.
2. Probabilities must be >0 and <1.

Example 1: For $\sigma_1 = \sqrt{2}\sigma_0$, what is the false alarm probability when the threshold is set for a detection probability of 0.9 and the signal-to-noise ratio is 10 dB.

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 08		Select program
	[SBR] [CLR]	0.	Initialize
2	[$\sqrt{x}$] [A]	1.414213562	Enter σ_1/σ_0
.9	[D]	0.9	Enter P(D)
10	[B]	3.16227766	Enter SNR
	[E] [C]	.0885568121	Compute P(FA)

Under the same conditions, what signal-to-noise ratio (in dB) is required to achieve a false alarm probability of 0.001. Note that σ_1/σ_0 and P(D) need not to be reentered.

ENTER	PRESS	DISPLAY	COMMENTS
.001	[C]	0.001	Enter P(FA)
	[E] [B]	13.80952205	Compute SNR

Example 2: For $\sigma_1 = 2\sigma_0$ and a signal-to-noise ratio of 20 dB, what is the probability of detection, if the probability of a false alarm is set at 1×10^{-9}.

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 08		Select program
	[SBR] [CLR]	0.	Initialize
2	[A]	2.	Enter σ_1/σ_0
9	[+/–] [INV] [2nd] [log] [C]	0.000000001	Enter P(FA)
20	[B]	10.	Enter SNR
	[E] [D]	.9773102447	Compute P(D)

Register Contents

R_{00}	R_{05}	R_{10} σ_1/σ_0	R_{15} Used	R_{20}	R_{25}
R_{01}	R_{06}	R_{11} μ/σ_0	R_{16} Used	R_{21}	R_{26}
R_{02}	R_{07}	R_{12} P(FA)	R_{17} Used	R_{22}	R_{27}
R_{03}	R_{08}	R_{13} P(D)	R_{18} Used	R_{23}	R_{28}
R_{04}	R_{09}	R_{14} Used	R_{19}	R_{24}	R_{29}

Method Used

SIGNAL DETECTION FORMULAS

Let U represent the normalized values of X for the normal curve in question.

1. $$P(D) = \frac{1}{\sqrt{2\pi}} \int_{U_{P(D)}}^{\infty} e^{-t^2/2}\, dt$$

2. $$P(FA) = \frac{1}{\sqrt{2\pi}} \int_{U_{P(FA)}}^{\infty} e^{-t^2/2}\, dt$$

3. $$U_{P(FA)} = \frac{X_{P(FA)}}{\sigma_0} = \frac{X_{P(D)}}{\sigma_0}$$

4. $$U_{P(D)} = \frac{\dfrac{X_{P(D)}}{\sigma_0} - \dfrac{\mu}{\sigma_0}}{\dfrac{\sigma_1}{\sigma_0}}$$

5. $$U_{P(FA)} = \frac{X_{P(FA)}}{\sigma_0} = U_{P(D)} \frac{\sigma_1}{\sigma_0} + \frac{\mu}{\sigma_0}$$

6. $$SNR = 20 \log_{10} \left(\frac{\mu}{\sigma_0}\right)$$

7. $$\frac{\mu}{\sigma_0} = 10^{SNR/20}$$

Find SNR given P(FA) and P(D)

1. Calculate $U_{P(D)}$ from Formula 1.
2. Calculate $U_{P(D)} \frac{\sigma_1}{\sigma_0}$.
3. Calculate $U_{P(FA)}$ from Formula 2.
4. Calculate $\frac{\mu}{\sigma_0}$ from Formula 4 using relationship in Formula 3.
5. Calculate SNR from Formula 6.

Find P(FA) given P(D) and SNR

1. Calculate $U_{P(D)}$ from Formula 1.
2. Calculate $U_{P(D)} \frac{\sigma_1}{\sigma_0}$.
3. Calculate $\frac{\mu}{\sigma_0}$ from Formula 7.
4. Calculate $U_{P(FA)}$ from Formula 5.
5. Calculate P(FA) from Formula 2.

Find P(D) given P(FA) and SNR

1. Calculate $U_{P(FA)}$ from Formula 2.
2. Calculate $\frac{\mu}{\sigma_0}$ from Formula 7.
3. Calculate $\frac{X_{P(D)}}{\sigma_0} - \frac{\mu}{\sigma_0}$ using relationship in Formula 3.
4. Calculate $U_{P(D)}$ from Formula 4.
5. Calculate P(D) from Formula 1.

METHOD USED FOR INTEGRAL EVALUATION

$$P(U) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{U} e^{-t^2/2}\, dt$$

$$P(U) \approx 1 - Z(U)\,(a_1\, t + a_2\, t^2 + a_3\, t^3) + \epsilon(U)$$

$$Q(U) = \frac{1}{\sqrt{2\pi}} \int_{U}^{\infty} e^{-t^2/2}\, dt = 1 - P(U)$$

$$Q(U) \approx Z(U)\,(a_1\, t + a_2\, t^2 + a_3\, t^3) + \epsilon(U)$$

where:

$$Z(U) = \frac{1}{\sqrt{2\pi}}\, e^{-U^2/2}$$

$t = (1 + pU)^{-1}$

$p = .33267$

$a_1 = .4361836$

$a_2 = -.1201676$

$a_3 = .9372980$

$|\epsilon(U)| < 1 \times 10^{-5}$

METHOD USED FOR FINDING U_P WHEN THE PROBABILITY (P) IS GIVEN

$$P = Q(U_P) = \frac{1}{\sqrt{2\pi}} \int_{U_P}^{\infty} e^{-t^2/2}\, dt$$

$$U_P \approx t - \frac{c_0 + c_1 t + c_2 t^2}{1 + d_1 t + d_2 t^2 + d_3 t^3} + \epsilon(P)$$

where:

$$t = \left(\ln \frac{1}{P^2}\right)^{1/2}$$

$c_0 = 2.515517$

$c_1 = .802853$

$c_2 = .010328$

$d_1 = 1.432788$

$d_2 = .189269$

$d_3 = .001308$

$|\epsilon(P)| < 4.5 \times 10^{-4}$

Reference: *Handbook of Mathematical Functions,* Abramowitz and Stegun, National Bureau of Standards, 1964.

ROOTS OF A POLYNOMIAL

The general form of a polynomial of degree n is $P_n(x) = a_1x^n + a_2x^{n-1} + ... + a_nx + a_{n+1}$. This program uses the Lin-Bairstow method to find all roots, real and complex, of up to a 21st degree polynomial in one variable with real coefficients. This method is based on the fact that the polynomial can be represented as a product of quadratic and linear factors. An initial guess is made of a trial quadratic factor which, for purposes of convergence, is hopefully near the desired factor of the polynomial. The trial factor is divided into the polynomial and a remainder results which is generally non-zero, assuming the initial guess is not an exact factor. The trial factor is transformed into an "exact" factor of the polynomial by an iterative process which modifies the coefficients of the quadratic term to make the remainder less than or equal to ϵ (error factor). The resulting roots can be found by using the quadratic formula. The original polynomial can now be reduced by dividing out the quadratic factor. The remaining roots of the reduced polynomial are computed in the same manner as above, with the program using the exact quadratic factor which was just found as the new guess or allowing the user the option of entering a quadratic factor for the new guess. If, in the course of finding the root pairs by dividing out the quadratic factors, the reduced polynomial has degree = 2, the remaining roots are found using the quadratic formula; if degree = 1, the obvious root is computed.

GENERAL NOTES REGARDING PROGRAM OPERATION

1. An initial estimate of u = v = 0 is recommended in searching for **all** roots. In general, there is no advantage in using previously computed values of u and v.

2. A plot of f(x) may be used to find the approximate location of **real** roots. Then, Master Library program ML-08 may be used to find the exact location (within ϵ) of real roots. The order of f(x) may then be lowered by dividing f(x) by factors of the form $(x - R_i)$ where R_i are the real roots.

3. If f(x) has **multiple sets** of **equal** roots, Δu and Δv will appear to be converging but then may never get smaller than the stated ϵ. The solution to this pathological case is to define a larger ϵ and thus accept a less precise approximation of the equal roots.

4. Do not assume divergence of Δu, Δv simply because monotonic convergence does not occur. Δu and Δv will often oscillate with large amplitude prior to convergence.

Solid State Software		TI © 1979		
ROOTS OF A POLYNOMIAL				EE-09
		→ roots(EG)		
n	ε	→ roots(CG)	→ Reduce $P_n(x)$	Control

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select program		[2nd] [Pgm] 09	
2	Initialize		[SBR] [CLR]	0.
3	Repartition (TI-58 only)	6	[2nd] [Op] 17	0.59
4	Enter degree of polynomial	n[1]	[A]	n
5	Enter coefficient of x^n, repeat until all coefficients have been entered.	a_1[2] ⋮ a_{n+1}	[R/S]	(n−1) ⋮ −1.
6	Enter allowable error	ε	[B]	ε
7	Enter control digit	CTRL[3]	[E]	CTRL
8	Enter initial estimate of u, then enter initial estimate of v in display	u v	[x⪌t]	 v
9	Calculate first two roots		[2nd] [C′][4]	Degree[5, †]
10	Calculate discriminant of quadratic		[R/S]	Disc.[6, †]
11	Calculate 1st root		[R/S]	Root 1[6, †]
12	Calculate 2nd root		[R/S]	Root 2[6, †]
13	Execute exit logic		[R/S]	2.[7]
14	Reduce degree of polynomial by 2 by dividing P(x) by $x^2 - ux - v$		[D]	1.
15	Calculate next pair[8] of roots		[C][9]	Degree[5, †]
16	Repeat steps 10 through 15 until all roots have been found			

NOTES:

1. $2 \leqslant n \leqslant 21$
2. All coefficients must be entered in sequence starting with the coefficient of x^n. Missing terms have coefficients = 0.
3. CTRL = 1: PRINT successive values of Δu and Δv and final values of u and v.
   CTRL = 0: Suppress output of Δu and Δv and final values of u and v.
   CTRL = −1: DISPLAY (FLASHING) successive values of Δu and Δv (use [R/S] to clear flashing and proceed with calculations of next Δu and Δv and final values of u and v, and degree of polynomial).
   CTRL = 1 or −1 is recommended if user is uncertain of convergence.
4. [2nd] [C′] is used to calculate a pair of roots when the user desires to specify initial estimates of u and v.
5. See Note 3 describing output which may occur prior to display of Degree = n.
6. If discriminant <0, the following root calculations give the imaginary and real parts, respectively, of the complex conjugate roots. If discriminant ⩾0, the following roots are both real.
7. Exit logic performs initialization of flags.
8. If reduced polynomial is of first degree, only one root exists.
9. [C] is used to calculate roots using previously calculated values of u and v.

†These values are printed if printer is connected.

Example: Find all roots of

$$x^5 - 17.8x^4 + 99.41x^3 - 261.218x^2 + 352.611x - 134.106$$

Set error limit (ϵ) at 1×10^{-9}, use 0 as initial guess for u and v.

The roots are:
x = 3.619868415
x = 0.5801315846
x = 10.3
x = 1.65 ± j 1.86480562

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 09		Select program
	[SBR] [CLR]	0.	Initialize
6	[2nd] [Op] 17	0.59	Repartition (TI-58 only)
5	[A]	5.	Enter degree of polynomial
1	[R/S]	4.	Enter coefficient of x^5
17.8	[+/−] [R/S]	3.	Enter coefficient of x^4
99.41	[R/S]	2.	Enter coefficient of x^3
261.218	[+/−] [R/S]	1.	Enter coefficient of x^2
352.611	[R/S]	0.	Enter coefficient of x
134.106	[+/−] [R/S]	−1.	Enter constant term
9	[+/−] [INV] [2nd] [log] [B]	0.000000001	Enter ϵ
1	[E]	1.	Enter CTRL (output Δu & Δv on printer)
0	[x≷t]		Enter initial guess of u
0		0.	Enter initial guess of v
	[2nd] [C']		Find first two roots (see tape)
		5.†	
	[R/S]	9.24†	Print discriminant value
	[R/S]	3.619868415†	Print root 1 (real)
	[R/S]	.5801315846†	Print root 2 (real)
	[R/S]	2.	Execute exit logic
	[D]	1.	Divide P(x) by $x^2 - ux - v$
0	[E]	0.	Change CTRL to suppress output of Δu & Δv
	[C]	3.†	Find next two roots using current values of u and v
	[R/S]	−13.91†	Print discriminant value
	[R/S]	1.86480562†	Print imaginary part of roots 3 and 4
	[R/S]	1.65†	Print real part of roots 3 and 4
	[R/S]	2.	Execute exit logic
	[D]	1.	Divide P(x) by $x^2 - ux - v$
	[C]	1.†	Find final root
	[R/S]	10.3†	Print root 5 (real)
	[R/S]	10.3	Execute exit logic

†These values are printed if printer is connected.

2.374099678	Δu
2.69134463	Δv
1.909294944	Δu
.0146947731	Δv
-.1559301479	Δu
-3.23729809	Δv
.0289236121	Δu
-1.250626589	Δv
.0395701278	Δu
-.2990634176	Δv
.0040246833	Δu
-.0189731566	Δv
.0000171029	Δu
-.0000781489	Δv
.0000000003	Δu
-.0000000013	Δv
-.0000000001	Δu
4.3203682-11	Δv
4.2	u
-2.1	v
5.	Degree
9.24	Discriminant
3.619868415	Real 1
.5801315846	Real 2
3.	Degree
-13.91	Discriminant
1.86480562	Imaginary Part 3 and 4
1.65	Real Part 3 and 4
1.	Degree
10.3	Real 5

Register Contents

R_{00} a_{n+1} (constant term)
R_{01} a_n (coefficient of x term)

↓ ↓

R_{21} a_1 (coefficient of x^{21} term)
R_{22} Used
R_{23} b_1
R_{24} b_2

↓ ↓

R_{44} b_{22}
R_{45} c_{i-2}
R_{46} c_{i-1}
R_{47} c_i
R_{48} η
R_{49} ϵ
R_{50} Used
R_{51} Used
R_{52} u
R_{53} v
R_{54} Used
R_{55} Used
R_{56} Used
R_{57} Used
R_{58} Not used
R_{59} Used

Method Used

The following is an outline of the Lin-Bairstow method.

Given a polynomial of degree n: $P_n(x) = a_1 x^n + a_2 x^{n-1} + \ldots + a_n x + a_{n+1}$ and trial quadratic factor: $x^2 - ux - v$

$$\begin{aligned} P_n(x) &= (x^2 - ux - v)\, Q_{n-2}(x) + \text{remainder} \\ &= (x^2 - ux - v)(b_1 x^{n-2} + b_2 x^{n-3} + \ldots + b_{n-2} x + b_{n-1}) + b_n (x - u) + b_{n+1} \end{aligned} \qquad (1)$$

Multiplying out and equating coefficients of like powers of x:

$$\begin{aligned} a_1 &= b_1 \\ a_2 &= b_2 - ub_1 \\ a_3 &= b_3 - ub_2 - vb_1 \\ &\vdots \\ a_i &= b_i - ub_{i-1} - vb_{i-2} \\ &\text{for } i = 4, \ldots, n+1 \end{aligned} \qquad \text{or} \qquad \begin{aligned} b_1 &= a_1 \\ b_2 &= a_2 + ub_1 \\ b_3 &= a_3 + ub_2 + vb_1 \\ &\vdots \\ b_i &= a_i + ub_{i-1} + vb_{i-2} \\ &\text{for } i = 4, \ldots, n+1 \end{aligned}$$

Note: Remainder term from equation (1) is $b_n (x - u) + b_{n+1}$, and b_n, b_{n+1} both zero implies $(x^2 - ux - v)$ is an exact quadratic factor. Normally, this is not the case. Thus, values of u and v must be modified so that b_n and b_{n+1} both approach zero.

Since b_n and b_{n+1} are functions of parameters u and v, $b_n(u,v)$ and $b_{n+1}(u,v)$ can be expanded as a Taylor series for a function of two variables in terms of $(u^* - u) = \Delta u$ and $(v^* - v) = \Delta v$ where Δu, Δv are presumed small so that terms of higher order than the first are negligible.

$$b_n(u^*, v^*) = b_n(u,v) + \frac{\partial b_n}{\partial u}\Delta u + \frac{\partial b_n}{\partial v}\Delta v + \ldots$$

$$b_{n+1}(u^*, v^*) = b_{n+1}(u, v) + \frac{\partial b_{n+1}}{\partial u}\Delta u + \frac{\partial b_{n+1}}{\partial v}\Delta v + \ldots$$

Taking (u^*, v^*) as the point where the remainder is zero, (Δu and Δv being the increments added the original u and v to get the new values u^* and v^*),

$$b_n(u^*, v^*) = 0 \approx b_n + \frac{\partial b_n}{\partial u}\Delta u + \frac{\partial b_n}{\partial v}\Delta v \qquad (2)$$

$$b_{n+1}(u^*, v^*) = 0 \approx b_{n+1} + \frac{\partial b_{n+1}}{\partial u}\Delta u + \frac{\partial b_{n+1}}{\partial v}\Delta v \qquad (3)$$

The problem now is to solve equations (2) and (3) simultaneously for Δu and Δv. Bairstow showed that the required partial derivatives can be obtained from the b's in the same way that the b's were obtained from the a's.

Define a set of c's as follows:

$$c_1 = b_1 \qquad \frac{\partial b_1}{\partial u} = \frac{\partial a_1}{\partial u} = 0 \qquad \frac{\partial b_1}{\partial v} = \frac{\partial a_1}{\partial v} = 0$$

$$c_2 = b_2 + uc_1 \qquad \frac{\partial b_2}{\partial u} = \frac{\partial b_1}{\partial u} u + b_1 = b_1 = c_1 \qquad \frac{\partial b_2}{\partial v} = \frac{\partial a_2}{\partial v} + \frac{\partial b_1}{\partial v} u = 0$$

$$c_3 = b_3 + uc_2 + vc_1 \qquad \frac{\partial b_3}{\partial u} = \frac{\partial b_2}{\partial u} u + b_2 = c_2 \qquad \frac{\partial b_3}{\partial v} = \frac{\partial b_2}{\partial v} u + \frac{\partial b_1}{\partial v} v + b_1 = b_1 = c_1$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

The above three columns can be generalized by the following recurrence relation:

$$c_i = b_i + uc_{i-1} + vc_{i-2}$$
$$\text{for } i = 1, n$$

where $\frac{\partial b_i}{\partial u} = \frac{\partial b_{i+1}}{\partial v} = c_{i-1}$

Equations (2) and (3) can be rewritten as:

$$-b_n = c_{n-1}\,\Delta u + c_{n-2}\,\Delta v$$

$$-b_{n+1} = c_n\,\Delta u + c_{n-1}\,\Delta v$$

Then, using Cramer's rule

$$\Delta u = \frac{\begin{vmatrix} -b_n & c_{n-2} \\ -b_{n+1} & c_{n-1} \end{vmatrix}}{\begin{vmatrix} c_{n-1} & c_{n-2} \\ c_n & c_{n-1} \end{vmatrix}}$$

$$\Delta v = \frac{\begin{vmatrix} c_{n-1} & -b_n \\ c_n & -b_{n+1} \end{vmatrix}}{\begin{vmatrix} c_{n-1} & c_{n-2} \\ c_n & c_{n-1} \end{vmatrix}}$$

If denominator = 0, $\Delta u = 1 + \Delta u'$, $\Delta v = 1 + \Delta v'$ where $\Delta u'$, $\Delta v'$ are values of Δu, Δv computed from previous iteration.

If $|\Delta u| \leqslant \epsilon$ and $|\Delta v| \leqslant \epsilon$ where ϵ is the allowable error factor, then $(x^2 - ux - v)$ is a factor of P(x). Roots of $(x^2 - ux - v)$ are found by using the quadratic formula.

$$\frac{P_n(x)}{(x^2 - ux - v)} = Q_{n-2}(x)$$

, the reduced polynomial used to find the remaining roots.

Reference:

Applied Numerical Analysis, Curtis F. Gerald, Addison-Wesley, 1970.

CHAINED MULTIPLICATION OF POLYNOMIALS

This program performs the multiplication of two polynomials $P_1 \times P_2 = P_3$. The polynomials (P) are of the form $a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$ where the coefficients are real numbers. The resulting product P_3 may subsequently be multiplied by another polynomial. This chained multiplication process $(P_1 \times P_2 \times P_3 \times \ldots \times P_K) = P_{product}$ can be repeated until the limits of the calculator are reached. The degree of the final product polynomial must be $\leqslant 25$ for the TI-58 and $\leqslant 40$ for the TI-59. The intermediate products may be output, if desired.

Solid State Software				TI © 1979
CHAINED MULTIPLICATION OF POLYNOMIALS				EE-10
Exit			No Print	TI-58
Initialize	Multiply	→ Product	Print	TI-59

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select program		[2nd] [Pgm] 10	
2a	Printer used		[D]	
2b	Printer not used		[2nd] [D']	
3a	TI-59 used[1]		[E]	
3b	TI-58 used[1]		[2nd] [E']	
4	Initialize		[A]	0.[2]
5	Enter the coefficients of P_1 starting with a_0. Repeat for all coefficients.		[R/S]	1.[2]
6	Prepare to multiply		[B]	0.[2]
7	Enter the coefficients of P_2 starting with a_0. Repeat for all coefficients.[3,4]		[R/S]	1.[2]
8a	Output product with printer[5]		[C]	$a_0, a_1, \cdots$ are printed
8b	1. Output product without printer[5]		[C]	No. of coefficients in product
	2. Display a_0 (Repeat for $a_1, a_2, \cdots, a_n$)		[R/S]	$a_0, a_1, \cdots, a_n$
9	Exit the program[6]		[2nd] [A']	479.59 or 239.29

NOTES:
1. A TI-59 will be partitioned 239.89, for products of up to 40th degree.
   A TI-58 will be partitioned 0.59, for products of up to 25th degree.
2. The number in the display shows the power of x for the coefficient to be entered. If this power of x is missing in the polynomial, a 0 must be entered as its coefficient.
3. Wait until the next power of x is displayed (when actual multiplication is completed).
4. If another polynomial is to be multiplied, repeat steps 6 and 7.
5. Intermediate products may be output.
6. [2nd] [A'] must be used to return to standard partitioning and to reset flags.

Example: Multiply the three polynomials $(3 + 2x - 5x^2)(x)(4 - x^2)$ in order from left to right. Rewriting these polynomials to include zero coefficients, we have $(3 + 2x - 5x^2)(0 + x)$ $(4 + 0x + x^2)$.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 10		Select program	
	[D]		Assume printer	
	[E]		Assume TI-59	
	[A]	0.	Initialize	
3	[R/S]	1.	Enter constant term	
2	[R/S]	2.	Enter coefficient of x	
5	[+/–] [R/S]	3.	Enter coefficient of x^2	
	[B]	0.	Prepare to multiply	
0	[R/S]	1.	Enter constant term	
1	[R/S]	2.	Enter coefficient of x	
	[C]		Print intermediate product	0.
				3.
				2.
		–5.		–5.
	[B]	0.	Prepare to multiply	
4	[R/S]	1.	Enter constant term	
0	[R/S]	2.	Enter coefficient of x	
1	[+/–] [R/S]	3.	Enter coefficient of x^2	
	[C]		Print final product	0.
				12.
				8.
				–23.
				–2.
		5.		5.
	[2nd] [A']	479.59	Exit program	

Register Contents

R_{00} Pointer to product/input area
R_{01} Number of coefficients in product area
R_{02} Next power of x to be input
R_{03} Used
R_{04} Used
R_{05} Used
R_{06} Used
R_{07} New Coefficient

R_{08} ↓ R_{48}	Coefficients of Product	TI-59
R_{49} ↓ R_{89}	Coefficients of Multiplicand	TI-59
R_{08} ↓ R_{33}	Coefficients of Product	TI-58
R_{34} ↓ R_{59}	Coefficients of Multiplicand	TI-58

REACTANCE CHART

This program simulates a reactance chart by computing the capacitance C in farads, capacitive reactance X_C in ohms, inductance L in henrys and the inductive reactance X_L in ohms at an applied frequency f in hertz. The equations used are:

$$X_C = \frac{1}{\omega C} \qquad C = \frac{1}{\omega X_C} \qquad X_L = \omega L \qquad L = \frac{X_L}{\omega} \qquad \omega = 2\pi f$$

$$f = \frac{1}{2\pi X_C C} = \frac{X_L}{2\pi L}$$

If $f = f_0$, where f_0 is the frequency of resonance ($X_C = X_L$), the following relationships apply:

$$f_0 = \frac{1}{2\pi\sqrt{LC}} \qquad C = \frac{1}{\omega_0^2 L} \qquad L = \frac{1}{\omega_0^2 C}$$

Given any two of the elements in the sets $\{f, X_C, C\}$, $\{f, X_L, L\}$, or $\{f_0, L, C\}$ which are comprised of the quantities used in the defining equations for frequency, this program computes the remaining element:

Given	Compute
C, f	X_C
X_C, f	C
L, f	X_L
X_L, f	L
C, X_C	f
L, X_L	f
C, f_0	L
L, f_0	C
C, L	f_0

Solid State Software				TI © 1979
REACTANCE CHART				EE-11
f_0	X_L	X_C		
f	L	C		→ Execute

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select program		[2nd] [Pgm] 11	
2	Initialize		[SBR] [CLR] [2nd] [Eng]	0
3	Enter any two of the following:			
	a. Applied frequency or Resonant frequency in hertz	f f_0	[A] [2nd] [A']	f f_0
	b. Inductance in henrys	L	[B]	L
	c. Inductive reactance in ohms	X_L	[2nd] [B']	X_L
	d. Capacitance in farads	C	[C]	C
	e. Capacitive reactance in ohms	X_C	[2nd] [C']	X_C
4	Compute remaining quantity in the frequency equation[1]			
	a. Applied frequency or Resonant frequency in hertz		[E] [A] [E] [2nd] [A']	f f_0
	b. Inductance in henrys		[E] [B]	L
	c. Inductive reactance in ohms		[E] [2nd] [B']	X_L
	d. Capacitance in farads		[E] [C]	C
	e. Capacitive reactance in ohms		[E] [2nd] [C']	X_C

NOTES: 1. Use computation chart as a guide.

Example 1: What is the reactance of 15 picofarads at 27 megahertz? What is the reactance of 10 microhenrys at 27 megahertz?

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 11		Select program
	[SBR] [CLR] [2nd] [Eng]*	0. 00	Initialize
15	[EE] 12 [+/−] [C]	15.−12	Enter C in farads
27	[EE] 6 [A]	27. 06	Enter f in hertz
	[E] [2nd] [C']	392.97517 00	Compute X_C in ohms
10	[EE] 6 [+/−] [B]	10.−06	Enter L in henrys
	[E] [2nd] [B']	1.69646 03	Compute X_L in ohms
	[INV] [2nd] [Eng]*	1696.460033	Remove engineering mode

Example 2: With L = 10 microhenrys, what value of C is required to resonate at 27 megahertz? With C = 15 picofarads, what value of L is required to resonate at 27 megahertz?

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 11		Select program
	[SBR] [CLR] [2nd] [Eng]*	0. 00	Initialize
27	[EE] 6 [2nd] [A']	27. 06	Enter f_0 in hertz
10	[EE] 6 [+/−] [B]	10.−06	Enter L in henrys
	[E] [C]	3.4746634−12	Compute C in farads
15	[EE] 12 [+/−] [C]	15.−12	Enter C in farads
	[E] [B]	2.3164422−06	Compute L in henrys
	[INV] [2nd] [Eng]*	.0000023164	Remove engineering mode

Example 3: With C = 15 picofarads and L = 10 microhenrys, what is the resonant frequency? If C = 17 picofarads and X_C = 400 ohms, what is the applied frequency, f?

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 11		Select program
	[SBR] [CLR] [2nd] [Eng]*	0. 00	Initialize
15	[EE] 12 [+/−] [C]	15.−12	Enter C in farads
10	[EE] 6 [+/−] [B]	10.−06	Enter L in henrys
	[E] [2nd] [A']	12.994947 06	Compute f_0 in hertz
17	[EE] 12 [+/−] [C]	17.−12	Enter C in farads
400	[2nd] [C']	400. 00	Enter X_C in ohms
	[E] [A]	23.405139 06	Compute f in hertz
	[INV] [2nd] [Eng]*	23405138.69	Remove engineering mode

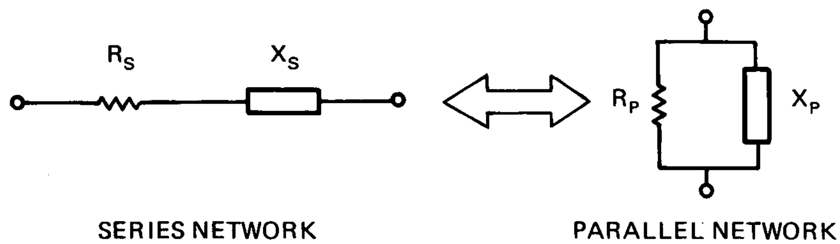
Register Contents

R_{00}	R_{05}	R_{10} C	R_{15}	R_{20}	R_{25}
R_{01}	R_{06}	R_{11} L	R_{16}	R_{21}	R_{26}
R_{02}	R_{07}	R_{12} X_C	R_{17}	R_{22}	R_{27}
R_{03}	R_{08}	R_{13} X_L	R_{18}	R_{23}	R_{28}
R_{04}	R_{09}	R_{14} $f \rightarrow \omega$	R_{19}	R_{24}	R_{29}

*These examples show optional use of the engineering mode. Be sure to remove engineering mode when computations are completed.

SERIES/PARALLEL IMPEDANCE CONVERSIONS

This program converts a two-element series impedance network to an equivalent two-element parallel impedance network, or vice versa. The conversion is illustrated by the following diagram where R_S, X_S, R_P, and X_P refer to the series resistance, series reactance, parallel resistance, and parallel reactance in ohms, respectively.



Any two-element impedance network can be represented equivalently as a series network or a parallel network.

The transformation equations are:

Given R_S and X_S

$$R_P = R_S + \frac{X_S^2}{R_S}$$

$$X_P = \frac{R_P R_S}{X_S}$$

Given R_P and X_P

$$R_S = \left(\frac{1}{R_P} + \frac{R_P}{X_P^2}\right)^{-1}$$

$$X_S = \frac{R_S R_P}{X_P}$$

Solid State Software				TI © 1979
SERIES/PARALLEL IMPEDANCE CONVERSIONS EE-12				
R_S	X_S	R_P	X_P	→ Execute

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select program		[2nd] [Pgm] 12	
2	Initialize		[SBR] [CLR]	
	Do either Step 3 or Step 4			
3	Enter the following in any order			
	a. Series resistance in ohms	R_S	[A]	R_S
	b. Series reactance in ohms	X_S	[B]	X_S
	Compute the following in any order			
	c. Parallel resistance in ohms		[E] [C]	R_P
	d. Parallel reactance in ohms		[E] [D]	X_P
4	Enter the following in any order			
	a. Parallel resistance in ohms	R_P	[C]	R_P
	b. Parallel reactance in ohms	X_P	[D]	X_P
	Compute the following in any order			
	c. Series resistance in ohms		[E] [A]	R_S
	d. Series reactance in ohms		[E] [B]	X_S

Example: Using an RX impedance meter, R_P of 75 ohms and C_P of 25 picofarads is measured at 125 megahertz. In order to solve our circuit application for purposes of power dissipation, an equivalent R_S and X_S would be easier to work with. Program EE-11 is used to solve for X_P before making the conversion.

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 11		Select program
	[SBR] [CLR] [2nd] [Eng]*	0. 00	Initialize
125	[EE] 6 [A]	125. 06	Enter frequency in hertz
25	[EE] 12 [+/−] [C]	25. −12	Enter capacitance in farads
	[E] [2nd] [C']	50.929582 00	Compute X_P in ohms
	[2nd] [Pgm] 12	50.929582 00	Select program
	[SBR] [CLR]	0. 00	Initialize
75	[C]	75. 00	Enter R_P in ohms
50.929582	[D]	50.929582 00	Enter X_P in ohms
	[E] [A]	23.669653 00	Compute R_S in ohms
	[E] [B]	34.856441 00	Compute X_S in ohms

At this point X_S can be converted to C_S using program EE-11.

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 11	34.856441 00	Select program
	[SBR] [CLR]	0. 00	Initialize
125	[EE] 6 [A]	125. 06	Enter frequency
34.856441	[2nd] [C']	34.856441 00	Enter X_S
	[E] [C]	36.528099 −12	Compute C_S

The results can be checked by converting R_S and X_S back to R_P and X_P

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 12	36.528099 −12	Select program
	[SBR] [CLR]	0. 00	Initialize
23.669653	[A]	23.669653 00	Enter R_S
34.856441	[B]	34.856441 00	Enter X_S
	[E] [C]	74.999999 00	Compute R_P
	[E] [D]	50.929581 00	Compute X_P
	[INV] [2nd] [Eng]*	50.929581	Remove engineering mode

Register Contents

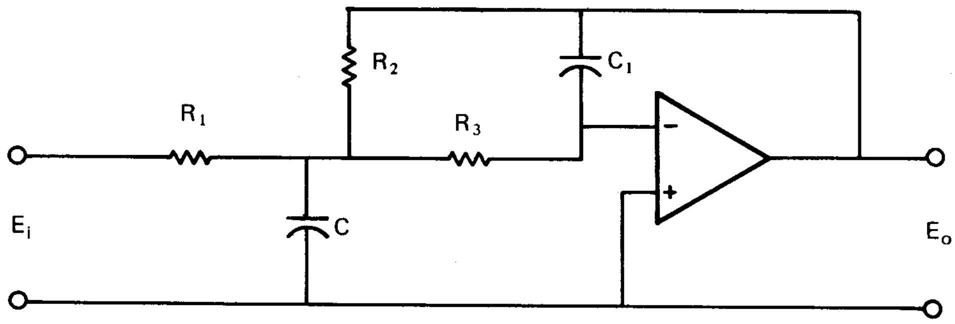
R_{00}	R_{05}	R_{10} R_S	R_{15}	R_{20}	R_{25}
R_{01}	R_{06}	R_{11} X_S	R_{16}	R_{21}	R_{26}
R_{02}	R_{07}	R_{12} R_P	R_{17}	R_{22}	R_{27}
R_{03}	R_{08}	R_{13} X_P	R_{18}	R_{23}	R_{28}
R_{04}	R_{09}	R_{14}	R_{19}	R_{24}	R_{29}

*This example shows optional use of engineering mode. Be sure to remove engineering mode when computations are completed.

ACTIVE LP, HP, BP FILTERS

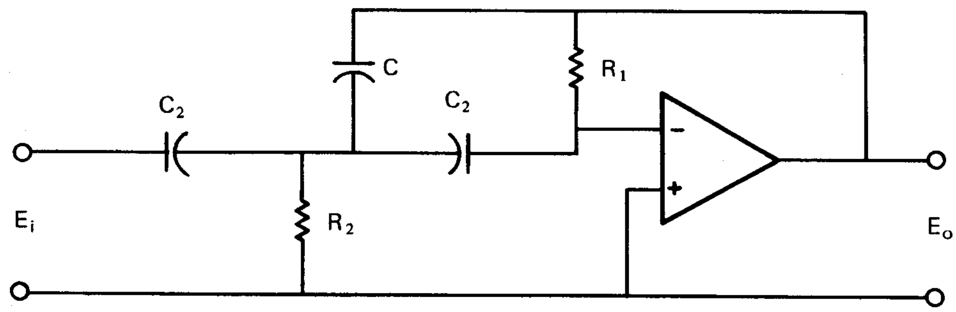
This program computes the necessary component values for use in the design of active lowpass, highpass, and bandpass filters as illustrated.

Active Lowpass Filter



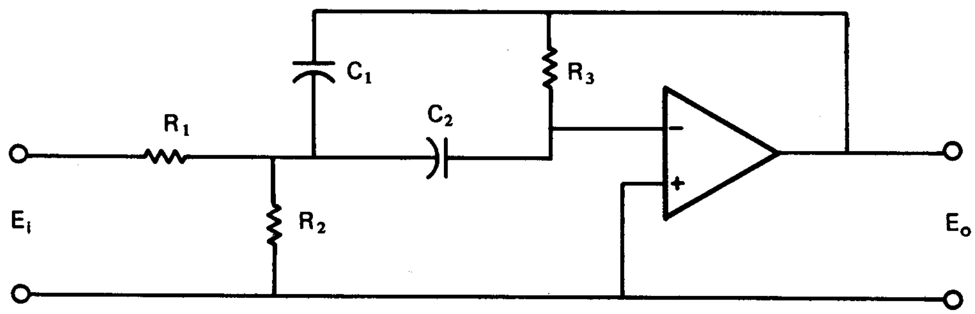
Given the peaking factor, the passband voltage gain in dB, the cutoff frequency in Hz, and C_1 in farads, the program calculates values of C, R_1, R_2, and R_3.

Active Highpass Filter



Given the peaking factor, the passband voltage gain in dB, the cutoff frequency in Hz, and C_2 in farads, the program calculates values for C, R_1, and R_2.

Active Bandpass Filter



Given the 3-dB bandwidth in Hz, the midband voltage gain in dB, the center frequency in Hz, and C_1 and C_2 in farads, the program computes values for R_1, R_2, and R_3.

Solid State Software			TI © 1979	
ACTIVE LP,HP,BP FILTERS				EE-13
C_1	C_2	→ BP	→ LP	→ HP
α	A	F	B	

USER INSTRUCTIONS

STEP	PROCEDURE		ENTER	PRESS	DISPLAY
1	Select program			[2nd] [Pgm] 13	
2	Initialize			[SBR] [CLR]	0.
3	Enter the following as needed:				
	a. peaking factor (LP, HP)[1]		α	[A]	α†
	b. passpand voltage gain (LP, HP) or midband voltage gain (BP) in dB[2]		A	[B]	A†
	c. cutoff frequency (LP, HP) or center frequency (BP) in hertz[1]		F	[C]	F†
	d. 3-dB bandwidth (BP) in hertz[1]		B	[D]	B†
	e. C_1 for (BP, LP) in farads[1]		C_1	[2nd] [A']	C_1†
	f. C_2 for (BP, HP) in farads[1]		C_2	[2nd] [B']	C_2†
4	Compute component values:				
	a. Bandpass[3,5]	R_1 in ohms		[2nd] [C']	R_1†
		R_2 in ohms		[R/S]	R_2†
		R_3 in ohms		[R/S]	R_3†
	b. Lowpass[4,5]	C in farads		[2nd] [D']	C†
		R_2 in ohms		[R/S]	R_2†
		R_1 in ohms		[R/S]	R_1†
		R_3 in ohms		[R/S]	R_3†
	c. Highpass[4]	C in farads		[2nd] [E']	C†
		R_1 in ohms		[R/S]	R_1†
		R_2 in ohms		[R/S]	R_2†

NOTES:

1. Input value must be greater than zero.
2. Because operational amplifiers are non-ideal, A should be chosen to insure H_0 is less than 10 when α is about 0.1. H_0 may be increased if α is also increased. A maximum H_0 of 100 is acceptable with $\alpha = 1$ for an operational amplifier with an open-loop gain of at least 80 dB.
3. If $Q \leqslant (H_0/2)^{1/2}$, program execution stops with a flashing nines display. To insure that all resistances are greater than zero, either $Q = F/B$, or H_0 should be adjusted so that $Q > (H_0/2)^{1/2}$. See steps 3b, 3c, and 3d.
4. A 12-dB per octave roll-off exists in the lowpass, highpass cases.
5. The value of R_1 calculated includes source resistance, i.e., actual R_1 = Calculated R_1 − Source R.

†These values are printed if printer is connected.

Example 1: Find the component values for an active bandpass filter with a center frequency of 150 hertz, a midband voltage gain of 30 dB, a 3-dB bandwidth of 16 hertz and $C_1 = C_2 = 0.1 \mu F$.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 13		Select program	
	[SBR] [CLR]	0.	Initialize	
16	[D]	16.	Enter 3-dB bandwidth	16.
30	[B]	30.	Enter midband voltage gain	30.
150	[C]	150.	Enter center frequency	150.
.0000001	[2nd] [A']	0.0000001	Enter C_1	0.0000001
.0000001	[2nd] [B']	0.0000001	Enter C_2	0.0000001
	[2nd] [C']	3145.575757	R_1	3145.575757 R1
	[R/S]	690.017292	R_2	690.017292 R2
	[R/S]	198943.6789	R_3	198943.6789 R3

Example 2: Find the component values for an active highpass filter with cutoff frequency = 400 hertz, passband voltage gain = 6 dB, peaking factor = 0.5 and $C_2 = 0.047 \mu F$.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 13		Select program	
	[SBR] [CLR]	0.	Initialize	
.5	[A]	0.5	Enter peaking factor	0.5
6	[B]	6.	Enter passband gain	6.
400	[C]	400.	Enter cutoff frequency	400.
.000000047	[2nd] [B']	0.000000047	Enter C_2	0.000000047
	[2nd] [E']	.0000000236	C	.0000000236 C
	[R/S]	84496.45356	R_1	84496.45356 R1
	[R/S]	1692.334014	R_2	1692.334014 R2

Example 3: Find the component values for an active lowpass filter with peaking factor $= \sqrt{2}$, passband gain = 20 dB, cutoff frequency = 1000 hertz and $C_1 = 0.02 \mu F$.

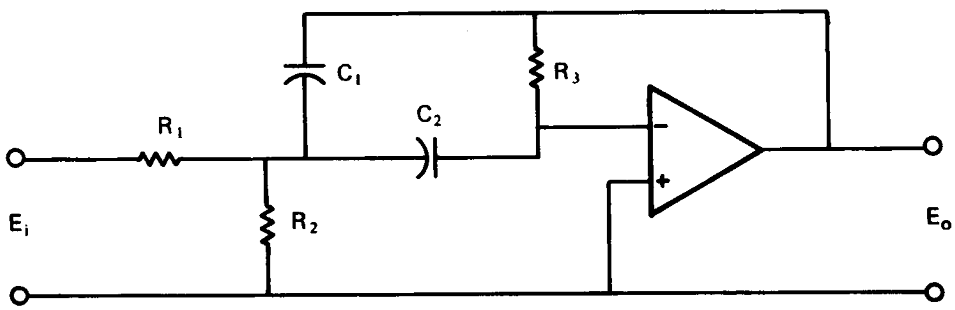
ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 13		Select program	
	[SBR] [CLR]	0.	Initialize	
2	[$\sqrt{x}$] [A]	1.414213562	Enter peaking factor	1.414213562
20	[B]	20.	Enter passband gain	20.
1000	[C]	1000.	Enter cutoff frequency	1000.
.00000002	[2nd] [A']	0.00000002	Enter C_1	0.00000002
	[2nd] [D']	0.00000044	C	0.00000044 C
	[R/S]	5626.976976	R_2	5626.976976 R2
	[R/S]	562.6976976	R_1	562.6976976 R1
	[R/S]	511.5433615	R_3	511.5433615 R3

Register Contents

R_{00}	R_{05}	R_{10} α	R_{15} Q	R_{20}	R_{25}
R_{01}	R_{06}	R_{11} $10^{A/20}$	R_{16} B	R_{21}	R_{26}
R_{02}	R_{07}	R_{12} F	R_{17} ω_0	R_{22}	R_{27}
R_{03}	R_{08}	R_{13} C_1	R_{18} R_2	R_{23}	R_{28}
R_{04}	R_{09}	R_{14} C_2	R_{19}	R_{24}	R_{29}

Method Used

ACTIVE BANDPASS FILTER



The voltage transfer function is:

$$\frac{E_o}{E_I}(s) = \frac{-(1/R_1C_1)s}{s^2 + \left(\frac{C_1 + C_2}{R_3C_1C_2}\right)s + \frac{R_1 + R_2}{R_1R_2R_3C_1C_2}}$$

and the corresponding bandpass network function is:

$$H(s) = \frac{-H_0\alpha\omega_0 s}{s^2 + \alpha\omega_0 s + \omega_0^2}$$

where

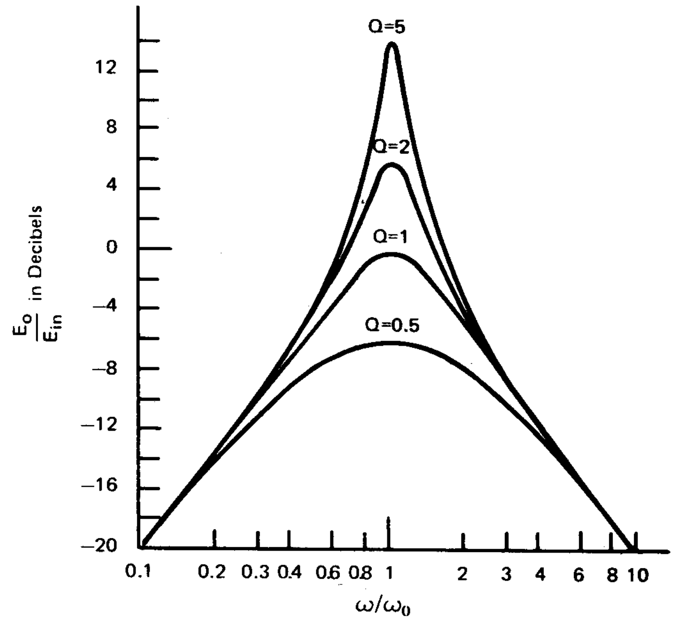
$H_0 = 10^{A/20}$, A = midband voltage gain in dB
$\alpha = 1/Q$
Q = F/B, quality factor measure of selectivity of filter
F = Center frequency of passband in hertz
B = 3-dB bandwidth in hertz
$\omega_0 = 2\pi F$

Given B, A, F, C_1 and C_2 in farads, the program calculates R_1, R_2 and R_3 using the following equations:

$$R_1 = \frac{Q}{2\pi F H_0 C_1}$$

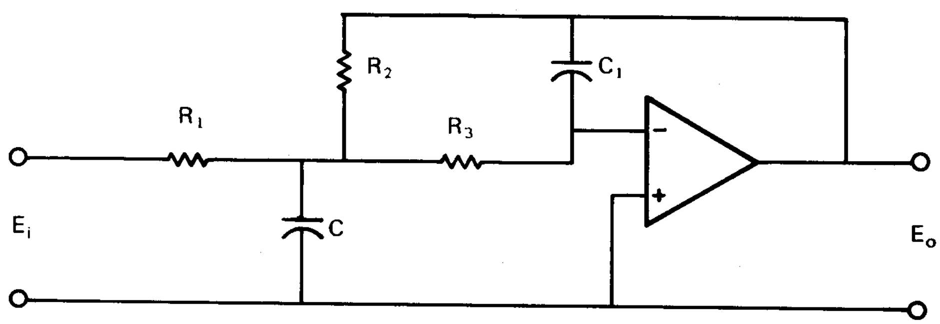
$$R_2 = [Q(C_1 + C_2)2\pi F - (1/R_1)]^{-1}$$

$$R_3 = \frac{Q}{2\pi F}\left(\frac{1}{C_1} + \frac{1}{C_2}\right)$$



Magnitude response of second order bandpass filter showing effect of Q on network voltage transfer function.

ACTIVE LOWPASS FILTER



The voltage transfer function is:

$$\frac{E_o}{E_i}(s) = \frac{-1/R_1 R_3 C C_1}{s^2 + \left(\frac{1}{R_1} + \frac{1}{R_3} + \frac{1}{R_2}\right)\left(\frac{1}{C}\right)s + \frac{1}{R_2 R_3 C C_1}}$$

The corresponding lowpass network function is:

$$H(s) = \frac{-H_0 \omega_0^2}{s^2 + \alpha\omega_0 s + \omega_0^2}$$

where

$H_0 = 10^{A/20}$, A = passband gain in dB

$\omega_0 = 2\pi F$

F = cutoff frequency in hertz

α = peaking factor = 2ζ

ζ = damping ratio

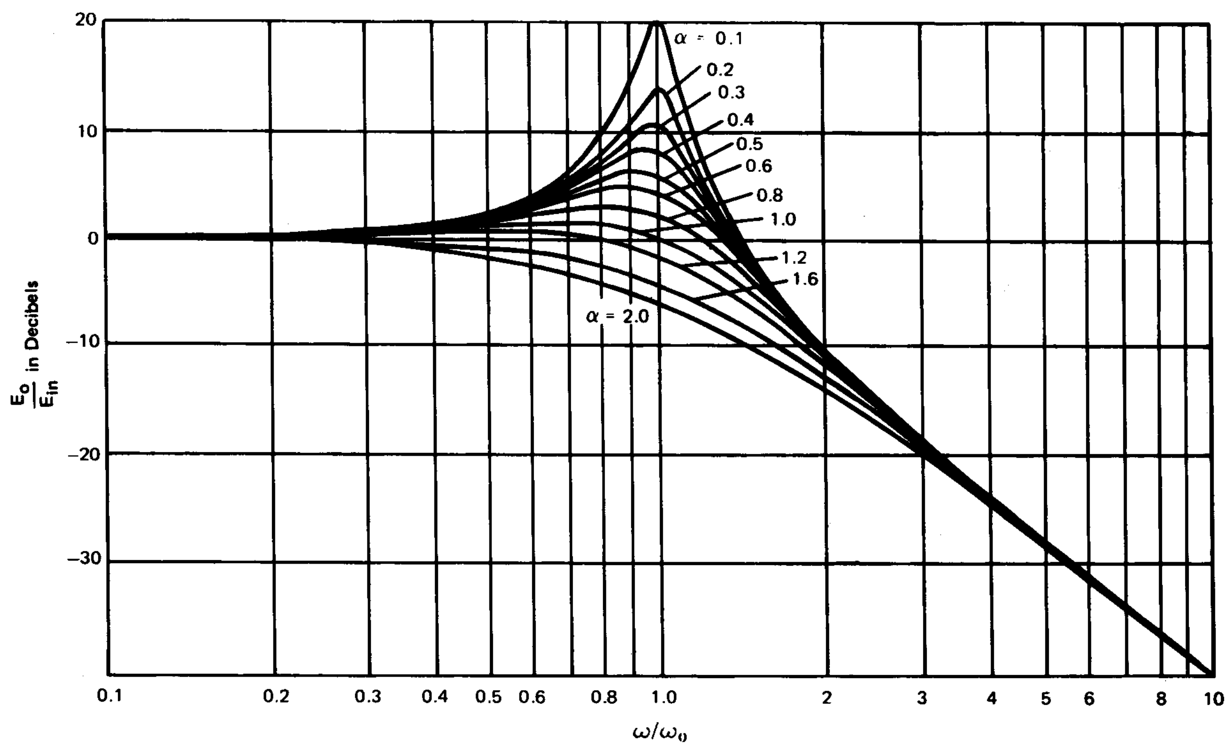
Given A, F, α and C_1 in farads, the program calculates R_1, R_2, R_3 and C using the following equations:

$$C = \frac{4(1 + H_0)C_1}{\alpha^2}$$

$$R_1 = \frac{R_2}{H_0}$$

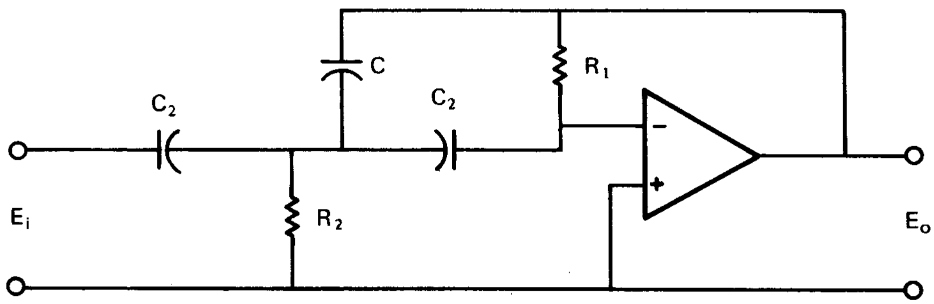
$$R_2 = \frac{\alpha}{4\pi F C_1}$$

$$R_3 = \frac{R_2}{H_0 + 1}$$



Magnitude response of second-order lowpass filters for several values of $\alpha \leqslant 2$.

ACTIVE HIGHPASS FILTER



The voltage transfer function is:

$$\frac{E_o}{E_i}(s) = \frac{-(C_2/C)s^2}{s^2 + \left(\frac{2}{C}+\frac{1}{C_2}\right)\left(\frac{1}{R_1}\right)s + \frac{1}{R_1 R_2 C_2 C}}$$

and the corresponding highpass network function is:

$$H(s) = \frac{-H_0 s^2}{s^2 + \alpha\omega_0 s + \omega_0^2}$$

where

$H_0 = 10^{A/20}$, A = passband gain in dB
$\omega_0 = 2\pi F$
F = cutoff frequency in hertz
α = peaking factor = 2ζ
ζ = damping ratio

Given A, F, α and C_2, the capacitance in farads, the program calculates R_1, R_2 and C using the following equations:

$$C = \frac{C_2}{H_0}$$

$$R_1 = \frac{2H_0 + 1}{2\pi F \alpha C_2}$$

$$R_2 = \frac{\alpha}{2\pi F C_2 (2 + 1/H_0)}$$

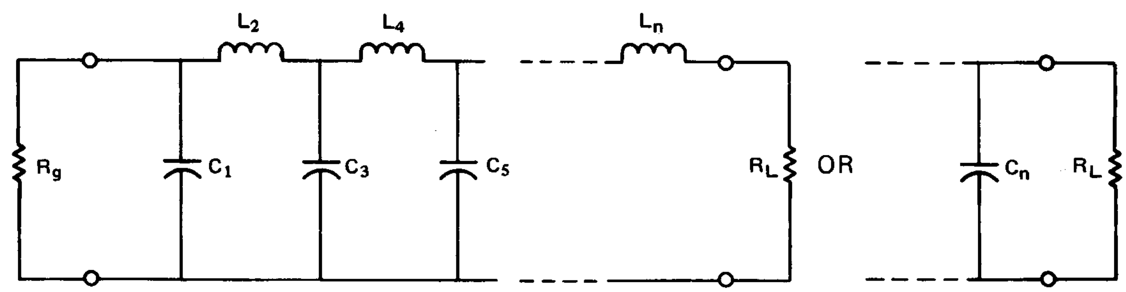
References: *Operational Amplifiers Design and Applications,* Graeme, Tobey and Huelsman, McGraw-Hill, 1971.

Theory and Design of Active RC Circuits, Huelsman, McGraw-Hill, 1968

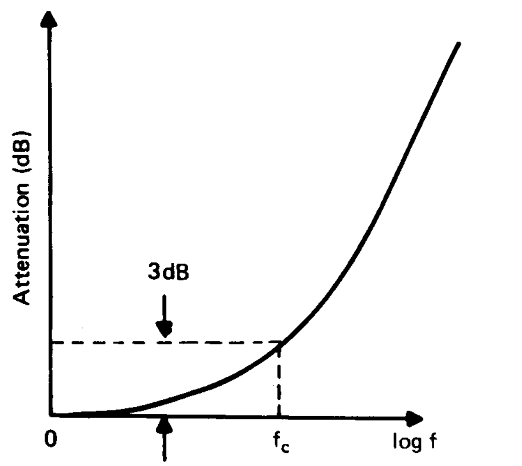
Electronics, pp 86-89, June 7, 1971

PASSIVE LOWPASS FILTERS

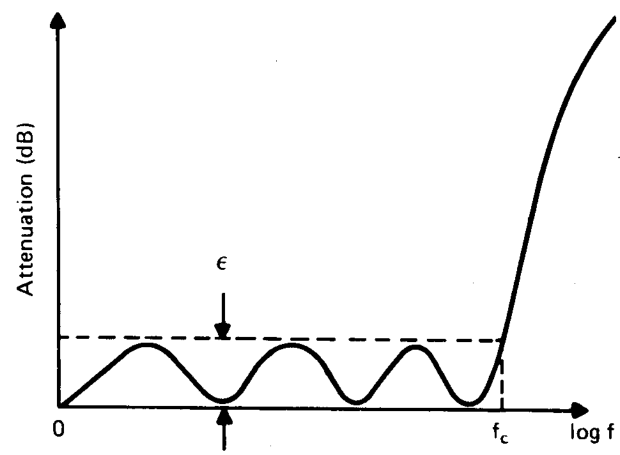
This program computes component values for passive lowpass Butterworth and Tchebycheff filters of the following form.



Required inputs are the filter order n (n, an integer ≥1) which defines the number of reactive components, allowable ripple ϵ in dB (ϵ = 0 for Butterworth filters), terminal resistance R in ohms, and the cutoff frequency f_c in hertz.



BUTTERWORTH LOWPASS CHARACTERISTIC



TCHEBYCHEFF LOWPASS CHARACTERISTIC

For Butterworth filters with response of the form shown, the component values are computed as follows:

$$C_i = \frac{1}{\pi f_c R} \sin\left[\frac{(2i-1)\pi}{2n}\right] \qquad i = 1, 3, 5, \ldots$$

$$L_i = \frac{R}{\pi f_c} \sin\left[\frac{(2i-1)\pi}{2n}\right] \qquad i = 2, 4, 6, \ldots$$

For Tchebycheff filters with response of the form shown, the component values are computed as follows:

$$\beta = \ln\left[\coth\left(\frac{\epsilon}{40 \log e}\right)\right]$$

$$\gamma = \sinh\left(\frac{\beta}{2n}\right)$$

$$a_i = \sin\left[\frac{(2i-1)\pi}{2n}\right] \qquad i = 1, 2, 3, \cdots, n$$

$$b_i = \gamma^2 + \sin^2\left(\frac{i\pi}{n}\right) \qquad i = 1, 2, 3, \cdots, n$$

$$g_1 = \frac{2a_1}{\gamma}$$

$$g_i = \frac{4a_{i-1}\, a_i}{b_{i-1}\, g_{i-1}} \qquad i = 2, 3, 4, \cdots, n$$

$$C_i = \frac{g_i}{2\pi f_c R} \qquad i = 1, 3, 5, \cdots$$

$$L_i = \frac{Rg_i}{2\pi f_c} \qquad i = 2, 4, 6, \cdots$$

NOTE: This program assumes that the generator resistance Rg is equal to the load resistance R_L for Butterworth filters and odd-order Tchebycheff filters. Thus, the termination resistance $R = Rg = R_L$. In the case of even-order Tchebycheff filters, $R_L = Rg/\coth^2(\beta/4)$ which, for small ripple results in the generator resistance Rg being approximately equal to the load resistance R_L. Thus, the termination resistance $R = Rg \approx R_L$.

References: *Handbook of Filter Synthesis,* Zverev, John Wiley and Sons, 1967

Microwave Filters, Impedance-Matching Networks, and Coupling Structures, Matthai, Young, Jones, McGraw-Hill, 1964

Solid State Software				TI © 1979
PASSIVE LOWPASS FILTERS				EE-14
				→ No Print
n	ϵ	R	f_c	→ Print

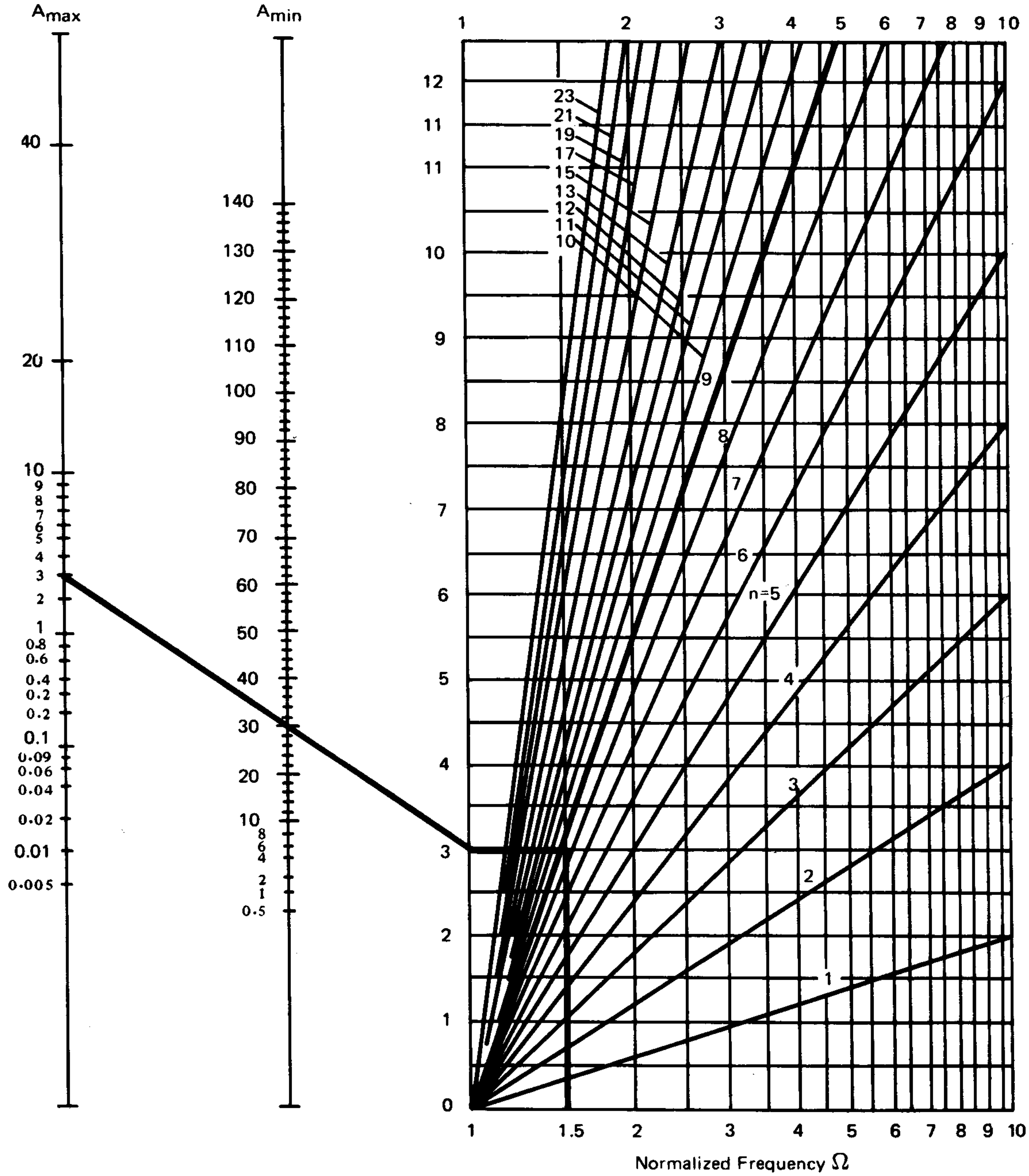
USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select program		[2nd] [Pgm] 14	
2	Initialize		[SBR] [CLR]	0.
3	Enter the following in any order:			
	a. Order of the filter[1]	n	[A]	n
	b. Allowable ripple in dB[3]	ϵ	[B]	ϵ
	c. Termination resistance in ohms[2]	R	[C]	R
	d. Cutoff frequency in hertz[2]	f_c	[D]	f_c
4	Compute and display component values. C is in farads. L is in henrys.		[2nd] [E']	"1."[4,5]
				C_1
	OR		[R/S]	"2."
				L_2
			[R/S]	"3."
				C_3
				⋮
5	Compute and print component values.		[E]	1.[4,6†]
				C_1†
				2.†
				L_2†
				3.†
				C_3†
				⋮

NOTES:
1. Input must be an integer ⩾ 1.
2. Input > 0.
3. Input ⩾ 0. For Butterworth, enter 0.
4. If ϵ < 0, display will flash.
5. " " indicates that the component number is displayed for about 2.5 seconds.
6. Last filter component value remains in the display.

†These values are printed only.

Example 1: Compute the component values for a Butterworth filter ($\epsilon = 0$), having equal generator and load resistances of 2000 ohms (R). The filter must pass frequencies between 0 and 10 kilohertz (f_c), the maximum attenuation at 10 kilohertz being 3 dB. At least 30 dB attenuation must be provided at 15 kilohertz and above. The order of the filter can be determined by using a nomograph as follows.



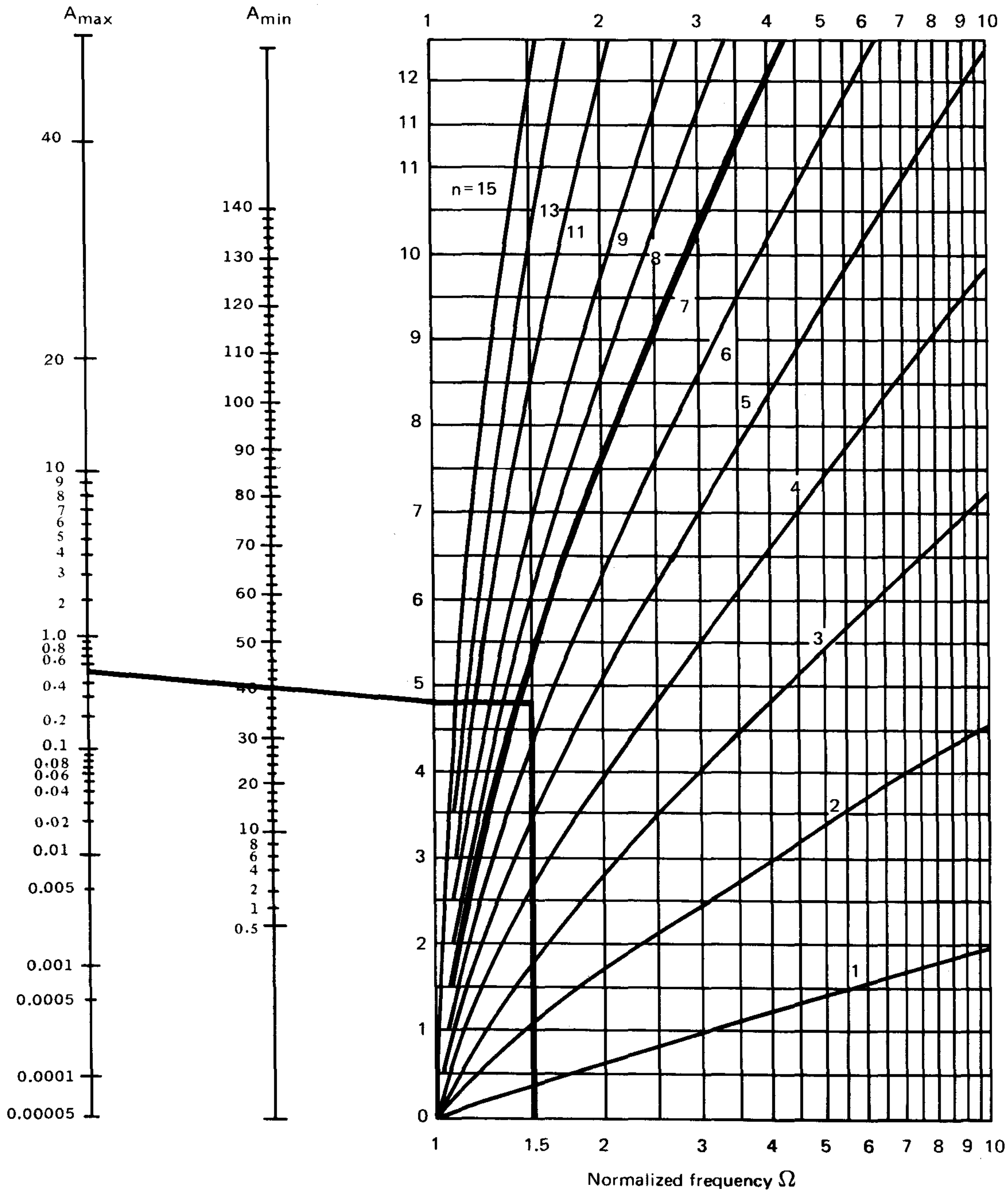
NOMOGRAPH FOR BUTTERWORTH FILTERS

A_{max} = 3 dB attenuation of passband
A_{min} = 30 dB attenuation of stopband
Ω = (15 kilohertz)/(10 kilohertz) = 1.5

⇒ at least a 9th order filter must be chosen.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 14		Select program	
	[SBR] [CLR]	0.	Initialize	
9	[A]	9.	Enter filter order	
0	[B]	0.	Enter ripple	
2000	[C]	2000.	Enter termination R	
10000	[D]	10000.	Enter f_c	
	[E]		Compute and print	1.
			Component values C_1	.0000000028
				2.
			Component values L_2	.0318309886
				3.
			Component values C_3	.0000000122
				4.
			Component values L_4	.0598226902
				5.
			Component values C_5	.0000000159
				6.
			Component values L_6	.0598226902
				7.
			Component values C_7	.0000000122
				8.
			Component values L_8	.0318309886
				9.
		.0000000028	Component values C_9	.0000000028

Example 2: Compute the component values for a Tchebycheff filter having equal generator and load resistances of 1000 ohms (R). The filters passband must range from 0 to 3500 hertz (f_c) with an attenuation of 40 dB required at 5250 hertz. The passband ripple is allowed to be 0.5 dB with 3500 hertz being the cutoff frequency. Again a nomograph is used to determine the order of the filter.



NOMOGRAPH FOR TCHEBYCHEFF FILTERS

A_{max} = .5 dB passband ripple

A_{min} = 40 dB attenuation of stopband

$$\Omega = \frac{5250 \text{ hertz}}{3500 \text{ hertz}} = 1.5$$

⇒ at least a 7th **order filter** must be chosen.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 14		Select program	
	[SBR] [CLR]	0.	Initialize	
7	[A]	7.	Enter filter order	
.5	[B]	0.5	Enter ripple	
1000	[C]	1000.	Enter termination R	
3500	[D]	3500.	Enter f_c	
	[2nd] [E']	"1."	Compute component values	
		0.000000079	C_1	
	[R/S]	"2."		
		.0572155871	L_2	
	[R/S]	"3."		
		0.00000012	C_3	
	[R/S]	"4."		
		0.06113069	L_4	
	[R/S]	"5."		
		0.00000012	C_5	
	[R/S]	"6."		
		.0572155871	L_6	
		"7."		
		0.000000079	C_7	

" " Indicate value is displayed for approximately 2.5 seconds.

Register Contents

R_{00} Dsz index	R_{05} b_{K-1}	R_{10} n	R_{15}	R_{20}	R_{25}
R_{01} K (Used)	R_{06} g_{K-1}	R_{11} ϵ	R_{16}	R_{21}	R_{26}
R_{02} Used (γ)	R_{07} g_K	R_{12} R	R_{17}	R_{22}	R_{27}
R_{03} a_{K-1}	R_{08}	R_{13} f_c	R_{18}	R_{23}	R_{28}
R_{04} a_K	R_{09}	R_{14}	R_{19}	R_{24}	R_{29}

CONVOLUTION

Given the impulse response for a linear system, one can use the convolution integral to find the system's output for a specified input waveform. The program generates outputs at intervals of Δt using the trapezoidal rule.

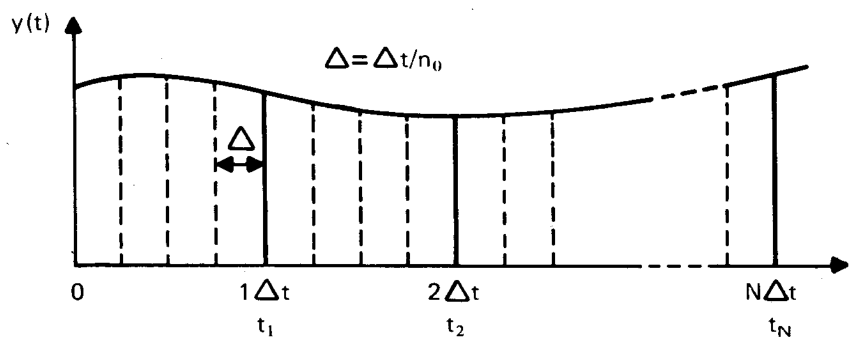
To calculate the output y(t) of a linear system, the convolution integral integrates the product of the input x(t) and the impulse response h(t) in the form:

$$y(t) = \int_0^t x(\lambda)h(t-\lambda)\,d\lambda$$

Applying the trapezoidal rule to this integral yields:

$$y(t) \approx \frac{\Delta}{2}\left[x(0)h(t) + 2\sum_{k=1}^{n-1} x(k\Delta)h(t-k\Delta) + x(t)h(0)\right]$$

The user specifies the integration step size (Δ) by the input of n_0. Then $\Delta = \Delta t/n_0$. Note that Δ is held constant for each successive evaluation of y(t).



The user specifies the number of equi-spaced arguments (N) for which the integral is to be evaluated. After the output of N values of y(t), one may generate additional values (in increments of Δt).

The functions x(t) and h(t) are entered in program memory as a series of keystrokes.

Reference: *Linear Systems in Communication and Control,* Frederick and Carlson, John Wiley and Sons 1971.

Solid State Software				TI © 1979
CONVOLUTION				EE-15
x(t)	h(t)		→ y(t) dspl	→ Dsp out
No. panels	N	Δt	→ y(t) prt	→ Prt out

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Initialize		[RST]	
2	Select learn mode.		[LRN]	
3	Use A′ as the label for x(t) subroutine.		[2nd] [Lbl] [2nd] [A′]	
4	Enter x(t) as a series of keystrokes. Assume t is in the display. Do not use [=] or [CLR] or any used registers.			
5	End subroutine with [INV] [SBR]		[INV] [SBR]	
6	Use B′ as the label for h(t) subroutine.		[2nd] [Lbl] [2nd] [B′]	
7	Enter h(t) as in steps 4 and 5			
8	Exit learn mode		[LRN]	
9	Select program 15		[2nd] [Pgm] 15	
10	Enter number of panels in each Δt	n_0	[A]	n_0
11	Number of values of y(t) desired	N	[B]	N
12	Enter increment Δt	Δt	[C]	Δt
13	Execute without printer		[2nd] [E′]	t_1 (flashing) $y(t_1)$
	Repeat until N values have been displayed		[R/S]	t_2 (flashing) $y(t_2)$
14	Execute with printer		[E]	$t_1, y(t_1)$ through $t_N, y(t_N)$ are printed
15	If additional values of y(t) are desired without printer[1]		[2nd] [D′]	t_{N+1} (flashing) $y(t_{N+1})$
16	If additional values of y(t) are desired with printer[1]		[D]	$t_{N+1}, y(t_{N+1})$ are printed

NOTES: 1. Step 15 or 16 may be repeated as many times as desired.

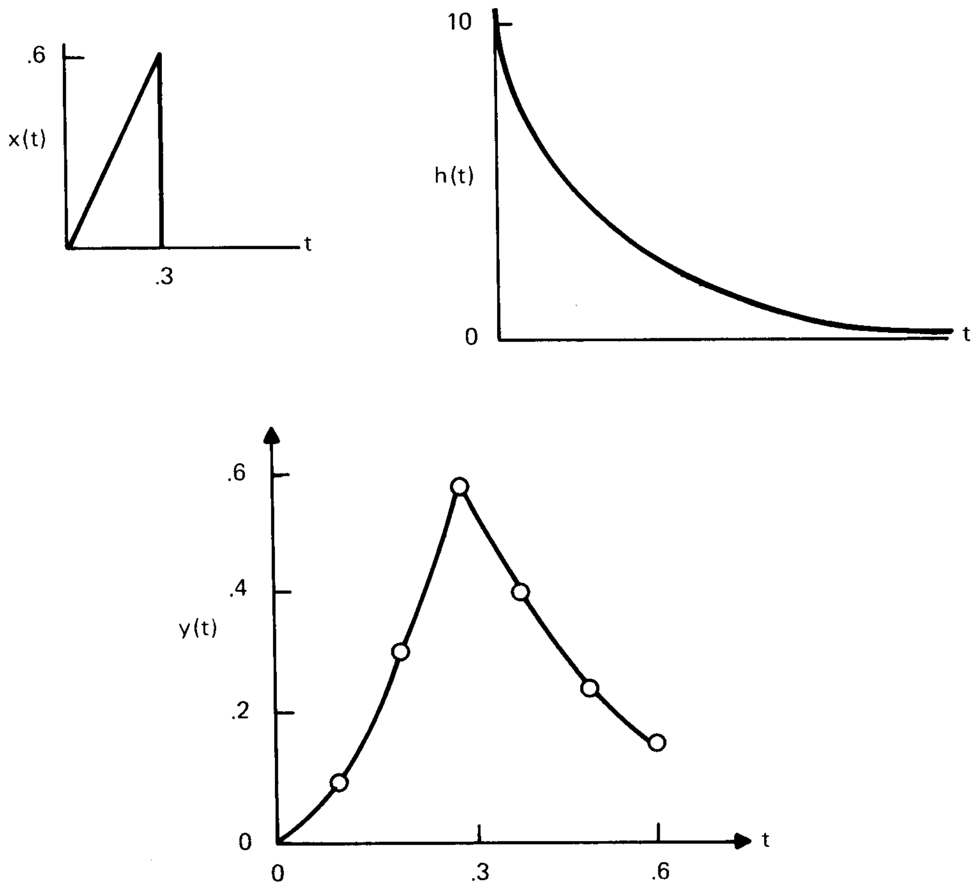
Example: Given $x(t) = 2t$ for $t \leqslant .3$

$= 0$ for $t > .3$

$h(t) = 10e^{-5t}$

Find $y(t)$ at $t = .1, .2, \cdots, .6$ using $n_0 = 4$. Then find $y(t)$ at $t = .7$

The problem may be graphed as follows:



ENTER	PRESS	DISPLAY	COMMENTS
	[RST]		Initialize
	[LRN]	000 00	Select learn mode
	[2nd] [Lbl] [2nd] [A′]		Enter x(t) subroutine
	[( ] [CE] [×] [2] [ )] [x≷t] .6 [2nd] [x≥t] [2nd] [Nop] [2nd] [CP] [2nd] [Lbl] [2nd] [Nop] [x≷t] [INV] [SBR]	017 00	

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Lbl] [2nd] [B']		Enter h(t) subroutine	
	[( ] [( ] [CE] [X] 5			
	[+/–] [ )] [INV] [lnx]			
	[X] 10 [ )] [INV] [SBR]	033 00		
	[LRN]		Leave learn mode	
	[2nd] [Pgm] 15		Select program 15	
4	[A]	4.	Enter n_0	
6	[B]	6.	Enter N	
.1	[C]	.1	Enter Δt	
	[E]		Execute	0.1 T
				.0861547694 Y(T)
				0.2 T
				.2960028933 Y(T)
				0.3 T
				0.580874829 Y(T)
				0.4 T
				.3978081927 Y(T)
				0.5 T
				.2412828656 Y(T)
				0.6 T
		.1463454556		.1463454556 Y(T)
	[D]		Execute for t=.7	0.7 T
		.0887630057		.0887630057 Y(T)

Register Contents

R_{00}	R_{05}	R_{10} n_0	R_{15} Used	R_{20}	R_{25}
R_{01}	R_{06}	R_{11} Used	R_{16} Used	R_{21}	R_{26}
R_{02}	R_{07}	R_{12} N	R_{17} y(t)	R_{22}	R_{27}
R_{03}	R_{08} Used	R_{13} Used	R_{18}	R_{23}	R_{28}
R_{04}	R_{09} Used	R_{14} Δt	R_{19}	R_{24}	R_{29}

ROOT LOCUS CALCULATIONS

Given the open-loop poles and zeros of a linear feedback system, this program calculates the following root-locus parameters: asymptote intersection point, asymptote angles, departure angles from complex poles, and arrival angles at complex zeros.

Given

$$F(s) = \frac{\prod_{i=1}^{m}(s - z_i)}{\prod_{k=1}^{n}(s - p_k)} \qquad m \leqslant n - 1$$

where z_i are the zeros and p_k the poles of $F(s)$, the root-locus has $n - m$ asymptotes intersecting at

$$\sigma_0 = \frac{1}{n-m}\left(\sum_{k=1}^{n} p_k - \sum_{i=1}^{m} z_i\right)$$

at the angles $\psi_\nu = \frac{1}{n-m}(180° + (\nu - 1)\,360°) \qquad \nu = 1, 2, \cdots, n - m$

The departure angle at any complex pole p_J is

$$\phi_J = \sum_{i=1}^{m} \angle(p_J - z_i) - \sum_{\substack{k=1 \\ k \neq J}}^{n} \angle(p_J - p_k) + 180° \qquad J = 1, 2, \cdots, n$$

The arrival angle at any complex zero z_K is

$$\theta_K = -\sum_{\substack{i=1 \\ i \neq K}}^{m} \angle(z_K - z_i) + \sum_{k=1}^{n} \angle(z_K - p_k) - 180° \qquad K = 1, 2, \cdots, m$$

The following program computes σ_0, ψ_ν, ϕ_J, and θ_K for systems no greater than 13th order ($n \leqslant 13$, $m \leqslant 12$). We assumed a real system in the program so that complex poles or zeros always have complex-conjugate mates. Thus

$$\sum_{k=1}^{n} p_k = \sum_{k=1}^{n} \mathrm{Re}(p_k) \quad \text{and} \quad \sum_{i=1}^{m} z_i = \sum_{i=1}^{m} \mathrm{Re}(z_i)$$

Reference: *Linear Systems in Communication and Control,* Frederick and Carlson, John Wiley and Sons, 1971.

Solid State Software				TI © 1979
ROOT LOCUS CALCULATIONS				EE-16
n(poles)	m(zeros)			
Real; Imag		→Asym. Par.	→ Dep∠	→ Arr∠

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select program		[2nd] [Pgm] 16	
2	Initialize		[SBR] [CLR]	0.
3	Repartition (TI-58 only)	6	[2nd] [Op] 17	0.59
4	Enter number of poles	n[2]	[2nd] [A']	n
5	Enter real part of 1st pole	$Re(p_1)$	[A]	1.
6	Enter imag. part of 1st pole	$Im(p_1)$	[R/S]	1.
7	Repeat steps 5 and 6 until n poles have been entered. Steps 5 and 6 must be performed in sequence.	$Re(p_i)$ $Im(p_i)$	[A] [R/S]	i i
8	Enter number of zeros	m[1,2]	[2nd] [B']	m
9	Enter real part of 1st zero	$Re(z_1)$	[A]	1.
10	Enter imag. part of 1st zero	$Im(z_1)$	[R/S]	1.
11	Repeat steps 9 and 10 until m zeros have been entered. Steps 9 and 10 must be performed in sequence	$Re(z_i)$ $Im(z_i$	[A] [R/S]	i i
12	Calculate asymptote intersection			σ_0 †
13	Calculate asymptote angles			ψ_1 †
14	Repeat step 13 until (n−m) angles have been calculated.			
15	Calculate departure angles from poles. Repeat for J = 1,n			J† ϕ_J †
16	Calculate arrival angles at zeros. Repeat for K = 1,m			K† θ_K †

NOTES: 1. If m>(n−1), error condition will result at execution time.

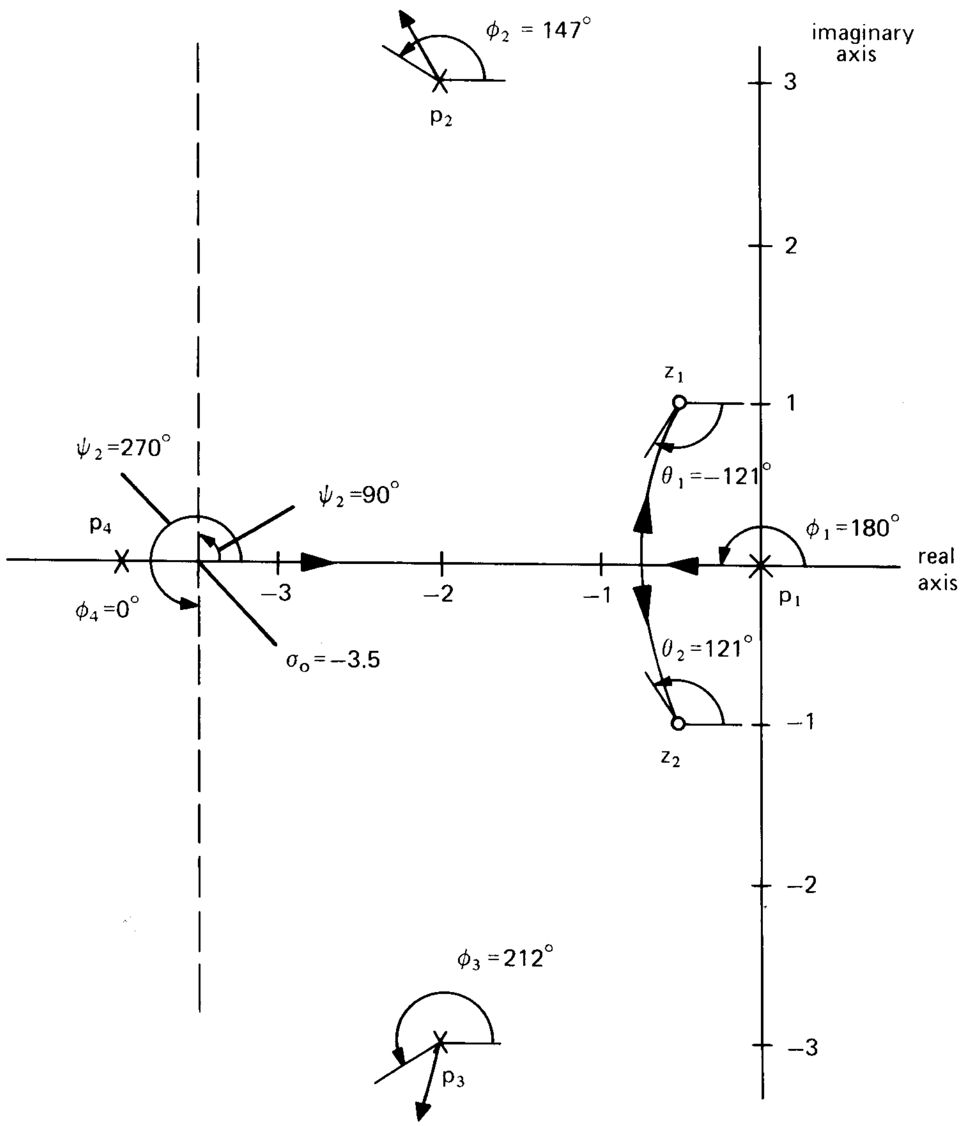
2. $n \leqslant 13$, $m \leqslant 12$

†These values are printed if printer is connected.

Example: Find root locus parameters for the following transfer function.

$$F(s) = \frac{[s-(-.5+j)]\ [s-(-.5-j)]}{(s-0)\ [s-(-2+3j)]\ [s-(-2-3j)]\ [s-(-4)]}$$

GRAPH OF ROOT LOCUS



ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 16		Select program	
	[SBR] [CLR]	0.	Initialize	
6	[2nd] [Op] 17	0.59	Repartition TI-58	
4	[2nd] [A′]	4.	Enter no. of poles	
0	[A]	1.	Enter real of p_1	
0	[R/S]	1.	Enter imag. of p_1	
2	[+/−] [A]	2.	Enter real of p_2	
3	[R/S]	2.	Enter imag. of p_2	

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
2	[+/−] [A]	3.	Enter real of p_3	
3	[+/−] [R/S]	3.	Enter imag. of p_3	
4	[+/−] [A]	4.	Enter real of p_4	
0	[R/S]	4.	Enter imag. of p_4	
2	[2nd] [B′]	2.	Enter no. of zeros	
.5	[+/−] [A]	1.	Enter real of z_1	
1	[R/S]	1.	Enter imag. of z_1	
.5	[+/−] [A]	2.	Enter real of z_2	
1	[+/−] [R/S]	2.	Enter imag. of z_2	
	[C]	−3.5	σ_0	−3.5
	[R/S]	90.	ψ_1	90.
	[R/S]	270.	ψ_2	270.
1	[D]	180.	ϕ_1	1. 180.
2	[D]	147.4259429	ϕ_2	2. 147.4259429
3	[D]	212.5740571	ϕ_3	3. 212.5740571
4	[D]	.0000000004	ϕ_4	4. .0000000004
1	[E]	−121.1757005	θ_1	1. −121.1757005
2	[E]	121.1757005	θ_2	2. 121.1757005

Register Contents

R_{00}	Used	R_{12}	Re (p_2)
R_{01}	Used	R_{13}	Im (p_2)
R_{02}	Used	⋮	
R_{03}	Used	R_{34}	Re (p_{13})
R_{04}	Used	R_{35}	Im (p_{13})
R_{05}	Used	R_{36}	Re (z_1)
R_{06}	Used	R_{37}	Im (z_1)
R_{07}	(n−m)	R_{38}	Re (z_2)
R_{08}	n(poles)	R_{39}	Im (z_2)
R_{09}	m(zeros)	⋮	
R_{10}	Re(p_1)	R_{58}	Re (z_{12})
R_{11}	Im(p_1)	R_{59}	Im (z_{12})

DISCRETE FOURIER TRANSFORM

This program transforms a real time series to the frequency domain (DFT) and performs the inverse transform (IDFT) from the frequency domain back to the time domain. The frequency representation of the transform is given in terms of magnitude in dB and phase in degrees. The program allows a DFT sampling rate of up to 88 points with the TI-59 and 48 points with the TI-58. For IDFT's, the user can generate a time-domain image of a signal by taking up to a 32-point sample (16-point, TI-58) of its symmetric frequency domain representation. It is assumed that the time series reflects a periodic function over some finite interval so that the frequency domain behavior can also be expressed as a discrete series representing a periodic function. The following equations are used.

Discrete Fourier Transform (DFT)

$$F_{re}(X) = \sum_{n=0}^{N-1} f(n) \cos\left(\frac{2\pi n X}{N}\right)$$

$$F_{im}(X) = -\sum_{n=0}^{N-1} f(n) \sin\left(\frac{2\pi n X}{N}\right) \qquad X = 0, 1, 2, \cdots, N-1$$

$$|F(X)| = \sqrt{[F_{re}(X)]^2 + [F_{im}(X)]^2}$$

$$\text{magnitude in dB} = 20 \log |F(X)|$$

$$\text{phase in degrees, } \angle F(X) = \tan^{-1}\left[\frac{F_{im}(X)}{F_{re}(X)}\right], \quad -90^\circ \leqslant \angle F(X) < 270^\circ$$

(see page V-31, *Personal Programming)*

where

$F(X) = F_{re}(X) + j F_{im}(X)$ is the transform of the sample at X.

$F_{re}(X)$ = real part of the transform

$F_{im}(X)$ = imaginary part of the transform

$f(n)$ = value of the waveform sampled at n

N = total number of samples

Inverse Discrete Fourier Transform (IDFT)

$$f(X) = \frac{1}{N} \sum_{n=0}^{N-1} F_{re}(n) \cos\left(\frac{2\pi n X}{N}\right) - F_{im}(n) \sin\left(\frac{2\pi n X}{N}\right)$$

where

$f(X)$ = the value of the waveform sampled at X

$F_{re}(n)$ = real part of the transform at sample n

$F_{im}(n)$ = imaginary part of the transform at sample n

N = total number of samples

The program assumes that the time function f and the frequency function F are approximated by N equally spaced samples at intervals of ΔT (DFT) or $1/\Delta T$ (IDFT). These N samples define the Discrete Fourier Transform pair $f \Leftrightarrow F$ because the N time and frequency values give a discrete representation of the time and frequency domain waveforms, respectively. To minimize the effects of aliasing, choose N such that the DFT sample interval is $\Delta T \leq 1/2f_c$ where f_c is the highest frequency component of the transform and the IDFT sample interval is $1/\Delta T \geq 2f_c$ which is the Nyquist sampling rate.

Data Input

Data is input through User Instruction steps 4 and 5 or steps 8 and 9. Alternately the data can be input directly into the following data registers.

For DFT:

$f(0) \rightarrow R_{12}$
$f(1) \rightarrow R_{13}$
⋮
$f(n) \rightarrow R_{n+12}$
⋮
$f(47) \rightarrow R_{59}$, 48 samples TI-58 limit
⋮
$f(87) \rightarrow R_{99}$, 88 samples TI-59 limit

For IDFT:

$F_{re}(0) \rightarrow R_{12}$; $F_{im}(0) \rightarrow R_{44}$
$F_{re}(1) \rightarrow R_{13}$; $F_{im}(1) \rightarrow R_{45}$
⋮
$F_{re}(n) \rightarrow R_{n+12}$; $F_{im}(n) \rightarrow R_{n+44}$
⋮
$F_{re}(15) \rightarrow R_{27}$; $F_{im}(15) \rightarrow R_{59}$, 16 samples TI-58 limit
⋮
$F_{re}(31) \rightarrow R_{43}$; $F_{im}(31) \rightarrow R_{75}$, 32 samples TI-59 limit

References: 1. *The Fast Fourier Transform*, E. Oran Brigham, Prentice Hall, 1974.
2. *An Introduction to Discrete Systems,* Kenneth Steiglitz, John Wiley and Sons, 1974.

Solid State Software				TI © 1979
DISCRETE FOURIER TRANSFORM				EE-17
				→ IDFT
n	f(n)	|F(n)|	∠F(n)	→ DFT

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select program		[2nd] [Pgm] 17	
2	Initialize		[SBR] [CLR]	0.
3	Repartition (6 TI-58, 10 TI-59)	6 or 10	[2nd] [Op] 17	0.59 or 159.99
	For DFT			
	Enter sample number (0≤n≤N − 1)[1]	n	[A]	n†
4	Enter sample number (0≤n≤N−1)[1]	n	[A]	n†
5	Enter sample value[1]	f(n)	[B]	f(n)†
6	Repeat steps 4 and 5 until all data is entered[1]			
7	Enter total number of samples and compute DFT[4]	N[2]	[E]	
	sample number			n†
	real part of transform			$F_{re}(n)$†
	imaginary part of transform			$F_{im}(n)$†
	magnitude of transform[3]			|F(n)|†
	magnitude of transform in dB			20 log |F(n)|†
	phase of transform in degrees			∠F(n)†
	This six-part sequence is repeated for all values of n.			
	For IDFT			
8	Enter sample number (0≤n≤N−1)[1]	n	[A]	n†
9a	Enter magnitude of transform at n[1]	|F(n)|	[C]	|F(n)|†
9b	Enter phase in deg of transform at n[1]	∠F(n)	[D]	∠F(n)†
10	Repeat steps 8 and 9 until all data is entered[1]			
11	Enter total number of samples and compute IDFT	N[2]	[2nd] [E']	
	sample number			n†
	time value at n			f(n)†
	This two-part sequence is repeated for all values of n.			

NOTES:

1. Steps 4, 5, and 6 or 8, 9, and 10 can be omitted if data is entered directly into registers. Step 2 clears memories, so zero values don't need to be entered.
2. For DFT, N ≤ 88 for TI-59; N ≤ 48 for TI-58.
   For IDFT, N ≤ 32 for TI-59; N ≤ 16 for TI-58.
3. When magnitude approaches zero, the value of the phase angle can be ignored as the phase becomes indeterminate. If magnitude = 0, the indeterminate state is indicated by flashing nines for magnitude in dB and 45. for phase in degrees.
4. Because DFT's for real time series have magnitudes and phases which are symmetric and antisymmetric respectively, about the sample point corresponding to Int(N/2), program execution may be terminated using [RST] after data from point Int(N/2) has been printed.

†These values are printed if a printer is connected. The output values are displayed only briefly. Therefore, for most applications a printer will be desirable.

Example 1: (Requires TI-59) Recover the original sinusoid S which has a

$$\text{DFT} = \begin{cases} -j16 \text{ when } n = 2 \\ +j16 \text{ when } n = 30 \\ 0 \text{ elsewhere} \end{cases}$$

where n = sample number, N = 32, the total number of samples. Enter the data directly into the data registers.

The original signal $S = \sin(n\pi/8)$ $n = 0, \cdots, N-1$ which is a sine wave at a frequency of 1/8, the Nyquist frequency, is verified by the output listing.

For further details, see reference 2, chapter 6.

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 17		Select program
	[SBR] [CLR]	0.	Initialize
10	[2nd] [Op] 17	159.99	Repartition (TI-59 required)
16	[+/−] [STO] 46	−16.	Enter data directly into registers
16	[STO] 74	16.	
32	[2nd] [E']		See tape printout, output values are displayed only briefly.

OUTPUT TAPE FOR EXAMPLE 1

n	f(n)
0.	0.
1.	.3826834324
2.	.7071067812
3.	.9238795325
4.	1.
5.	.9238795325
6.	.7071067812
7.	.3826834323
8.	-1.5-12
9.	-.3826834324
10.	-.7071067812
11.	-.9238795325
12.	-1.
13.	-.9238795325
14.	-.7071067812
15.	-.3826834323
16.	3.05-11
17.	.3826834324
18.	.7071067812
19.	.9238795325
20.	1.
21.	.9238795325
22.	.7071067812
23.	.3826834323
24.	-2.8-11
25.	-.3826834324
26.	-.7071067812
27.	-.9238795325
28.	-1.
29.	-.9238795325
30.	-.7071067812
31.	-.3826834324

Example 2a: A problem which is inherent in the Discrete Fourier Transform because of the required time domain truncation is leakage. Truncation of a periodic function at other than a multiple of the period results in a sharp discontinuity in the time domain, or equivalently results in side-lobes in the frequency domain. These side-lobes are responsible for the additional frequency components which are called leakage. The following example illustrates the effect of leakage. A cosine waveform is sampled at N = 32 points with $\Delta T = 1$ and $f_o = 1/9.143$. The waveform is defined by $f(t) = \cos(2\pi f_o t)$. The definition of f_o results in the time truncation interval not chosen equal to a multiple of the period, which causes leakage in the DFT. (Note DFT output showing non-zero frequency components at all discrete frequencies of the discrete transform.) For further information see Reference 1, Chapter 9.

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 17		Select program
	[SBR] [CLR]	0.	Initialize
6	[2nd] [Op] 17	0.59	Repartition (TI-58)
0	[A]	0.†	Enter sample number
1	[B]	1.†	Enter sample value
1	[A]	1.†	Repeat for all samples. See input tape for
.773	[B]	0.773†	values.
⋮	⋮	⋮	
31	[A]	31.†	
.7727	[+/−] [B]	−0.7727†	
32	[E]	0.*	n
		1.0006*	$F_{re}(n)$
		0.*	$F_{Im}(n)$
		1.0006*	$\|F(n)\|$
		.0052099709*	$20 \log \|F(n)\|$
		0.*	$\angle F(n)$
		⋮	See output tape
		31.*	
		1.000639549*	
		−.9390590304*	
		1.372265051*	
		2.748760053*	
		−43.18161483*	

†These displayed values are printed.

*These values are displayed briefly and printed.

INPUT TAPE FOR EXAMPLE 2a

```
 0.  n
 1.  f(n)

      1.
  0.773

      2.
  0.1951

      3.
 -0.4713

      4.
 -0.9238

      5.
 -0.9569

      6.
 -0.5556

      7.
  0.0979

      8.
  0.707

      9.
  0.9951

     10.
  0.8315

     11.
  0.2903

     12.
 -0.3825

     13.
 -0.8818

     14.
 -0.9808

     15.
 -0.6345

     16.
 -0.0001

     17.
  0.6342

     18.
  0.9807

     19.
  0.882

     20.
  0.3828

     21.
 -0.29

     22.
 -0.8313

     23.
 -0.9952

     24.
 -0.7072

     25.
 -0.0982

     26.
  0.5553

     27.
  0.9568

     28.
  0.9239

     29.
  0.4716

     30.
 -0.1947

     31.
 -0.7727
```

OUTPUT TAPE FOR EXAMPLE 2a

0.	n	5.	10.
1.0006	$F_{re}(n)$	.9995790734	.9999412478
0.	$F_{im}(n)$	-3.82334953	-.7990914541
1.0006	$\|F(n)\|$	3.951855255	1.280011582
.0052099709	$20 \log \|F(n)\|$	11.93602058	2.144277989
0.	$\angle F(n)$	-75.34849318	-38.62969454
1.		6.	11.
1.000639549		.9996930668	.9995972063
.9390590304		-2.366527645	-.6261383867
1.372265051		2.569015244	1.179510006
2.748760053		8.195333626	1.434032588
43.18161484		-67.09932919	-32.06264597
2.		7.	12.
1.0008672		0.999613047	1.000077817
2.536411113		-1.696769076	-.4777605807
2.726741		1.969327688	1.108336958
8.712877768		5.886359738	.8934363079
68.46582011		-59.49649849	-25.534898
3.		8.	13.
1.001877504		0.9999	.9999119019
9.503991626		-1.2937	-.3462773071
9.556652937		1.635071772	1.058173797
19.60611629		4.270736419	0.491140064
83.98230479		-52.29958375	-19.10136951
4.		9.	14.
.9935221826		.9997629142	.9998984859
-10.72776058		-1.012977911	-.2257526962
10.77413076		1.423253362	1.025066467
20.64764484		3.065644368	.2150405341
-84.68232514		-45.37617991	-12.72267462

OUTPUT TAPE FOR EXAMPLE 2a (continued)

```
        15. n
 .9998188047 Fre(n)
-.1110298813 Fim(n)

 1.005964849 |F(n)|
 .0516561098 20 log |F(n)|
-6.336733188 ∠F(n)

        16.
         1.
 2.57135-11

         1.
         0.
 .0000000015

        17.
 .9998188047
 .1110298815

 1.005964849
 .0516561101
 6.336733197

        18.
 .9998984859
 .2257526963

 1.025066467
 .2150405343
 12.72267462

        19.
 .9999119019
 .3462773072

 1.058173797
 .4911400638
 19.10136951
```

```
        20.
 1.000077817
 .4777605808

 1.108336958
 .8934363074
 25.53489801

        21.
 .9995972063
 .6261383867

 1.179510006
 1.434032588
 32.06264598

        22.
 .9999412478
 .7990914542

 1.280011582
  2.14427799
 38.62969454

        23.
 .9997629142
 1.012977911

 1.423253362
 3.065644368
 45.37617991

        24.
 .9998999999
      1.2937

 1.635071772
 4.270736419
 52.29958375
```

```
        25.
 .9996130469
 1.696769076

 1.969327688
 5.886359737
 59.49649849

        26.
 .9996930666
 2.366527645

 2.569015244
 8.195333626
 67.09932919

        27.
 .9995790734
  3.82334953

 3.951855255
 11.93602058
 75.34849318

        28.
 0.998522182
 10.72776058

 10.77413076
 20.64764484
 84.68232514

        29.
 1.001877504
-9.503991627

 9.556652938
 19.60611629
-83.98230478
```

OUTPUT TAPE FOR EXAMPLE 2a (continued)

30.	n	31.	
1.0008672	$F_{re}(n)$	1.000639549	
-2.536411112	$F_{im}(n)$	-.9390590304	
2.726741	$\|F(n)\|$	1.372265051	
8.712877768	20 log $\|F(n)\|$	2.748760053	
-68.46582011	$\angle F(n)$	-43.18161483	

Example 2b: A technique for reducing leakage employs a time domain truncation function

$$X(t) = \frac{1}{2} - \frac{1}{2}\cos\left(\frac{2\pi t}{N}\right)$$

where N = total number of samples.

X(t) is known as the Hanning function. The cosine waveform of example 2a will be multiplied by the Hanning function and the resulting waveform

$$f(t) \cdot X(t) = \cos(2\pi f_o t)\left[\frac{1}{2} - \frac{1}{2}\cos(2\pi t/N)\right]$$

will be sampled at N = 32 points with $\Delta T = 1$ and $f_o = 1/9.143$. The effect of the Hanning function is to reduce the discontinuity which results from the time domain truncation and, consequently, minimize the leakage. Note the DFT outputs, particularly for the discrete frequencies from sample 6 to sample 26, which are now closer to zero.

ENTER	PRESS	DISPLAY	COMMENTS
	[2nd] [Pgm] 17		Select program
	[SBR] [CLR]	0.	Initialize
6	[2nd] [Op] 17	0.59	Repartition (TI-58)
0	[A]	0.†	Enter sample number
0	[B]	0.†	Enter sample value
⋮	⋮	⋮	Repeat for all samples. See input tape for values.
31	[A]	31.†	
.0074	[+/–] [B]	–0.0074†	
32	[E]	0.*	n
		0.0002*	$F_{re}(n)$
		0.*	$F_{im}(n)$
		0.0002*	$\|F(n)\|$
		–73.97940009*	20 log $\|F(n)\|$
		0.*	$\angle F(n)$
		⋮	See output tape
		31.*	
		.0000007759*	
		.1646458942*	

ENTER	PRESS	DISPLAY	COMMENTS
		.1646458942*	
		–15.6689819*	
		89.99973*	

†These displayed values are printed.

*These values are displayed briefly and printed.

INPUT TAPE FOR EXAMPLE 2b

n	f(n)
0.	0.
1.	0.0074
2.	0.0074
3.	-0.0397
4.	-0.1352
5.	-0.2126
6.	-0.1714
7.	0.0394
8.	0.3535
9.	0.5946
10.	0.5748
11.	0.2258
12.	-0.3265
13.	-0.8075
14.	-0.9434
15.	-0.6284
16.	-0.0001
17.	0.6281
18.	0.9434
19.	0.8076
20.	0.3268
21.	-0.2256
22.	-0.5747
23.	-0.5946
24.	-0.3536
25.	-0.0395
26.	0.1714
27.	0.2126
28.	0.1353
29.	0.0397
30.	-0.0074
31.	-0.0074

OUTPUT TAPE FOR EXAMPLE 2b

n	$F_{re}(n)$	$F_{im}(n)$	$\|F(n)\|$	$20 \log \|F(n)\|$	$\angle F(n)$
0.	0.0002	0.	0.0002	-73.97940009	0.
1.	.0000007758	-.1646458943	.1646458943	-15.66898189	-89.99973001
2.	-.0002937549	-1.342759838	1.34275987	2.559967068	269.9874654
3.	.0008544733	6.799605919	6.799605973	16.64967493	89.99279992
4.	-.0010949747	-6.783699198	6.783699287	16.62933176	269.9907517
5.	0.000291618	1.362082295	1.362082327	2.684067158	89.98773314
6.	.0000865619	0.19686324	0.196863259	-14.11670659	89.9748067
7.	.0000427157	.0667184439	.0667184576	-23.51508005	89.96331706
8.	0.0001	0.0307	0307001629	-30.25718641	89.81336947
9.	-.0000490217	.0167893998	.0167894714	-35.49925956	90.16729186
10.	.0000548594	.0101008511	0.010101	-39.91271254	89.68882002
11.	.0000064488	.0063063648	.0063063681	-44.00441363	89.94141006
12.	-.0001050253	.0041008017	.0041021464	-47.73977698	91.46707649
13.	-.0001868547	.0025290107	.0025359042	-51.91734328	94.22559234
14.	.0001523336	0.001677773	.0016846744	-55.46968039	84.8120484

OUTPUT TAPE FOR EXAMPLE 2b (continued)

n	$F_{re}(n)$	$F_{im}(n)$	$\lvert F(n)\rvert$	$20 \log \lvert F(n)\rvert$	$\angle F(n)$
15.	-.0001601552	.0009240838	.0009378596	-60.55724333	99.8324027
16.	0.0004	1.3272-12	0.0004	-67.95880017	.0000001901
17.	-.0001601552	-.0009240838	.0009378596	-60.55724384	260.1675979
18.	.0001523336	-0.001677773	.0016846744	-55.46968041	-84.8120479
19.	-.0001868547	-.0025290107	.0025359041	-51.91734335	265.7744077
20.	-.0001050253	-.0041008017	.0041021464	-47.73977699	268.5329236
21.	.0000064488	-.0063063648	.0063063681	-44.00441362	-89.94141006
22.	.0000548594	-.0101008511	0.010101	-39.91271254	-89.68882015
23.	-.0000490217	-.0167893999	.0167894714	-35.49925952	269.8327081
24.	0.0001	-.0306999999	.0307001628	-30.25718643	-89.81336954
25.	.0000427157	-.0667184439	.0667184576	-23.51508005	-89.96331706
26.	.0000865619	-0.19686324	0.196863259	-14.11670659	-89.97480673
27.	.0002916181	-1.362082295	1.362082327	2.684067158	-89.98773313
28.	-.0010949751	6.783699198	6.783699287	16.62933176	90.00924827
29.	.0008544736	-6.799605919	6.799605973	16.64967493	-89.99279992

OUTPUT TAPE FOR EXAMPLE 2b (continued)

30.	n	31.
-0.000293755	$F_{re}(n)$	.0000007759
1.342759838	$F_{im}(n)$	.1646458942
1.34275987	$\|F(n)\|$	.1646458942
2.559967069	$20 \log \|F(n)\|$	-15.6689819
90.01253457	$\angle F(n)$	89.99973

Register Contents

R_{00} Used
R_{01} Used
R_{02} Used
R_{03} Used
R_{04} N − 1
R_{05} $F_{re}(X)$ [DFT], f(X) [IDFT]
R_{06} $F_{im}(X)$ [DFT], f(X) [IDFT]
R_{07} X
R_{08} $2\pi nX/N$
R_{09} Used
R_{10} Used
R_{11} N

R_{12} ↓ R_{59} Data, TI-58

R_{12} ↓ R_{99} Data, TI-59

SMITH CHART CALCULATIONS

Graphical constructions on the Smith Chart require some special techniques, and it can be difficult to obtain accurate results. The program given here performs various transmission-line calculations equivalent to the graphical constructions on the Smith Chart. The program makes provisions for lines with attenuation and complex characteristic impedance. Thus you can obtain accurate results quickly and easily with your programmable calculator.

Consider a transmission line with characteristic impedance Z_0 (possibly complex). The normalized impedance (z) at any point will be the impedance Z at that point divided by Z_0 ($z = Z/Z_0$). Let the normalized distance in wavelengths from the load end of the line be $x = d/\lambda$. Then the per-wavelength phase constant is given by

$$b = \beta\lambda = 2\pi \text{ (radians)}$$

and the attenuation constant is

$$a = 8.686\alpha\lambda \text{ (dB per wavelength)}$$

where α is the attenuation factor (nepers per unit length).

This program has subroutines which will perform different calculations depending on the various inputs. The following terms are used in many equations and will be defined here for convenience.

Z_0 = characteristic impedance
Z = impedance*
z = normalized impedance*
Y = admittance*
P = reflection factor* (rectangular coordinates)
$|P|\angle P$ = reflection factor* (polar coordinates)
σ = voltage standing wave ratio (VSWR)

*Note that a subscript of "L" indicates values at termination, and a subscript of "i" indicates values at input.

The program uses rectangular representation a + bj for entry of complex quantities. If polar to rectangular conversion is desired, see page V-31 of *Personal Programming* or use Program EE-05.

The program allows calculation of the input impedance Z_i, the reflection factor P, and the VSWR σ when the characteristic impedance Z_0, the termination resistance Z_L, the attenuation a, and the distance in wavelengths X from the termination are entered. Provisions are made to allow conversion from admittance to impedance.

This program is designed for use with the printer.

Solid State Software			TI © 1979	
SMITH CHART CALCULATIONS				EE-18
Z_0	Z	Y		
a	σ	X	P	→ Execute

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select program		[2nd] [Pgm] 18	
2	Initialize		[SBR] [CLR]	0.
3	Enter complex characteristic impedance. Omit for $Z_0 = 50 + 0_j$ ohms[1]	$Re(Z_0)$ $Im(Z_0)$		$Re(Z_0)^\dagger$ $Im(Z_0)^\dagger$
4	Enter termination impedance Z_L in ohms. If admittance given, see step 10.	$Re(Z_L)$ $Im(Z_L)$	[2nd] [B'] [R/S]	$Re(Z_L)$ $Re(P_L)^*$ $Im(P_L)^\dagger$
5	Enter attenuation in dB. Omit for a = 0.[1]	a	[A]	$a^\dagger$
6	Enter distance X in wavelengths from termination	X	[C]	$x^\dagger$
7	Compute input impedance Z_i		[E]	$Re(Z_i)^*$ $Im(Z_i)^\dagger$
8	Display magnitude of P_i[2] Display angle of P_i		[D] [R/S]	$\|P_i\|^\dagger$ $\angle P_i^\dagger$
9	Enter magnitude of P and calculate VSWR[3]	|P|	[B]	$\sigma^\dagger$
10	Perform Y⇌Z conversions	Re(Y) Im(Y)	[2nd] [C'] [R/S]	Re(Y) $Re(Z)^*$ $Im(Z)^\dagger$
		Re(Z) Im(Z)	[2nd] [C'] [R/S]	Re(Z) $Re(Y)^*$ $Im(Y)^\dagger$

NOTES:

1. Values for Z_O and a are set at $50 + 0_j$ and 0, respectively, if they are not entered.
2. This calcualtion is performed during execution of input impedance at step 7. Thus step 8 must immediately follow step 7 (Display of P_i is optional).
3. If $|P_i|$ is desired for VSWR, enter $|P_i|$ using [D] in step 8. If $|P_L|$ is desired for VSWR, enter $|P_L|$ by pressing [RCL] 10 after execution of step 7.

†These displayed values are printed if printer is connected.

*These values are printed only.

Example: Given a transmission line with a characteristic impedance (Z_0) of 50 + 0j ohms, an attenuation (a) of 0.4 dB, and a termination impedance of 10 + 40j ohms, find the input impedance (Z_i) and the VSWR (σ) at 1, 2, and 3 wavelengths (X) from the termination.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 18		Select program	
	[SBR] [CLR]	0.	Initialize	
10	[2nd] [B']	10.	Enter Z_L in ohms	
40	[R/S]		Real P_L	−.1538461538
		.7692307692	Imag P_L	.7692307692
.4	[A]	0.4	Enter a in dB	0.4
1	[C]	1.	Enter X_1 in wavelengths	1.
	[E]		Real Z_{i1}	13.61653082
		39.1384787	Imag Z_{i1}	39.1384787
	[D]	.7154401641	Display $\|P_{i1}\|$	.7154401641
	[B]	6.028398767	Calculate σ_1	6.028398767
2	[C]	2.	Enter X_2 in wavelengths	2.
	[E]		Real Z_{i2}	17.07404104
		38.04661406	Imag Z_{i2}	38.04661406
	[D]	.6524891845	Display $\|P_{i2}\|$	.6524891485
	[B]	4.75521656	Calculate σ_2	4.75521656
3	[C]	3.	Enter X_3 in wavelengths	3.
	[E]		Real Z_{i3}	20.34245021
		36.75665428	Imag Z_{i3}	36.75665428
	[D]	.5950772089	Display $\|P_{i3}\|$	.5950772089
	[B]	3.939213212	Calculate σ_3	3.939213212

Register Contents

R_{00}	R_{05} X	R_{10} $\|P_L\|$	R_{15} $\angle P$	R_{20} a	R_{25}
R_{01} Re X	R_{06} Re Z_0	R_{11}	R_{16} Re Z_L	R_{21}	R_{26}
R_{02} Im X	R_{07} Im Z_0	R_{12}	R_{17} Im Z_L	R_{22}	R_{27}
R_{03} Re Y	R_{08} Re P_L	R_{13}	R_{18}	R_{23}	R_{28}
R_{04} Im Y	R_{09} Im P_L	R_{14} $\|P_i\|$	R_{19}	R_{24}	R_{29}

Method Used

Assuming a low-loss line, the following computations are performed.

Reflection Factor at Termination

Given the termination impedance

$$Z_L = \mathrm{Re}(Z_L) + j\,\mathrm{Im}(Z_L)$$

or admittance

$$Y_L = \mathrm{Re}(Y_L) + j\,\mathrm{Im}(Y_L)$$

the complex reflection factor is calculated as follows.

$$Z_L = \frac{1}{Y_L} \qquad \text{(if } Y_L \text{ entered)}$$

$$z_L = \frac{Z_L}{Z_0} \qquad P_L = \frac{z_L - 1}{z_L + 1}$$

Input Impedance

Given Z_L and P_L from a previous calculation, the input impedance

$$Z_i = \mathrm{Re}(Z_i) + j\,\mathrm{Im}(Z_i)$$

at distance X from the termination is computed by the following sequence.

$$|P_i| = |P_L|\,(10^{-0.1a})^X \qquad \angle P_i = \angle P_L - 4\pi X$$

$$z_i = \frac{1 + P_i}{1 - P_i} \qquad Z_i = Z_0 z_i$$

Note that this sequence applies in the reverse direction (from Z_i to Z_L) if X is entered as a negative quantity.

Admittance and VSWR

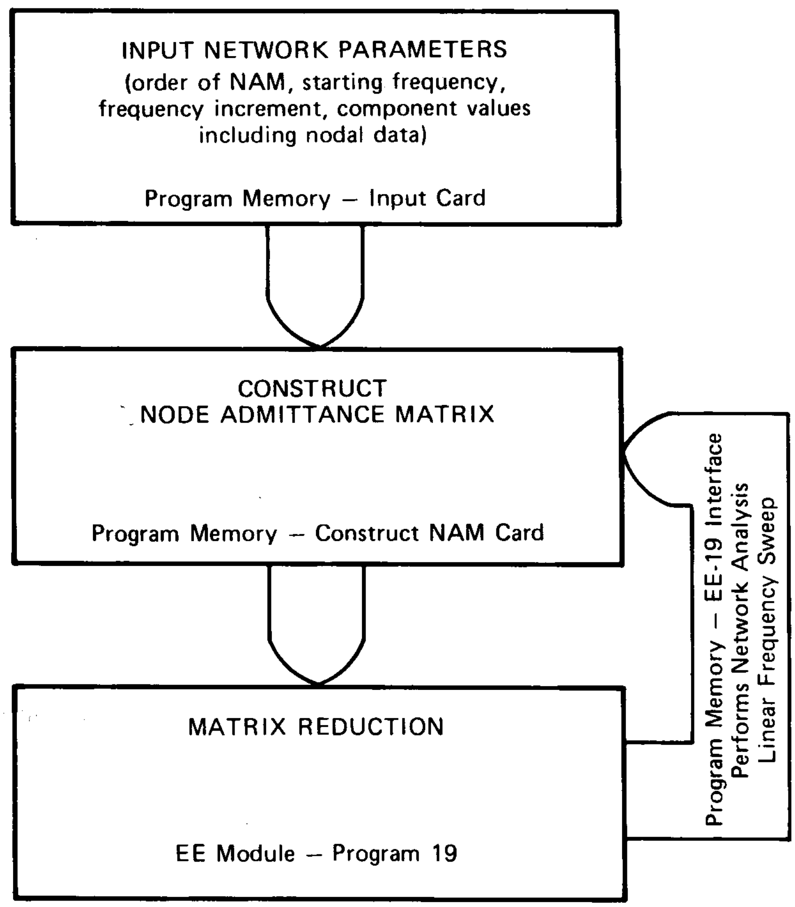
The following calculations are the same for either input or termination values.

$$Y = \frac{1}{Z} \qquad \sigma = \frac{1 + |P|}{1 - |P|}$$

Reference: *Electric Transmission Lines,* Skilling, McGraw-Hill 1951.

NETWORK ANALYSIS

This program is designed for use with the TI-59 and the printer. The program computes frequency response of a general linear network made up of resistors, capacitors, inductors, and voltage controlled dependent current sources. Networks of up to 5 nodes (exclusive of ground) and 9 components, 4 nodes and 14 components, or 3 or 2 nodes and 24 components can be analyzed. The analysis is accomplished by interfacing three blocks of code as illustrated below.



The first block consists of the input code and should be keyed into program memory and recorded on the designated magnetic card per specified instructions. Execution of this code in User Instructions 1 through 6 specifies the order of the NAM (node admittance matrix), starting frequency and increment in hertz to initiate a linear frequency sweep, and remaining network parameters by entering component values and node numbers associated with each component. The number of a component is printed following input of its data to provide easy editing reference. (See User Instructions, Note 3.)

The second block consisting of the Construct NAM code should be keyed into program memory and recorded on the designated card per specified instructions. Execution of this code in User Instructions 7 through 9 constructs the network node admittance matrix and initiates the loop between program memory and the third block of code, EE module program 19, which performs the matrix reduction. (See Method Used section for explanation of NAM construction and matrix reduction.)

Program output is in the form of a linear frequency sweep which depends on starting frequency and increment specifications. Printout consists of the frequency in hertz, magnitude of the output voltage in dB and the phase in degrees.

Two magnetic cards are used in conjunction with this program. Instructions for coding and recording these cards are included in the following sections.

Input Card Preparation

PROCEDURE	PRESS	DISPLAY
1. Repartition	**9 [2nd] [Op] 17**	239.89
2. Clear program memory	**[2nd] [CP]**	239.89
3. Enter learn mode	**[LRN]**	000 00
4. Key in input code (see Listing)		
5. Check input code against listing	**[CLR] [2nd] [List]**	
6. Prepare to record Input Card	**1 [2nd] [Write]**	Blank
7. Insert card		1.

Label this card Input Card and place it in the card case for storage.

See *Personal Programming*, pp. V-44, 45 and VII-2, 3 for information on keying in programs and recording magnetic cards.

INPUT CODE LISTING

000	76	LBL
001	11	A
002	99	PRT
003	32	X⇌T
004	02	2
005	32	X⇌T
006	22	INV
007	67	EQ
008	00	00
009	11	11
010	03	3
011	42	STO
012	87	87
013	42	STO
014	05	05
015	92	RTN
016	99	PRT
017	42	STO
018	78	78
019	42	STO
020	06	06
021	92	RTN
022	99	PRT
023	42	STO
024	79	79
025	42	STO
026	07	07
027	92	RTN
028	76	LBL
029	14	D
030	32	X⇌T
031	93	.
032	03	3
033	61	GTO
034	00	00
035	57	57
036	76	LBL
037	16	A'
038	32	X⇌T
039	93	.
040	01	1
041	61	GTO
042	00	00
043	57	57
044	76	LBL
045	12	B
046	98	ADV
047	99	PRT
048	35	1/X
049	72	ST*
050	02	02
051	00	0
052	72	ST*
053	01	01
054	61	GTO
055	00	00
056	64	64
057	72	ST*
058	01	01
059	32	X⇌T
060	98	ADV
061	99	PRT
062	72	ST*
063	02	02
064	92	RTN
065	22	INV
066	52	EE
067	58	FIX
068	01	01
069	99	PRT
070	65	×
071	01	1
072	00	0
073	95	=
074	74	SM*
075	01	01
076	58	FIX
077	09	09
078	87	IFF
079	00	00
080	01	01
081	03	03
082	02	2
083	94	+/-
084	44	SUM
085	01	01
086	44	SUM
087	02	02
088	43	RCL
089	01	01
090	42	STO
091	03	03
092	43	RCL
093	02	02
094	42	STO
095	04	04
096	01	1
097	44	SUM
098	89	89
099	43	RCL
100	89	89
101	99	PRT
102	92	RTN
103	22	INV
104	86	STF
105	00	00
106	43	RCL
107	04	04
108	42	STO
109	02	02
110	43	RCL
111	03	03
112	42	STO
113	01	01
114	92	RTN
115	76	LBL
116	15	E
117	86	STF
118	00	00
119	98	ADV
120	99	PRT
121	94	+/-
122	65	×
123	02	2
124	85	+
125	07	7
126	09	9
127	75	-
128	42	STO
129	02	02
130	01	1
131	95	=
132	42	STO
133	01	01
134	92	RTN
135	76	LBL
136	10	E'
137	22	INV
138	86	STF
139	00	00
140	22	INV
141	52	EE
142	07	7
143	07	7
144	42	STO
145	02	02
146	07	7
147	06	6
148	42	STO
149	01	01

INPUT CODE LISTING (continued)

```
150  98 ADV
151  69 OP
152  00  00
153  02  2
154  04  4
155  03  3
156  01  1
157  69 OP
158  02  02
159  03  3
160  03  3
161  04  4
162  01  1
163  03  3
164  07  7
165  00  0
166  00  0
167  00  0
168  00  0
169  69 OP
170  03  03
171  69 OP
172  05  05
173  04  4
174  02  2
175  01  1
176  03  3
177  69 OP
178  02  02
179  02  2
180  07  7
181  04  4
182  01  1
183  01  1
184  07  7
185  00  0
186  00  0
187  00  0
188  00  0
189  69 OP
190  03  03
191  69 OP
192  05  05
193  02  2
194  01  1
195  03  3
196  00  0
197  69 OP
198  02  02
199  04  4
200  00  0
201  03  3
202  07  7
203  03  3
204  02  2
205  00  0
206  00  0
207  69 OP
208  03  03
209  69 OP
210  05  05
211  98 ADV
212  92 RTN
213  76 LBL
214  17 B'
215  98 ADV
216  43 RCL
217  05  05
218  42 STO
219  87  87
220  99 PRT
221  43 RCL
222  06  06
223  42 STO
224  78  78
225  99 PRT
226  43 RCL
227  07  07
228  42 STO
229  79  79
230  99 PRT
231  92 RTN
232  76 LBL
233  13  C
234  32 X:T
235  93  .
236  02  2
237  61 GTO
238  00  00
239  57  57
```

Construct NAM Card Preparation

PROCEDURE	PRESS	DISPLAY
1. Repartition	**9 [2nd] [Op] 17**	239.89
2. Clear program memory	**[2nd] [CP]**	239.89
3. Enter learn mode	**[LRN]**	000 00
4. Key in Construct NAM code (see listing)		
5. Check construct NAM code against listing	**[CLR] [2nd] [List]**	
6. Prepare to record Construct NAM Card	**1 [2nd] [Write]**	Blank
7. Insert card		1.

Label this card Construct NAM Card and place it in the card case for storage.

See *Personal Programming,* pp. V-44, 45 and VII-2, 3 for information on keying in programs and recording magnetic cards.

Note: The DEG and NOP instructions which appear at various places from location 160 to location 239 form a necessary part of the code since this part of program memory is recovered from program data in EE-19 to afford maximum register availability for analysis.

CONSTRUCT NAM CODE LISTING

000	76	LBL
001	11	A
002	09	9
003	09	9
004	42	STO
005	05	05
006	01	1
007	00	0
008	42	STO
009	03	03
010	08	8
011	07	7
012	42	STO
013	04	04
014	73	RC*
015	04	04
016	63	EX*
017	05	05
018	72	ST*
019	04	04
020	69	OP
021	34	34
022	69	OP
023	35	35
024	97	DSZ
025	03	03
026	00	00
027	14	14
028	92	RTN
029	76	LBL
030	15	E
031	06	6
032	69	OP
033	17	17
034	47	CMS
035	09	9
036	69	OP
037	17	17
038	07	7
039	07	7
040	42	STO
041	02	02
042	07	7
043	06	6
044	42	STO
045	01	01
046	43	RCL
047	89	89
048	42	STO
049	00	00
050	73	RC*
051	01	01
052	59	INT
053	55	÷
054	01	1
055	00	0
056	95	=
057	42	STO
058	06	06
059	59	INT
060	42	STO
061	05	05
062	22	INV
063	44	SUM
064	06	06
065	01	1
066	00	0
067	49	PRD
068	06	06
069	43	RCL
070	05	05
071	42	STO
072	07	07
073	73	RC*
074	02	02
075	42	STO
076	09	09
077	93	.
078	01	1
079	32	X:T
080	73	RC*
081	01	01
082	22	INV
083	59	INT
084	67	EQ
085	01	01
086	29	29
087	43	RCL
088	05	05
089	71	SBR
090	01	01
091	54	54
092	29	CP
093	43	RCL
094	06	06
095	67	EQ
096	01	01
097	34	34
098	42	STO
099	07	07
100	71	SBR
101	01	01
102	54	54
103	43	RCL
104	05	05
105	42	STO
106	07	07
107	73	RC*
108	02	02
109	94	+/-
110	42	STO
111	09	09
112	43	RCL
113	06	06
114	71	SBR
115	01	01
116	54	54
117	43	RCL
118	06	06
119	42	STO
120	07	07
121	43	RCL
122	05	05
123	71	SBR
124	01	01
125	54	54
126	61	GTO
127	01	01
128	34	34
129	43	RCL
130	06	06
131	71	SBR
132	01	01
133	54	54
134	02	2
135	94	+/-
136	44	SUM
137	01	01
138	44	SUM
139	02	02
140	97	DSZ
141	00	00
142	00	00
143	50	50
144	01	1
145	00	0
146	69	OP
147	17	17
148	11	A
149	36	PGM

CONSTRUCT NAM CODE LISTING (continued)

150	19	19
151	15	E
152	61	GTO
153	15	E
154	42	STO
155	08	08
156	43	RCL
157	87	87
158	75	-
159	01	1
160	00	0
161	95	=
162	65	×
163	43	RCL
164	07	07
165	75	-
166	43	RCL
167	87	87
168	42	STO
169	04	04
170	49	PRD
171	04	04
172	85	+
173	43	RCL
174	08	08
175	85	+
176	60	DEG
177	09	9
178	85	+
179	43	RCL
180	07	07
181	65	×
182	01	1
183	68	NOP
184	00	0
185	95	=
186	42	STO
187	03	03
188	44	SUM
189	04	04
190	68	NOP
191	93	.
192	02	2
193	32	X:T
194	73	RC*
195	01	01
196	22	INV
197	59	INT
198	68	NOP
199	77	GE
200	02	02
201	07	07
202	43	RCL
203	09	09
204	74	SM*
205	03	03
206	92	RTN
207	32	X:T
208	02	2
209	65	×
210	43	RCL
211	78	78
212	65	×
213	43	RCL
214	09	09
215	65	×
216	60	DEG
217	89	π
218	95	=
219	42	STO
220	88	88
221	93	.
222	03	3
223	67	EQ
224	02	02
225	33	33
226	43	RCL
227	88	88
228	94	+/-
229	35	1/X
230	42	STO
231	88	88
232	60	DEG
233	43	RCL
234	88	88
235	74	SM*
236	04	04
237	92	RTN
238	68	NOP
239	68	NOP

Program Assumptions

1. Phase output is always between –90° and 270°. (See *Personal Programming,* p. V-31.)
2. No independent current sources exist and all dependent sources are voltage controlled current sources.
3. Node 0 is the reference or ground node, node 1 is the input node, and node 2 is the output node.
4. A one-volt source is connected between node 1 and node 0.
5. The From Node is not zero when entering the node numbers that a component (R, L, C, or g_m) is connected between.
6. Active components (voltage controlled sources) are specified as follows:

   The controlling voltage, V, is measured between the nodes V^+ and V^- (may be ground or node 0).

   The controlled current g_mV leaves the From node and enters the To node.

7. Networks with inductors are not analyzed at, or very near, zero frequency. (Insure specification of non-zero starting frequency.)
8. DC analysis is realized by redesigning the network, shorting all inductors and specifying a starting frequency of zero.

Outline of Program Usage

1. Review Program Assumptions carefully.
2. Draw circuit to be analyzed.
3. Number the nodes.
4. Enter data as specified. (See User Instructions 1–6.)

5. Perform Network Analysis. (See User Instructions 7–9.)

6. Terminate sweep after desired number of intervals. (See User Instruction 10.)

Once a sweep has been completed, the following instructions can be performed to modify existing network. (This assumes that Input Card, side 2 was recorded per User Instruction Note 5.)

1. Perform User Instruction 2a.

2. Perform User Instruction 2b. (Load Input Card side 1. A 1 will appear in the display.)

3. Press [CLR] and repeat User Instruction 2b. (Load Input Card side 2. A 4 will appear in the display.)

4. To effect a change in starting frequency and/or increment, perform User Instructions 3a, 3b, 3c, entering desired frequency changes.

   If no changes desired, press [2nd] [B']. Old NAM order, starting frequency, and increment will be printed.

5. To change an existing component value, see User Instructions Note 3.

6. To add component values to existing network, perform User Instructions 5 and 6 as needed.

7. For further network updating, re-record bank 4 data. (See User Instructions Note 5.)

8. Perform Network Analysis using new data. (See User Instructions 7–9.)

9. Terminate sweep after desired number of intervals. (See User Instruction 10.)

Solid State Software			TI © 1979	
NETWORK ANALYSIS				EE-19
g_m				
n	R	L	C	→ Execute

USER INSTRUCTIONS

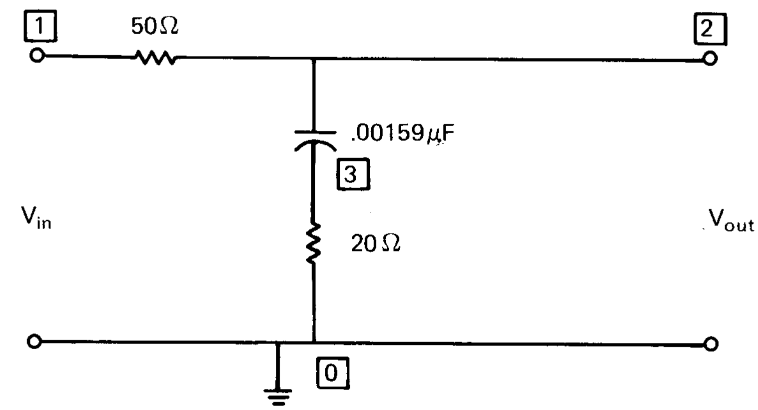
STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1a	Repartition	9	[2nd] [Op] 17	239.89
1b	Initialize		[2nd] [CMs] [RST]	239.89
2a	Prepare to load input card[1]		[2nd] [CP] [CLR]	0
2b	Load input card			1.
3a	Enter order of NAM[2]	n	[A]	$n^{\dagger}$
3b	Enter starting frequency in hertz	f	[R/S]	$f^{\dagger}$
3c	Enter frequency increment in hertz	Δf	[R/S]	$\Delta f^{\dagger}$
4	Initialize for component input		[2nd] [E']	INPUT* VALUE* FM.TO*
	Enter, in any order, values and nodal data for both passive and active components			
	Step 5a, b, c, or d must be followed by Step 6.[3]			
5a	Enter resistor value in ohms OR	R	[B]	$R^{\dagger}$
5b	Enter inductor value in henrys OR	L	[C]	$L^{\dagger}$
5c	Enter capacitor value in farads OR	C	[D]	$C^{\dagger}$
5d	Enter source value in mhos	g_m	[2nd] [A']	$g_m^{\dagger}$
6	Enter nodal data, From node.To node	Fm.To	[R/S]	Fm.To* Component No.†
	Repeat Steps 5 and 6 until all components are entered[5]			
7a	Prepare to load Construct NAM card[1]		[2nd] [CP] [CLR]	0
7b	Load Construct NAM card			1.
	If NAM order = 5, go to step 9			
8a	Enter learn mode		[GTO] [E] [LRN]	031 06
8b	If NAM order = 4	5	[LRN]	1.
8c	If NAM order = 3 or 2	3	[LRN]	1.
9	Perform network analysis[4] (A linear sweep continues until user termination in step 10)		[E]	Frequency* Magnitude (dB)* Phase (deg)* ⋮
10	Terminate linear sweep and repartition	9	[R/S] [RST] [2nd] [Op] 17	239.89

NOTES:
1. Reference: *Personal Programming,* pg. VII-5.
2. n, order of NAM, corresponds to the total number of nodes (excluding ground node zero) and must meet the constraint ($2 \leq n \leq 5$).
3. To edit a miskeyed entry in steps 5 and 6, enter component no., press [E] and repeat these steps.
4. When NAM order = 2, a "?" is printed with frequency during linear sweep. This is a normal occurrence resulting from a two-node network being handled internally in three nodes during matrix reduction. When desired sweep is completed, execute step 10 and press [CE] to stop flashing.
5. For easy updating, record contents of bank 4 (reference – *Personal Programming,* pg. VII-1) when steps 5 and 6 are completed. Enter 4 in the display, press [2nd] [Write]. Load Input Card, Side 2. A "4" will appear in the display.

†These displayed values are printed.

*These values are printed only.

Example 1: For the passive RC filter shown, determine the frequency response from 1 MHz to 10 MHz at 1 MHz increments. Then continue the sweep from 10 MHz to 100 MHz in increments of 10 MHz.



ENTER	PRESS	DISPLAY	COMMENTS
9	[2nd] [Op] 17	239.89	Repartition
	[2nd] [CMs] [RST]	239.89	Initialize
	[2nd] [CP] [CLR]	0.	Prepare to load Input Card, side 1
		1.	Load card
3	[A]	3.†	Enter order of NAM
1000000	[R/S]	1000000.†	Enter starting f in hertz
1000000	[R/S]	1000000.†	Enter Δf in hertz
	[2nd] [E']	INPUT*	Initialize for input
		VALUE*	
		FM.TO*	
55	[B]	55.†	Enter R in ohms
1.2	[R/S]	1.2*	Enter nodal data
		1.†	
20	[B]	20.†	Enter R in ohms
3.0	[R/S]	3.0*	Enter nodal data
		2.†	
1	[E]	1.†	Change value of component 1
50	[B]	50.†	Enter new value

ENTER	PRESS	DISPLAY	COMMENTS
1.2	[R/S]	1.2*	Enter nodal data
.00159	[EE] 6 [+/–] [D]	1.59–09†	Enter C in farads
1.3	[R/S]	1.3*	Enter nodal data
		3.†	
3	[E]	3.†	Change nodal data of component 3
.00159	[EE] 6 [+/–] [D]	1.59–09†	Enter C
2.3	[R/S]	2.3*	Enter new nodal data
4	[2nd] [Write]	4.	Record bank 4. Insert Input card, side 2
	[2nd] [CP] [CLR]	0	Prepare to load Construct NAM card
		1.	Load card
	[GTO] [E] [LRN]	031 06	Enter learn mode
3	[LRN]	1.	For NAM order = 3
	[E]	See Tape 1**	Perform analysis
	[R/S] [RST]	–18.44946143	Terminate linear sweep and repartition
9	[2nd] [Op] 17	239.89	
	[2nd] [CP] [CLR]	0.	Prepare to load Input Card, side 1
		1.	Load card
	[CLR]	0	Prepare to load bank 4 data
		4.	Load Input Card, side 2
3	[A]	3.†	Enter order of NAM
10000000	[R/S]	10000000.†	Enter starting f in hertz
10000000	[R/S]	10000000.†	Enter Δf in hertz
4	[2nd] [Write]	4.	Record bank 4. Insert Input Card, side 2
	[2nd] [CP] [CLR]	0	Prepare to load Construct NAM Card
		1.	Load card
	[GTO] [E] [LRN]	031 06	Enter learn mode
3	[LRN]	1.	For NAM order = 3
	[E]	See Tape 2**	Perform analysis
	[R/S] [RST]	–2.045937026	Terminate linear sweep and repartition
9	[2nd] [Op] 17	239.89	

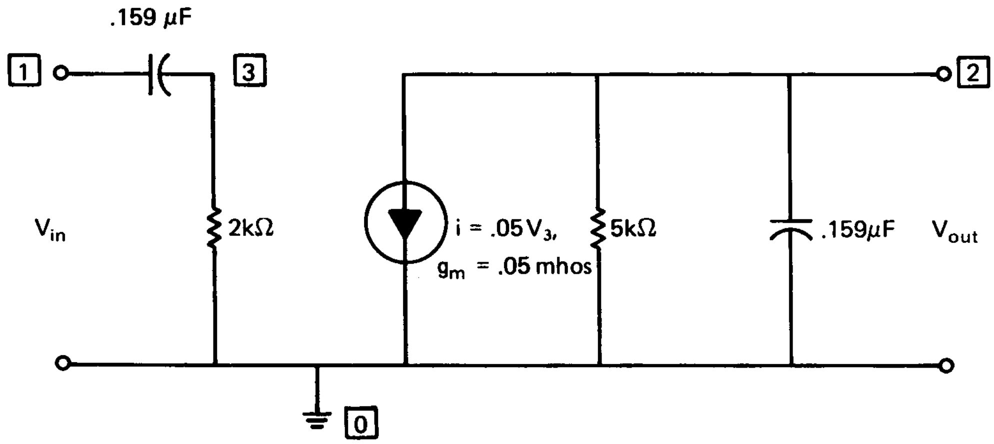
†These displayed values are printed
*These values are printed only
**Approximately 19 minutes is required for these calculations.

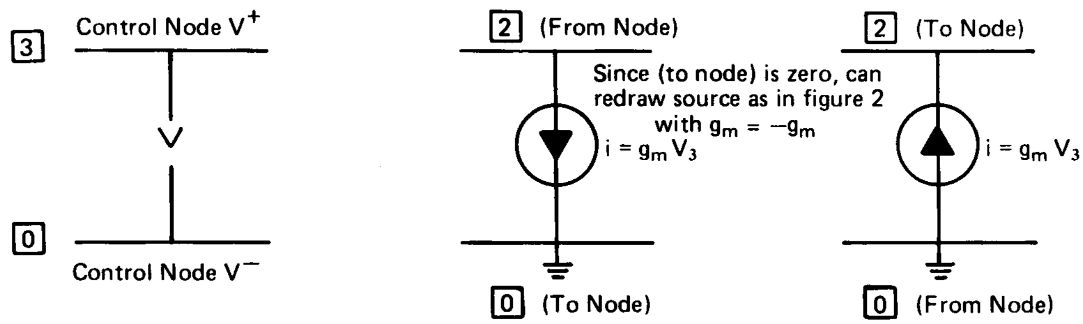
OUTPUT TAPE FOR EXAMPLE 1

Tape 1		Tape 2
1000000.	f(hertz)	10000000.
-1.559073999	mag (dB)	-9.998476016
-23.66660142	phase (degrees)	-18.44946143
2000000.		20000000.
-4.063903011		-10.63971792
-32.6537504		-9.959795447
3000000.		30000000.
-5.991926308		-10.77199574
-33.57584422		-6.742412384
4000000.		40000000.
-7.311881441		-10.81944351
-31.69590456		-5.084488094
5000000.		50000000.
-8.20826577		-10.84161403
-29.06775995		-4.077950198
6000000.		60000000.
-8.827545839		-10.85371347
-26.42794172		-3.40300624
7000000.		70000000.
-9.265698392		-10.86102831
-24.01851772		-2.91930633
8000000.		80000000.
-9.583468724		-10.8657837
-21.89613662		-2.555783497
9000000.		90000000.
-9.819429644		-10.86904753
-20.05033477		-2.272655976
10000000.		100000000.
-9.998476016		-10.87138391
-18.44946143		-2.045937026

Example 2: Examine the response of a common-emitter amplifier from 1 kHz to 10 kHz at increments of 1 kHz. The equivalent circuit representation is shown below.

COMMON-EMITTER AMPLIFIER (Equivalent Circuit Representation)





ENTER	PRESS	DISPLAY	COMMENTS
9	[2nd] [Op] 17	239.89	Repartition
	[2nd] [CMs] [RST]	239.89	Initialize
	[2nd] [CP] [CLR]	0	Prepare to load Input Card, side 1
		1.	Load card
3	[A]	3.†	Enter order of NAM
1000	[R/S]	1000.†	Enter starting f in hertz
1000	[R/S]	1000.†	Enter Δf in hertz
	[2nd] [E']	INPUT*	Initialize for input
		VALUE*	
		FM.TO*	
.000000159	[D]	.000000159†	Enter C in farads
1.3	[R/S]	1.3*	Enter nodal data
		1.†	
2000	[B]	2000.†	Enter R in ohms
3.0	[R/S]	3.0*	Enter nodal data
		2.†	
.05	[2nd] [A']	.05†	Enter source value in mhos
2.3	[R/S]	2.3*	Enter nodal data
		3.†	
5000	[B]		Enter R in ohms
2.0	[R/S]	2.0*	Enter nodal data
		4.†	

ENTER	PRESS	DISPLAY	COMMENTS
.000000159	[D]	.000000159†	Enter C in farads
2.0	[R/S]	2.0*	Enter nodal data
		5.†	
	[2nd] [CP] [CLR]	0	Prepare to load Construct NAM card
		1.	Load card
	[GTO] [E] [LRN]	031 06	Enter learn mode
3	[LRN]	1.	For NAM order = 3
	[E]	See Tape 3**	Perform analysis
	[R/S] [RST]	94.01206902†	Terminate linear sweep and repartition
9	[2nd] [Op] 17	239.89	

†These displayed values are printed.
*The values are printed only
**Approximately 17 minutes is required for these calculations.

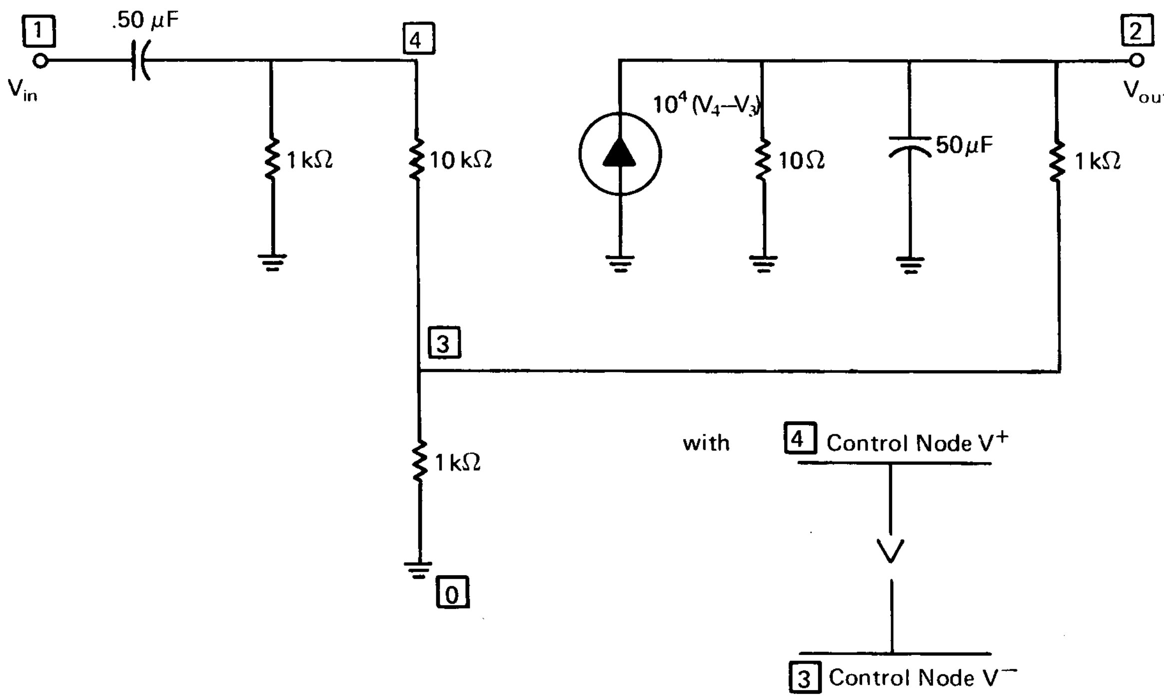
OUTPUT TAPE FOR EXAMPLE 2

Tape 3

```
        1000.  f (hertz)
  32.84640767  mag (dB)
  127.9080497  phase (degrees)

        2000.
  27.66017524
  109.7655013

        3000.
  24.30691736
  103.2891567

        4000.
  21.86833137
  99.99707806

        5000.
  19.95820588
  98.00896102

        6000.
  18.38988951
  96.67927389

        7000.
  17.06020418
  95.72775564

        8000.
  15.90637718
  95.01330156

        9000.
  14.88745217
  94.45719261

       10000.
  13.97525509
  94.01206902
```

Example 3: Using the following equivalent circuit, evaluate the gain of an active high-pass filter from 100 kHz to 1 MHz at increments of 100 kHz.

ACTIVE HIGH-PASS FILTER (Equivalent Circuit Representation)



ENTER	PRESS	DISPLAY	COMMENTS
9	[2nd] [Op] 17	239.89	Repartition
	[2nd] [CMs] [RST]	239.89	Initialize
	[2nd] [CP] [CLR]	0	Prepare to load Input Card, side 1
		1.	Load card
4	[A]	4.†	Enter order of NAM
100000	[R/S]	100000.†	Enter starting f in hertz
100000	[R/S]	100000.†	Enter Δf in hertz
	[2nd] [E']	INPUT*	Initialize for input
		VALUE*	
		FM.TO*	
.00005	[D]	.00005†	Enter C in farads
1.4	[R/S]	1.4*	Enter nodal data
		1.†	
1000	[B]	1000.†	Enter R in ohms
4.0	[R/S]	4.0*	Enter nodal data
		2.†	
10000	[B]	10000.†	Enter R in ohms
4.3	[R/S]	4.3*	Enter nodal data
		3.†	
1000	[B]	1000.†	Enter R in ohms
3.0	[R/S]	3.0*	Enter nodal data
		4.†	
10	[B]	10.†	Enter R in ohms
2.0	[R/S]	2.0*	Enter nodal data
		5.†	

ENTER	PRESS	DISPLAY	COMMENTS
1000	[B]	1000.†	Ėnter R in ohms
2.3	[R/S]	2.3*	Enter nodal data
		6.†	
10000	[2nd] [A']	10000.†	Enter source value in mhos
2.3	[R/S]	2.3*	Enter nodal data
		7.†	
10000	[+/–] [2nd] [A']	–10000.†	Enter source value in mhos
2.4	[R/S]	2.4*	Enter nodal data
		8.†	
.00005	[D]	0.00005†	Enter C in farads
2.0	[R/S]	2.0*	Enter nodal data
		9.†	
	[2nd] [CP] [CLR]	0	Prepare to load Construct NAM Card
		1.	Load card
	[GTO] [E] [LRN]	031 06	Enter learn mode
5	[LRN]	1.	For NAM order = 4
	[E]	See Tape 4**	Perform analysis
	[R/S] [RST]	–3.774268231	Terminate linear sweep and repartition
9	[2nd] [Op] 17	239.89	

†These displayed values are printe

*These values are printed only

**Approximately 39 minutes is required for these calculations

OUTPUT TAPE FOR EXAMPLE 3

Tape 4

```
     100000.   f (hertz)
 6.020227407   mag (dB)
-.3761627445   phase (degrees)

     200000.
 6.019660417
-.7550282678

     300000.
 6.018715592
-1.133220083

     400000.
 6.017393182
-1.511161289

     500000.
 6.015693532
-1.888910281

     600000.
 6.013617085
-2.266464726

     700000.
 6.011164382
-2.643805011

     800000.
 6.008336059
 -3.02090511

     900000.
 6.005132852
-3.397736221

    1000000.
 6.001555591
-3.774268231
```

Register Contents

R_{00}–R_{09}, R_{80}–R_{86}, R_{90}–R_{99} are work registers having multiple program uses.
R_{78} f
R_{79} Δf
R_{87} n
R_{88} Used
R_{89} Component Counter

NAM Register Assignments

(Admittance y_{ij} is the ij[th] complex element of the matrix)

NAM order = 5			NAM order = 4			NAM order = 3 or 2		
real y_{ij} register	matrix element i j	imag y_{ij} register	real y_{ij} register	matrix element i j	imag y_{ij} register	real y_{ij} register	matrix element i j	imag y_{ij} register
10	1 1	35	10	1 1	26	10	1 1	19
11	1 2	36	11	1 2	27	11	1 2	20
12	1 3	37	12	1 3	28	12	1 3	21
13	1 4	38	13	1 4	29			
14	1 5	39						
15	2 1	40	14	2 1	30	13	2 1	22
16	2 2	41	15	2 2	31	14	2 2	23
17	2 3	42	16	2 3	32	15	2 3	24
18	2 4	43	17	2 4	33			
19	2 5	44						
20	3 1	45	18	3 1	34	16	3 1	25
21	3 2	46	19	3 2	35	17	3 2	26
22	3 3	47	20	3 3	36	18	3 3	27
23	3 4	48	21	3 4	37			
24	3 5	49						
25	4 1	50	22	4 1	38			
26	4 2	51	23	4 2	39			
27	4 3	52	24	4 3	40			
28	4 4	53	25	4 4	41			
29	4 5	54						
30	5 1	55						
31	5 2	56						
32	5 3	57						
33	5 4	58						
34	5 5	59						
R_{60}–R_{77} used for storage of 9 component values			R_{50}–R_{77} used for storage of 14 component values			R_{30}–R_{77} used for storage of 24 component values		

Method Used

An electrical network with n nodes (excluding node 0, ground) can be specified for steady-state analysis by its n X n node admittance matrix (NAM) [Y] which is dependent on the specified frequency, network topology, and component values.

Starting with a null matrix (all elements are zero), the complex admittance matrix [Y] is formed by adding components to the null matrix as follows:

Let y = admittance of a given component

i and j = numbers of the nodes that the component is connected between (i is the from node ($i \neq 0$) and j is the to node)

Let y_{ij} be the element in the i^{th} row and j^{th} column of the NAM [Y] being formed.

Add y to y_{ii}, and if $j \neq 0$, add y to y_{jj}, add $-y$ to y_{ij} and $-y$ to y_{ji}.

Since the matrix element y_{ij} is a complex quantity dependent on frequency, each network component adds to either the real or imaginary part of y_{ij}.

For resistors $y = 1/R$ where R = resistance in ohms

For inductors $y = -(1/\omega L)j$ where L = inductance in henrys
$\omega = 2\pi f$ where f is frequency in hertz

For capacitors $y = \omega Cj$ where C = capacitance in farads

Once the complex NAM [Y] has been formed, the following matrix equation can be used to represent the relationship between node voltages and currents. This equation describes the steady-state performance of a network at a given frequency.

$$[Y]_{n \times n} [V]_{n \times 1} = [I]_{n \times 1}$$

where

$[V]_{n \times 1}$ is the matrix (column vector) of the node voltages.

$[I]_{n \times 1}$ is the matrix (column vector) of the current sources.

Solving for $[V]_{n \times 1}$:

$$[Y]^{-1}_{n \times n} [Y]_{n \times n} [V]_{n \times 1} = [Y]^{-1}_{n \times n} [I]_{n \times 1}$$

or

$$[V]_{n \times 1} = [Y]^{-1}_{n \times n} [I]_{n \times 1}$$

Since, for this program it is necessary to find only one node voltage, the complex matrix inversion technique required above is dropped in favor of a faster and more accurate matrix reduction method which can be applied to [Y] to find the network transfer function V_{out}/V_{in}.

At this point, all active elements (i.e., voltage controlled dependent current sources) present in the network should be added to [Y]. For each voltage controlled current source, compute the transconductance g_m. Add $-g_m$ to y_{ij} where i is the number of the node the controlled current is entering and j is the number of the controlling voltage V^+ of the source. If $j \neq 0$, add g_m to y_{ij} where i is the same as before and $j \neq 0$ is the number of the controlling voltage V^- of the source.

(Note that allowances for sources have to be made by the user at input. See User Instruction 5d and program reference 1., pp 9–11, 34, 35 and 2., pp 22, 23, 63–68.)

[Y] is now ready for matrix reduction which is outlined below.

To find the network transfer function $V_{out}/V_{in} = V_2/V_1$

where node 1 is the input node and node 2 is the output node

1. Partition $[Y]_{n \times n}$

$$[Y]_{n \times n} = \begin{bmatrix} [Y_{11}] & [Y_{12}] \\ [Y_{21}] & y_{nn} \end{bmatrix}$$

where

$[Y_{11}]$ is an $(n-1) \times (n-1)$ square submatrix

$[Y_{12}]$ is a column vector with $(n-1)$ elements

$[Y_{21}]$ is a row vector with $(n-1)$ elements

y_{nn} is the indicated element of the NAM [Y]

2. Reduce $[Y]_{n \times n}$ to a $(n-1) \times (n-1)$ matrix $[Y^{(1)}]_{(n-1) \times (n-1)}$ by the relation.

$$[Y^{(1)}]_{(n-1) \times (n-1)} = [Y_{11}] - [Y_{12}]\,[Y_{21}]\left(\frac{1}{y_{nn}}\right)$$

3. Partition the matrix $[Y^{(1)}]$ in a manner similar to that shown in 1 and repeat the reduction in 2 which produces the matrix $[Y^{(2)}]_{(n-2) \times (n-2)}$. Continue in this manner until a 2×2 matrix

$$[Y^{(n-2)}]_{2 \times 2} = \begin{bmatrix} y'_{11} & y'_{12} \\ y'_{21} & y'_{22} \end{bmatrix}$$ is derived.

The complex division $\dfrac{-y'_{21}}{y'_{22}} = \dfrac{V_2}{V_1}$ gives the required transfer function.

$$20 \log \left|\frac{V_2}{V_1}\right| = \text{magnitude in dB and}$$

$$\tan^{-1}\left(\frac{\text{Imag}\,(V_2/V_1)}{\text{Real}\,(V_2/V_1)}\right) = \text{phase in degrees where } -90° \leqslant \text{phase} < 270°$$

(See *Personal Programming,* pg. V-31)

References:

1. *Active Networks,* edited by Julius J. Hupert, published for Department of Physics, DePaul University by University Microfilms International 1978.
2. *Theory and Design of Active RC Circuits,* Lawrence P. Huelsman, McGraw-Hill 1968.
3. *Circuits Matrices and Linear Vector Spaces,* Lawrence P. Huelsman, McGraw-Hill 1963.
4. *Digital Computations in Basic Circuit Theory,* Lawrence P. Huelsman, McGraw-Hill 1968.

APPENDIX A: PROGRAM REFERENCE DATA

Program Number	Title	No. of Steps	Data Reg. Used	Flags Used	SBR Levels	Paren. Levels	Calls Pgm.	Special Functions Used	x≷t	ABS Address	Fix Decimal Format	Angular Mode	Program Number
01	Module Check	32	1–6						CP	X			01
02	Phase-Locked Loop	344	10–20	0,1,2	1	2	11		X	X			02
03	S-Parameter Conversions	273	1–17		2		4	P→R	CP	X		Deg.	03
04	Complex Arithmetic	212	1–4	1	2	1	5	P→R	X	X		Deg.	04
05	Complex Functions	146	1–4		2	2	4	P→R	X	X		Deg.	05
06	Complex Trig Functions	298	0–4	0	2	3	4,5	P→R	X	X		Deg.	06
07	dB, Np, P, V, I, Ratio Conversions	159	10–13	0	1	1				X			07
08	Signal Detection	459	10–18	0,1,2	2	2			X	X			08
09	Roots of a Polynomial	467	0–59	0,1		2			CP	X			09
10	Chained Multiplication of Polynomials	228	0–89	0,1	1	1			X	X			10
11	Reactance Chart	218	10–14	0,1,2		2				X			11
12	Series/Parallel Impedance Conversions	117	10–13	0	1	1	7			X			12
13	Active LP, HP, BP Filters	312	10–18		1	2			X	X			13
14	Passive Lowpass Filters	373	0–13	0	2	6			X	X			14
15	Convolution	178	8–17	0	1	2	0		X	X			15
16	Root Locus Calculations	333	0–59	0	2	1		P→R	X	X	Deg.		16
17	Discrete Fourier Transform	255	0–99	0	1	2		P→R	X	X	Deg.		17
18	Smith Chart Calculations	201	1–10,14–17, 20		3	2	4,5,11	P→R	X	X			18
19	Network Analysis	832*	0–99	0	4	3	0,4,19	P→R	X	X	Deg.		19

*Includes program memory

NOTES

ONE YEAR LIMITED WARRANTY FOR CALCULATOR AND/OR LIBRARY MODULE

WARRANTEE: This Texas Instruments Electronic Calculator Warranty extends only to the original purchaser of the calculator or module.

WARRANTY DURATION: This Texas Instruments Electronic Calculator and/or Library Module is warranted to the original purchaser for a period of one (1) year from the original purchase date.

WARRANTY COVERAGE: This Texas Instruments Electronic Calculator and/or Library Module is warranted against defective materials and workmanship. **THIS WARRANTY IS VOID IF: (I) THE CALCULATOR OR MODULE HAS BEEN DAMAGED BY ACCIDENT OR UNREASONABLE USE, NEGLECT, IMPROPER SERVICE OR OTHER CAUSES NOT ARISING OUT OF DEFECTS IN MATERIAL OR WORKMANSHIP, (II) THE SERIAL NUMBER HAS BEEN ALTERED OR DEFACED.**

WARRANTY PERFORMANCE: During the above one (1) year warranty period your defective calculator or module will either be repaired or replaced with a reconditioned model of an equivalent quality (at TI's option) when the calculator or module is returned, postage prepaid and insured, to a Texas Instruments Service Facility listed below. In the event of replacement with a reconditioned model, the replacement unit will continue the warranty of the original unit or 90 days, whichever is longer. Other than the postage and insurance requirement, no charge will be made for such repair, adjustment, and/or replacement.

WARRANTY DISCLAIMERS: ANY IMPLIED WARRANTIES ARISING OUT OF THIS SALE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO THE ABOVE ONE (1) YEAR. TEXAS INSTRUMENTS SHALL NOT BE LIABLE FOR LOSS OF USE OF THE CALCULATOR OR OTHER INCIDENTAL OR CONSEQUENTIAL COSTS, EXPENSES OR DAMAGES INCURRED BY THE PURCHASER.

Some states do not allow the exclusion or limitation of implied warranties or consequential damages, so the above limitations or exclusions may not apply to you.

LEGAL REMEDIES: This warranty gives you specific legal rights, and you may also have other rights that vary from state to state.

TEXAS INSTRUMENTS CONSUMER SERVICE FACILITIES

Texas Instruments Service Facility
P.O. Box 2500
Lubbock, Texas 79408

Texas Instruments Service Facility
41 Shelley Road
Richmond Hill, Ontario, Canada

Consumers in California and Oregon may contact the following Texas Instruments offices for additional assistance or information:

Texas Instruments Consumer Service
3186 Airway Drive Bldg. K
Costa Mesa, California 92626
(714) 540-7190

Texas Instruments Consumer Service
10700 Southwest Beaverton Highway
Park Plaza West, Suite 11
Beaverton, Oregon 97005 (503) 643-6758

IMPORTANT NOTICE REGARDING PROGRAMS AND BOOK MATERIALS

TEXAS INSTRUMENTS MAKES NO WARRANTY, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR PARTICULAR PURPOSE, REGARDING THESE PROGRAMS OR BOOK MATERIALS OR ANY PROGRAMS DERIVED THEREFROM AND MAKES SUCH MATERIALS AVAILABLE SOLELY ON AN "AS-IS" BASIS.

IN NO EVENT SHALL TEXAS INSTRUMENTS BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF THE PURCHASE OR USE OF THESE BOOK MATERIALS OR PROGRAMS AND THE SOLE AND EXCLUSIVE LIABILITY TO TEXAS INSTRUMENTS, REGARDLESS OF THE FORM OF ACTION, SHALL NOT EXCEED THE PURCHASE PRICE OF THIS BOOK. MOREOVER, TEXAS INSTRUMENTS SHALL NOT BE LIABLE FOR ANY CLAIM OF ANY KIND WHATSOEVER AGAINST THE USER OF THESE PROGRAMS OR BOOK MATERIALS BY ANY OTHER PARTY.

TEXAS INSTRUMENTS
INCORPORATED
DALLAS, TEXAS

Printed in U.S.A.

1014984-101