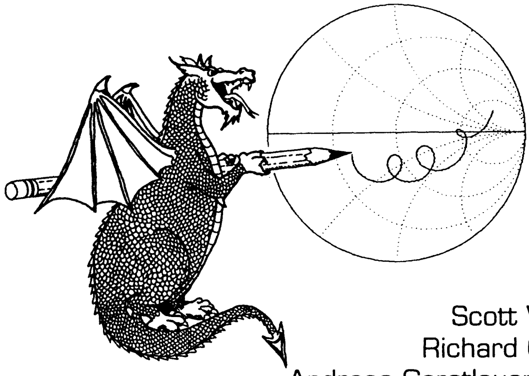

# APPENDIX C

# *Puff* 2.1

Scott Wedge, *Tanner Research, Inc.*
Richard Compton, *Lucent Technologies*
Andreas Gerstlauer, *University of California, Irvine*
David Rutledge, *California Institute of Technology*

*Puff* is a circuit simulator for linear circuits. It calculates scattering parameters and makes microstrip and stripline layouts. It also makes time-domain plots. The program is named after the magic dragon in the song by the popular American singing group Peter, Paul and Mary. *Puff* originated as a teaching tool for Caltech's microwave circuits course. It was created as an inexpensive and simple-to-use alternative to professional software whose high costs, copy protection schemes, and training requirements create difficulties in the academic environment. *Puff* uses a simple interactive schematic-capture type environment. After a circuit is laid out on the screen using cursor keys, a frequency or time domain analysis is available with a few keystrokes. This process is faster than using net lists, and errors are rare since the circuit is always visible on the screen. Intended for students and researchers, public distribution of the program began in 1987. *Puff* use, originally limited to Caltech, UCLA, and Cornell, has since spread to many other universities and colleges. The program has also become popular with working engineers, scientists, and amateur radio operators. Over 20,000 copies of versions 1.0, 1.5, 2.0, and 2.1 have been distributed worldwide, and translations have been made to Russian, Polish, and Japanese.

In this book, *Puff* is used for analyzing the filter and tuned transformer circuits in Chapters 5 and 6. Here we give an introduction to the use of the program, using the filters in Figures 5.4b and 5.8a as examples. Readers who are interested in more information about the program and its use in microwave circuits can get information about purchasing the complete manual at cost from our web site at http://www.its.caltech.edu/~mmic/puff.html. Lecture notes that give examples for using *Puff* in microwave circuits are also available at cost. You may write us at puff@caltech.edu or alternatively, write to Puff Distribution, MS 136-93, Caltech, Pasadena, CA, 91125.

## C.1 WINDOWS

*Puff* is a DOS program, and for a computer running Windows95, *Puff* runs best in the Restart MS-DOS mode. Click on Restart in MS-DOS mode as you shut down windows. There is an alternate way to run *Puff* using your old version of DOS if you installed Windows95 as an upgrade: Boot your computer. When it is almost finished booting and just about ready to open the Windows95 page (timing is essential), push the *F8* button. This will give you a screen with choices. You should choose to run your previous version of DOS. Then follow the instructions in the next section.

*Puff* will run in the MS-DOS Prompt window within Windows95, but the program runs many times slower than under DOS. To preserve compatibility with MS-DOS programs, Windows95 allows the user to alter certain settings before launching the program. Older versions of Windows achieved this by the use of a PIF, or program information file. A PIF was required to run each DOS program. Windows95 condenses this feature into the shortcut links that are used to launch applications.

Two problems exist when using Puff with Windows95. The Print Screen button in Windows95 is configured to paste a copy of the screen to the clipboard instead of sending a screen dump to the printer. The shortcut link will have to altered to return the Print Screen key back to its original function. Also, the proper screen dump routine needs to be executed before *Puff* is started up. Windows95 allows the user to run a short program, called a batch file, before starting up the main application. The shortcut link should be configured to have the screen dump routine listed as its batch file.

To set up the shortcut link to *Puff*, follow these steps:

1. Using Windows Explorer, go to the directory that *Puff* is located in. Click on the `puff.exe` file using the right mouse button and select Create Shortcut. This will place a shortcut link to the executable in the *Puff* directory.
2. Now right click on this shortcut and select *Properties*. This will bring up the dialog window that allows you to alter the MS-DOS settings for the shortcut link.
3. Click on the *Program* tab near the top of the window. Go to the entry for *Batch file:* and type in the name for the required screen dump routine. Printers that are compatible with the HP Laserjets can use the routine named `vga2lasr.com`.
4. Make sure the *Close on exit* box is checked at the bottom.
5. Click on the *Screen* tab at the top. In the *Usage* section, select *Full-screen*. *Puff* doesn't run within a window.
6. Click on the *Misc* tab. In the *Windows shortcut keys* section, make sure that *PrtSc* should not be checked. When this box is checked, pressing the Print Screen button will dump an image of the screen to Window's clipboard instead of directing the output to the printer port.
7. Click on the *OK* box to save the settings and close the properties dialog window. *Puff* can now be run by simply clicking on this shortcut link. The screen dump

routine that was entered as the batch file will run before *Puff* is started, and the Print Screen key will allow you to send a screen dump to the LPT1 printer.

Some computers require that you follow this additional instruction: Click the right mouse button on the Puff.exe shortcut. Click on *Properties;* click on *Advanced.* This will open the Advanced properties window. Check the box *Prevent MS DOS programs from detecting Windows.* By default, the *Suggest MS DOS mode as necessary* box should also be checked.

To run *Puff* inside Windows NT, run the program in a full-screen DOS box. Some users also partition the hard drive and install DOS as the operating system on one of the partitions. When you boot your computer, the boot manager will ask you which operating system you wish to use. You must reboot when you switch between operating systems.

In Windows NT, the *PrtSc* key can be used to send an image to the clipboard when *Puff* is running in a window. *Alt Enter* will reduce the full screen to a window. You can paste the clipboard image into MSPaint or MS Photo Editor, and crop and invert the colors before printing.

## C.2 GETTING STARTED

To run *Puff,* you need an IBM PC-compatible, running DOS 3.0 or later. In order to make a hard copy, you will need a screen-dump routine for your display and printer. Many printers are compatible with HP LaserJet printers. You should run the program `vgalasr.com` before you start *Puff.* A screen dump will then be possible using the *PrtSc* key. Alternatively, later DOS versions allow a graphics screen dump configured for your printer. For example, if your printer is compatible with an HP Deskjet, you can type `graphics DESKJET` to set up the screen dump.

To run the program, copy the files into a *Puff* directory on your hard disk. Put the copy in the logged disk drive, or with a hard disk, change to the *Puff* directory. Type `puff` and the return key. The computer will first display a system detect screen with a copyright notice and information concerning the hardware that *Puff* recognizes. After typing another key, the circuit file `setup.puf` will be loaded. *Puff* will automatically detect the type of display in use and set up a screen similar to Figure C.1. If something is wrong with the screen, edit the `setup.puf` file, listed in Table C.1, to try a different display type and override automatic selection. If it still does not work, check for the necessary graphics hardware. Once *Puff* has loaded, the word `setup` will appear on the screen to indicate that this file was used to start. You can specify a different starting file, say `yourfile.puf`, by typing `puff yourfile` when you start the program. *Puff* will add the extension `.puf` if none is given. If you specify a filename other than `setup` when starting, *Puff* will not show the system detect screen.

There are several files on the *Puff* diskette in addition to the screen dump programs. The program file `puff.exe` and overlay file `puff.ovr` must be present. In

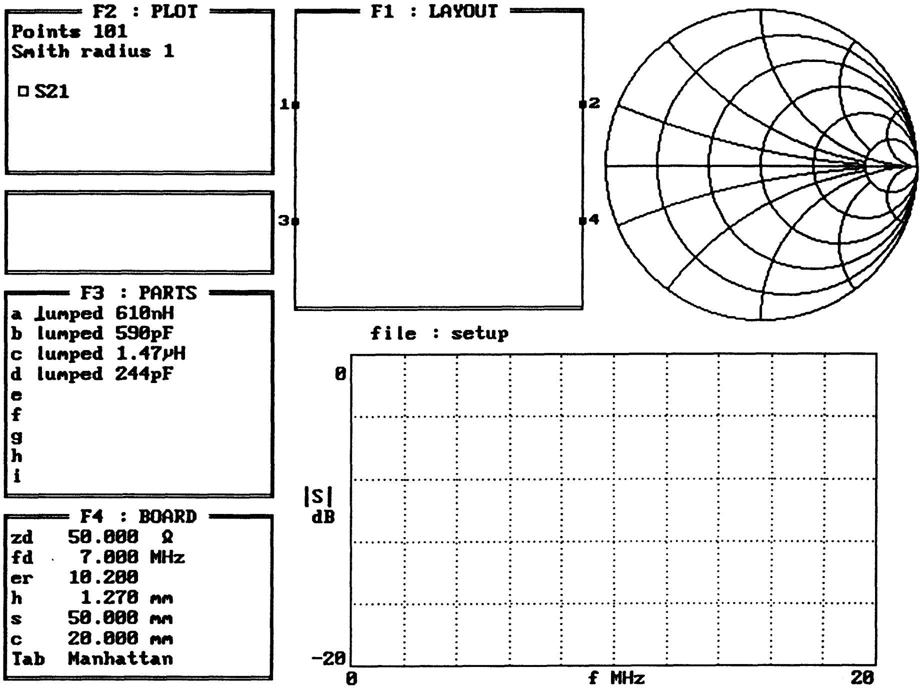


**Figure C.1.** A *Puff* screen dump.

addition, the `setup.puf` file should be available. Device files have the extension `.dev`, and there are several on your diskette, including a Fujitsu FHX04 HEMT transistor, `fhx04.dev`, a voltmeter, `vmeter.dev`, a voltage source, `vsource.dev`, a voltage-controlled current source, `cs.dev`, a unity gain differential amplifier, `da.dev`, and an operational amplifier with an open loop gain of 10,000, `op.dev`.

The *Puff* screen is subdivided into *windows*. Movement from window to window is accomplished by pressing function keys *F1* through *F4*. These keys are hot: They are always active. Other hot keys include *Esc*, which allows you to exit *Puff* and return to the operating system, *F10*, which toggles a small help window, and the cursor keys. At top center is the *Layout* window, where the circuit is constructed. This window is reached by typing the *F1* key. The numbers around the side of the circuit represent external connectors, or ports. There are two connectors on the left side, placed symmetrically above and below the center, and two on the right. The network appearing in the *Layout* window is analyzed from the *Plot* window. Located in the upper left corner of the screen, the *Plot* window is reached by typing *F2*. Here the scattering parameters are specified and their numerical values displayed. The rectangular and Smith chart plots are also controlled from the *Plot* window. The rectangular plot gives either a log-magnitude plot in the frequency domain or a linear impulse or step response in the time domain. The Smith chart, located at top right, gives a polar plot of the scattering coefficients. The circles inside the Smith chart are curves of constant resistance and reactance. Just below the *Plot* window, in the left center of the screen, is a three-line *Message*

**Table C.1.** Listing of `setup.puf`.

```
\b{oard} {setup.puf file for PUFF, version Electronics of Radio}
d 0 {display: 0 VGA or PUFF chooses, 1 EGA, 2 CGA, 3 Mono}
o 1 {artwork output: 0 dot-matrix, 1 LaserJet, 2 HPGL file}
t 2 {type: 0 microstrip, 1 stripline, 2 Manhattan}
zd   50.000 Ohms {normalizing impedance. zd>0}
fd    7.000 MHz  {design frequency. fd>0}
er   10.200      {dielectric constant. er>0}
h     1.270 mm   {dielectric thickness. h>0}
s    50.000 mm   {circuit-board side length. s>0}
c    20.000 mm   {connector separation. c>=0}
r     0.200 mm   {circuit resolution, r>0, Um=micrometers}
a     0.000 mm   {artwork width correction.}
mt    0.010 mm   {metal thickness, Um=micrometers.}
sr    0.000 Um   {metal surface roughness.}
lt    0.0E+00    {dielectric loss tangent.}
cd    5.8E+07    {conductivity of metal in mhos/meter.}
p     5.000      {photo reduction ratio. p<=203.2mm/s}
m     0.600      {mitering fraction. 0<=m<1}
\k{ey for plot window}
du    0  {upper dB-axis limit}
dl  -20  {lower dB-axis limit}
fl    0  {lower frequency limit. fl>=0}
fu   20  {upper frequency limit. fu>fl}
pts 101  {number of points, positive integer}
sr    1  {Smith-chart radius. sr>0}
S    21  {subscripts must be 1, 2, 3, or 4}
\p{arts window} {O=Ohms, D=degrees, U=micro, |=parallel}
lumped 610nH
lumped 590pF
lumped 1.47UH
lumped 244pF
```

*Note:* It is safest to edit a copy of this file, rather than the original. Be careful when you edit; if some of the board parameters are missing, the program will abort. Comments may be added at the end of the lines in braces, but they should not extend into the next line.

box, where error messages and requests for file names appear. Below the *Message* box is the *Parts* window, reached by typing *F3*. This window gives the current list of parts that may appear in the circuit. This list can only be edited from the *Parts* window. The equivalent physical dimensions of the *Layout* window are specified in the *Board* window, accessed by typing *F4*. The parameters given here also influence individual component dimensions. The two most important parameters are the normalizing impedance `zd`, and the design frequency `fd`. The normalizing impedance is used to calculate the scattering parameters. The design frequency is used to calculate the electrical length of transmission lines. Typing *F10* in any of the windows will bring up a small help screen. This screen will usually list the commands that are active for the current window. When in the *Board* window, however, *F10* will provide an explanation of the board parameters.

Maneuver through the *Puff* windows by pressing function keys *F1* through *F4*. In each of the windows, test the effects of typing and retyping the *F10* and *Tab* keys. When done, return *Puff* to its original state.

The spacing of the connectors in the *Layout* window is controlled by parameter `c` in the *Board* window. Hit *F4* to move to the *Board* window. Use the cursor keys and the number keys at the top of your keyboard to set `c` to zero. Type *F1* and see what happens to the connectors. When done, return `c` to its original value.

## C.3 SCATTERING PARAMETERS

For computer simulations with *Puff*, we need to extend the idea of reflection and transmission coefficients to filters by defining scattering parameters. The voltage waves going into the filter are called *incident* waves, and we use a plus subscript for them. The voltage waves leaving the filter are called *scattered* waves, and we use a minus subscript for them. We call them incident and scattered waves instead of forward and reverse waves because it is not always clear which direction is forward. The input and output connections are called *ports*. *Puff* allows up to four ports in a simulation. This allows two filters to be run at the same time for comparison. The ports are indicated by the subscripts 1 to 4, and they are all given the same resistance. In *Puff*, the port resistance is equal to the parameter `zd` in the `F4` *Board* window. We can use any of the ports as an input or output. For example, for a filter, we might use port 1 for the input and port 2 for the output. In Figure C.2, the input wave is $V_{+1}$, the reflected wave at the input is $V_{-1}$, and the output wave is $V_{-2}$. We can write the input and output powers in terms of these voltage waves. The input power $P_+$ is given by

$$P_+ = |V_{+1}|^2/(2R_s), \tag{C.1}$$

where $R_s$ is the source resistance. This is the available power from the source. The reflected power $P_-$ is given by

$$P_- = |V_{-1}|^2/(2R_s) \tag{C.2}$$

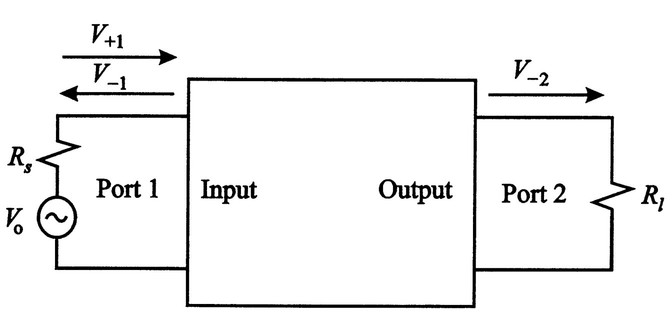


**Figure C.2.** A filter circuit with a source at the input (Port 1) and a load at the output (Port 2).

and the load power $P$ is

$$P = |V_{-2}|^2/(2R_l), \tag{C.3}$$

where $R_l$ is the load resistance.

The scattering parameters are complex quantities, and we have separate definitions for the magnitudes and phases. We write the magnitudes as

$$|s_{11}|^2 = P_-/P_+, \tag{C.4}$$

$$|s_{21}|^2 = P/P_+ \tag{C.5}$$

and the phases as

$$\angle s_{11} = \angle V_{-1} - \angle V_{+1}, \tag{C.6}$$

$$\angle s_{21} = \angle V_{-2} - \angle V_{+1}, \tag{C.7}$$

where $s$ is a *scattering parameter,* and the subscripts are port numbers. Scattering parameters with equal indices like $s_{ii}$ are reflection coefficients, and those like $s_{ji}$, where $i$ and $j$ are different port numbers, are transmission coefficients from port $i$ to port $j$. $|s|^2$ gives the proportion of power transmitted or reflected, while $\angle s$ gives the phase difference between a scattered voltage wave and an incident voltage wave. *Puff* makes two plots of the scattering parameters. One is a Smith-chart plot in the complex plane. The other is a dB plot, given as

$$|s_{11}| = 10\log(P_-/P_+)\,\text{dB}, \tag{C.8}$$

$$|s_{21}| = 10\log(P/P_+)\,\text{dB}. \tag{C.9}$$

People use the same letter $s$ whether it is expressed in dB or as a complex number. This makes it important to include the dB units, particularly since the complex scattering parameter has no units itself. Expressed in dB, $|s_{21}|$ is just the power gain, although we will say loss if it is negative.

## C.4 EXAMPLES

As an example, we analyze the filter in Figure 5.4b. We begin with the screen in Figure C.1. Hit the *F3* key to make the *Parts* window active. This is indicated by the flashing cursor for part a, and the highlighted *F3* at the top of the window. Now hit *F1* to move to the *Layout* window. A large white × will appear in the center of the circuit board, and the first line in the *Parts* window will become highlighted. Part a is a lumped 610-nH inductor. Push ←, and *Puff* will draw a hollow blue rectangle, labeled a, to the left, and the × will move to the other end of the inductor. In the *Message* box, Δx -5.00mm will appear. This shows you the change in the $x$ coordinate. When no length is given for a lumped part, *Puff* assumes it to be one tenth the size of the layout. Type 1 to connect to the first port. You should use the number key at the top of the keyboard rather than the numeric keypad. In fact, the *NumLock* key is disabled when *Puff* runs, to avoid mix ups between arrows and digits. *Puff* will make a gray outline path up and to the left to the first connector. Push →, and *Puff* will move back to the other end of the inductor. Type b to select the capacitor, and ↓ to draw it. Now hit the =

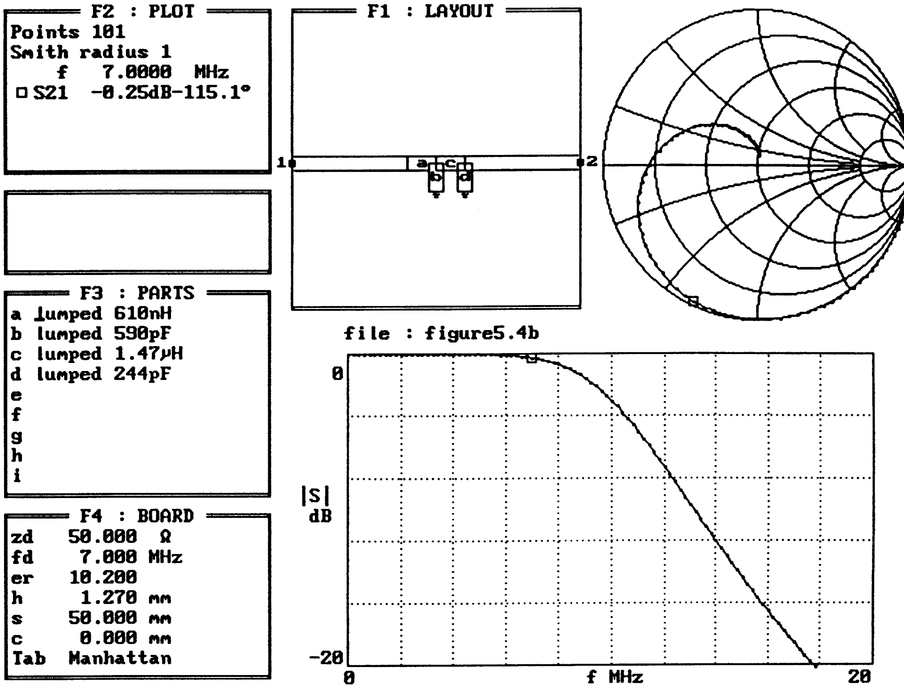


**Figure C.3.** Analyzing the filter in Figure 5.4b in *Puff*.

key to ground the end of the capacitor and type ↑ to move back to the other end. Without the ground, *Puff* thinks that the end of the capacitor is open-circuited. Now type `c` to select the second inductor, and → to draw it. Type `2` to connect to the second port. Type `d` to select the second capacitor, and ↓ to draw it. Hit the = key to ground the end of the capacitor. This completes the filter in Figure 5.4b.

When you type a key, *Puff* first checks for a valid keystroke. If it is not, *Puff* will beep and do nothing else. You can press `z` to hear the beep. Next, *Puff* checks to see if it can carry out the command. If it cannot, it will give an error message in the red *Message* box. For example, if you retype `1` in the present circuit, *Puff* will say, `Port 1 is already joined`, because multiple connections to the same port are not allowed.

This circuit is ready to be analyzed. Push *F2* to go to the *Plot* window, and type `p` to plot. The calculated $s_{21}$ values will appear in complex form on a Smith chart as small dots joined by a cubic spline curve. A square marker indicates the transmission coefficient at the design frequency, and the numerical values are given in the *Plot* window. Any magnitude greater than 100 dB will be reported as ∞, and any magnitude as small as −100 dB will be listed as zero. When *Puff* finishes, the *Message* box indicates the time required for the calculation. The rectangular graph shows the loss of the filter. It is a low-pass filter with 3-dB frequency of 10 MHz. You can see this by pushing the *PageUp* key repeatedly to move the cursor to higher frequencies until the loss listed in the *Plot* window reaches 3 dB.

To plot the reflection coefficient, push *F2* to go to the *Plot* window. Use the ↑ and ↓ keys to move to the line below `S21`. For the input reflection coefficient, $s_{11}$, type 11. Now push p to plot the reflection and transmission coefficients. To find the input impedance, use the ↑ and ↓ keys to move to the S11 line, and type =. The real and imaginary parts of the input impedance, and the equivalent inductance or capacitance, will appear in the message box below the plot window.

Go into the *Layout* window and make sure that the × is at the right of part c. Hold down the shift key and type ←. Such a shift-cursor operation in the direction of a part will cause it to be erased. Hold down the shift key and type 1 to erase the path to the connector. What happens if the shift-cursor operation is not in the direction of a part?

Erase the entire layout by typing *Ctrl*-e from either the *Parts* or *Layout* window. Now we can analyze the band-pass filter in Figure 5.8a. Type *F3* to return to the *Parts* window. Use the *Delete* key and backspace keys to erase the part descriptions. Then type the following on line *a*:

```
l 15μH + 34.5pF
```

The character $\mu$ is typed as *Alt*-m. The plus sign indicates a series connection. Any sequence of letters without numbers that starts with l and ends with a space indicates a lumped part. Now type the following on line *b*:

```
l 86nH || 6.0nF
```

The character || is typed as *Alt*-p. It indicates a parallel connection. Return to the *Layout* window, and draw the circuit as it appears in Figure C.4. Now move to

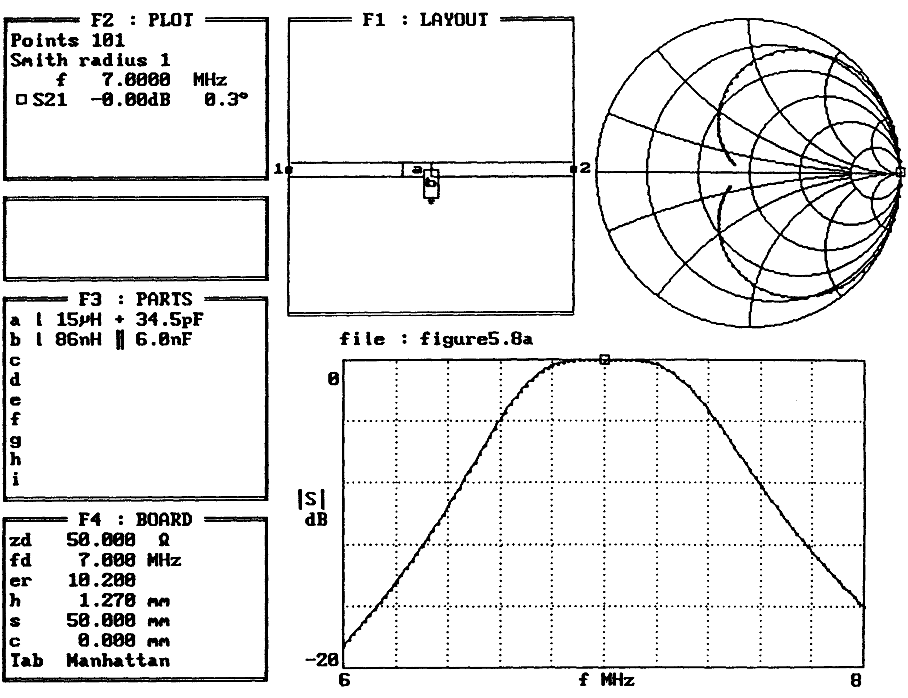


**Figure C.4.** Analyzing the filter in Figure 5.8a in *Puff*.

the *Plot* window. This is a band-pass filter, and we need to change the frequency scale to see its behavior clearly. To do this, use the up and down cursor keys to move the cursor to one of the frequencies at the bottom of the rectangular plot. Now type over the frequencies to narrow the plotting range from 6 MHz to 8 MHz. Push `p` again to plot. This should give the plot shown in Figure C.4. This shows the roll-off down to the 20-dB range. To see the behavior near the pass band, you can change the dB-scale to go from 0 to −3 in the same way that you changed the frequencies. Use the *PageUp* and *PageDown* keys to find the 3-dB bandwidth.

## C.5 PARTS WINDOW

The program begins in the *Parts* window. The initial parts list is taken from the setup file. A previously saved circuit file can be read in by typing *Ctrl*-`r`. *Puff* will prompt for a file name and will add `.puf` if no extension is given. You can edit the parts list with the arrow keys, the backspace key, the carriage return (or enter) key, *Ins*, and *Del*. The first letter on each line, from `a` to `i`, identifies each part for use in the layout. By hitting the *Tab* key, the parts list may be doubled in size, allowing extra parts `j` through `r` to be defined. Hitting the *Tab* key a second time will shrink the parts list back to its normal size, unless an extra part was defined and used in the layout. The types of parts available in *Puff* are listed Table C.2, including examples of their use. Only the first letter of a part name is needed for it to be recognized: `l` is equivalent to `lumped`, `t` is equivalent to `tline`, `q` is equivalent to `qline`, etc. Free up space in the parts list by using single-character part descriptions. Commas and periods are both treated as decimal points. Spaces are ignored, except around `device` and `indef` file names. When you leave the *Parts* window, the list will be checked. There will be an error message if a part is longer than the board or shorter than the circuit resolution `r` given in the circuit file. *Puff* will redraw the circuit if it has changed and report an error if changes result in a part being drawn off the board.

The simplest part available in *Puff* is the attenuator, written as `atten`. It is drawn as an open blue square with red dots denoting its two ports. It is ideal; *Puff* will always consider it as matched to the normalizing impedance `zd`, and its attenuation will be frequency independent. The desired amount of attenuation is entered in `dB`. If no units are entered, *Puff* will assume dB. The `atten` part is reciprocal and symmetrical.

Plot the scattering parameters for an `atten` part that has a negative value of attenuation. Why is this different than that for an ideal amplifier?

The `xformer` part denotes a lossless and frequency-independent transformer. The only numerical parameter needed is the (unitless) turns ratio. *Puff* accepts a single real number for the turns ratio, not a ratio of two numbers. However, a colon may be used to write the number as a ratio to one. Therefore, `1.5:1` is a valid entry, whereas `3:2` is not. Because the transformer is antisymmetric, it is drawn in the layout as a trapezoid. Red dots are again used to denote the ports. For an $n:1$ transformer, the wide end of the trapezoid represents the $n$ side, while

**Table C.2.** Descriptions and Examples of the Parts Available in *Puff*.

| Part | Description and Examples |
|---|---|
| atten | Ideal attenuator. Enter attenuation in dB:<br>`atten 3dB` |
| xformer | Ideal transformer. Enter turns ratio:<br>`xformer 2:1` |
| lumped | Resistor, capacitor, and/or inductor:<br>`lumped 50Ω‖1 nH‖1pF` {parallel RLC}<br>`lumped 10+j10−j10Ω` {series RLC resonant at `fd`} |
| tline | Ideal transmission line. Enter impedance and length:<br>`tline 50Ω 90°` {λ/4 length at `fd`}<br>`tline1.0z 5mm +1.0h` {artwork length correction added}<br>`tline! 50Ω 90°` {analyze with advanced models} |
| qline | Finite $Q$ `tline`:<br>`qline 50Ω 90° 75Qd` {$Q = 75$ due to dielectric losses}<br>`qline 20mS 90° 50Qc` {$Q = 50$ due to conductor losses} |
| clines | Ideal coupled transmission lines. Enter $Z_e$, $Z_o$, and length:<br>`clines 60Ω 90°` {$Z_e$ specified}<br>`clines 60Ω 40Ω 6mm` {$Z_e$ and $Z_o$ specified}<br>`clines! 60Ω 90°` {analyze with advanced models} |
| device | Read file with *s*-parameter data. Specify filename:<br>`device fhx04` {read filename `fhx04.dev`}<br>`device atf131.s2p` {read file in EEsof format} |
| indef | Indefinite *s*-parameter generator. Converts *n* port *s*-parameter file to $n + 1$ port device. Specify filename as in `device`. |

the narrow end represents the 1 side. If a negative turns ratio is entered, *Puff* will add a 180° phase shift to the transmission parameters.

Lay out and plot the scattering parameters for an ideal 180° phase shifter using an `xformer` with a `-1:1` turns ratio. How would the schematic diagrams differ for a positive and negative turns ratio?

The `lumped` part is used to specify series and parallel combinations of resistance, capacitance, and/or inductance. It is drawn as an open blue rectangle. *Puff* understands four units for impedance and admittance: Ω (ohms, typed as *Alt*-o), `S` (siemens), `z` (normalized impedance), and `y` (normalized admittance). For example, a 100-Ω resistor may be specified as `100Ω`, `0.01S`, `2z`, or `0.5y`, assuming that `zd` is 50 Ω. Capacitance and inductance values are entered in units of farads (`F`) and henries (`H`), respectively. Note that these must be upper case. Resistance, capacitance, and inductance values may be positive, zero, or negative. Reactance values are specified by placing a `j` before or after a number, and using the impedance units Ω or `z`. For example, `25j`Ω and `0.5jz` specify a 25-Ω reactance at `fd`. *Puff* scales positive reactances proportionally and negative reactances inversely with frequency, interpreting them as inductive and capacitive reactances, respectively. This means that at `2fd`, $j50\Omega$ becomes $j100\Omega$, and $-j50\Omega$ becomes $-j25\Omega$.

Specifications with the admittance units `s` and `y` are treated in a dual way: Positive susceptances scale proportionally with frequency, and the negative susceptances scale inversely.

Series circuits with two or three different lumped elements are specified by combining a real number, a positive imaginary number, and a negative imaginary number, all in the same `lumped` part. For example, `1+j10-j10z` specifies a circuit that is resonant at the design frequency with a $Q$ of 10. The resistance is equal to `zd` and the inductive and capacitive reactance are 10 `zd` at the design frequency. Note that the unit appears only once, after all the numbers. This type of lumped element entry is very convenient, for example, when taking values from filter tables. Parallel circuits are specified in a dual way using admittance units `s` and `y`. Series and parallel lumped elements that combine resistance and values of inductance and capacitance may also be formed. *Puff* has a special character for forcing a `lumped` part to be a parallel circuit when it does not contain admittance units. This character is ‖, typed as *Alt*-`p`, and it is referred to as the parallel sign. When used with a `lumped` part, the parallel sign forces the part's description to be interpreted as a parallel circuit, regardless of the unit. This allows part entries such as `50Ω‖1nH‖1pF` for a parallel RLC circuit. Note that you can use symbols for the engineering prefixes *nano* and *pico*. These symbols may appear in front of units in the parts list, in `.puf` files, and in some *Board* window parameters. *Puff* recognizes the all the unit prefixes given in Table 1.1b. Note that each symbol is case sensitive (there is a big difference, for example, between `10mΩ` and `10MΩ`). The $\mu$ symbol is obtained by typing *Alt*-`m`.

Ground one terminal and compare the one-port scattering parameters for the following lumped parts:

```
lum 50Ω 1nH 1pF
lum 50Ω‖1nH‖1pF
lum 50Ω +1nH +1pF
lum 50Ω -1nH -1pF
lum 50Ω‖-1nH‖-1pF
```

Use *Alt*-`p` to obtain the ‖ sign and *Alt*-`o` to obtain the Ω sign. Use the *Tab* key, while in the *Plot* window, to align your plots with the Smith-chart circles.

The `atten`, `xformer`, and `lumped` parts do not have electrical length. When included in a circuit they are drawn in a default length, referred to here as the *Manhattan* length. An alternate length may be entered for the `lumped` part, although it is optional. This comes in handy when trying to align parts in the layout. The `lumped` length specification should come last, and the units must be in meters (`m` with an appropriate prefix). The *Manhattan* length is one tenth the size of the layout. All parts using *Manhattan* dimensions will be drawn with the same spacing between terminals.

The `tline` part is an ideal transmission line section. It is lossless and without dispersion. In the layout, it is drawn as a copper rectangle. The characteristic impedance or admittance of a `tline` may be specified in the same way as the resistance of a `lumped` part, except that it must be positive. Length units are required. Valid units include: meters (`m` with an appropriate prefix), `h` (the substrate thickness), and ° (degrees, typed as *Alt*-d). A transmission line specified with a 360° length will be one wavelength long at the design frequency `fd`. It is usually convenient to specify degrees other than, say millimeters (mm), but sometimes the physical units are useful for aligning a `tline` with other parts. You may also specify an artwork length correction to compensate for discontinuities. This is done by adding a plus or a minus sign and a second length. These corrections are made on the screen and in the artwork, but they do not affect the electrical length used in the analysis. For example, the length `90°-0.5h` will be treated in the analysis as a quarter-wavelength section, but it will be drawn shorter on the screen and in the artwork. The `h` units are particularly useful here because open-circuit end corrections and phase shifts in tee junctions are proportional to the substrate thickness.

Often it happens that a desired electrical length results in a physical length larger than the size of the board or is perhaps negative for a de-imbedding problem. *Puff* provides for this eventuality by allowing you to force *Manhattan* dimensions upon the `tline`. This is done by placing an `M` (must be upper case) as the last character in the part's description. Then, regardless of the parameters specified, *Puff* will make the `tline` length one tenth the size of the layout. To find out the physical dimensions of a `tline`, place the cursor on the part, and hit the = key. *Puff* will calculate the dimensions for the part and list them in the *Message* box. The dimensions listed will include any artwork corrections or will be Manhattan dimensions if these were called for. Manhattan dimensions may be forced on all parts using the *Tab* key in the *Board* window.

The `qline` is similar to the `tline` but is lossy. In addition to impedance and length, you can add a value for the quality factor, or Q. The attenuation in the line is calculated by enforcing the given Q at the design frequency `fd`. Outside the design frequency, the attenuation is made to follow one of two models: Specify `Qd` or `Q` if you wish a dielectric loss model, and `Qc` if you prefer a conductor loss model. Manhattan dimensions are requested by placing an `M` (must be upper case) as the last character in the part's description. Artwork corrections are not allowed for the `qline`.

The `clines` part is a pair of coupled transmission lines. Its specification is similar to a `tline`, except that either one or two impedances or admittances may be given. If only one appears, then the specification looks the same as a `tline`. If this impedance is larger than `zd`, *Puff* interprets it as the even-mode impedance, and if smaller, as the odd-mode impedance. The other mode impedance is chosen to match the lines by forcing the product of even- and odd-mode impedances to be $zd^2$. If two impedances are given, the larger is the even-mode and the smaller the odd-mode impedance. As with the `tline`, use an upper case `M` at the end of a `clines` specification to force Manhattan dimensions.

The `device` part is used to read in a file containing multiport scattering parameters. Files may contain transistor data, measured data to be plotted, or parameters that define idealized parts, meters, or sources. A file name specifying the *s*-parameters must be given, preceded and followed by a space. *Puff* will assume a `.dev` extension if none is given. An optional length may be specified after the file name, as with the `lumped` part. The `device` is drawn in the *Layout* window as a

**Table C.3.** The File for the Fujitsu FHX04 HEMT transistor `fhx04.dev`. The frequencies are in GHz.

| {FHX04 Fujitsu HEMT (89/90), f = 0 extrapolated; Vds = 2V, Ids = 10mA} | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| f | s11 | | s21 | | s12 | | s22 | |
| 0.0 | 1.000 | 0.0 | 4.375 | 180.0 | 0.000 | 0.0 | 0.625 | 0.0 |
| 1.0 | 0.982 | -20.0 | 4.257 | 160.4 | 0.018 | 74.8 | 0.620 | -15.2 |
| 2.0 | 0.952 | -39.0 | 4.113 | 142.0 | 0.033 | 62.9 | 0.604 | -28.9 |
| 3.0 | 0.910 | -57.3 | 3.934 | 124.3 | 0.046 | 51.5 | 0.585 | -42.4 |
| 4.0 | 0.863 | -75.2 | 3.735 | 107.0 | 0.057 | 40.3 | 0.564 | -55.8 |
| 5.0 | 0.809 | -92.3 | 3.487 | 90.4 | 0.065 | 30.3 | 0.541 | -69.2 |
| 6.0 | 0.760 | -108.1 | 3.231 | 75.0 | 0.069 | 21.0 | 0.524 | -82.0 |
| 7.0 | 0.727 | -122.4 | 3.018 | 60.9 | 0.072 | 14.1 | 0.521 | -93.6 |
| 8.0 | 0.701 | -135.5 | 2.817 | 47.3 | 0.073 | 7.9 | 0.524 | -104.7 |
| 9.0 | 0.678 | -147.9 | 2.656 | 33.8 | 0.074 | 1.6 | 0.538 | -115.4 |
| 10.0 | 0.653 | -159.8 | 2.512 | 20.2 | 0.076 | -4.0 | 0.552 | -125.7 |
| 11.0 | 0.623 | -171.1 | 2.367 | 7.1 | 0.076 | -10.1 | 0.568 | -136.4 |
| 12.0 | 0.601 | 178.5 | 2.245 | -5.7 | 0.076 | -15.9 | 0.587 | -146.4 |
| 13.0 | 0.582 | 168.8 | 2.153 | -18.4 | 0.076 | -21.9 | 0.611 | -156.2 |
| 14.0 | 0.564 | 160.2 | 2.065 | -31.2 | 0.077 | -28.6 | 0.644 | -165.4 |
| 15.0 | 0.533 | 151.6 | 2.001 | -44.5 | 0.079 | -36.8 | 0.676 | -174.8 |
| 16.0 | 0.500 | 142.8 | 1.938 | -58.8 | 0.082 | -48.5 | 0.707 | 174.2 |
| 17.0 | 0.461 | 134.3 | 1.884 | -73.7 | 0.083 | -61.7 | 0.733 | 163.6 |
| 18.0 | 0.424 | 126.6 | 1.817 | -89.7 | 0.085 | -77.9 | 0.758 | 150.9 |
| 19.0 | 0.385 | 121.7 | 1.708 | -106.5 | 0.087 | -97.2 | 0.783 | 139.1 |
| 20.0 | 0.347 | 119.9 | 1.613 | -123.7 | 0.098 | -119.9 | 0.793 | 126.6 |

blue arrowhead with red dots denoting the ports. The dot at the wide end of the arrowhead represents port 1 in the file, and the remaining ports follow at equal intervals along the arrow's axis.

If you wish to make your own `device` files, you should study the format of the file `fhx04.dev` given in Table C.3. There is an optional comment line in braces, followed by a template line. The `f` at the beginning of the template stands for frequency. If left out, *Puff* assumes that the scattering coefficients are independent of frequency. The scattering parameters that follow the frequency in the template may appear in any order, and *Puff* will assume that any parameters that are not given on the template line are zero. The program will figure out how many ports the device has from the highest port number that appears in the template line. *Puff* can handle up to four-port device files. The numbers that follow the template are separated by one or more spaces or carriage returns. The frequency is first (if it appears) followed by the magnitude (linear, not dB) and phase (in degrees) of each of the scattering parameters in turn. When *Puff* is calculating scattering parameters in the *Plot* window, it will interpolate linearly between points in the `device` file, as necessary. *Puff* will not extrapolate beyond a `device` file's frequency range and it will give an error when this is attempted. A previously saved `.puf` file (that includes saved scattering parameters) can also be recalled as a `device` file. Complex networks can then be formed by combining many smaller circuits.

The `indef` part is similar to the `device`, but it is used to generate *indefinite* scattering parameters from a file containing definite scattering parameter data. Indefinite parameters are those with an undefined ground terminal. If a one-port file is specified, `indef` will convert it to a two-port. If a two-port file is specified, `indef` will convert it to a three-port; and so on up to a four-port to five-port conversion. The `indef` part is most often used to model a transistor as a three-port. The $n$ port to $n+1$ port conversion is made possible by assuming that Kirchhoff's current law is valid, allowing what was the ground terminal to be converted to a port. The `indef` part is drawn on the screen in the same way as the `device` part, except that the port created from ground is drawn as a yellow dot. The extra port generally appears as the last port, except for the two-port to three-port `indef` where the extra port appears in the center of the part. Note that to turn an `indef` part into its `device` equivalent, the extra port should be shorted.

In addition to the `device` file format given in Table C.3, *Puff* can also read *s*-parameter data files in the EEsof format. This is done when the appropriate file extension is given for `device` or `indef` parts: A one-port file requires a `.S1P` extension, the two-port uses `.S2P`, and so on up to a four-port `.S4P`. *Puff*, however, cannot read every possible data format for these files. It is assumed that they possess the following format:

```
# xHZ S MA R yy
```

where $x$ is the same engineering prefix used for `fd` in the *Board* window, and *yy* is the value for `zd`. *Puff* reads only scattering parameters given in the magnitude/angle format. These restrictions require some caution. If a device file contains frequencies given in GHz, *Puff* will give erroneous results if you try to plot data in MHz. The prefix for `fd` and the impedance of `zd` in the *Board* window *must* coincide with those used in all `device` and `indef` files. This goes for files in both EEsof and `.dev` formats. Noise parameters present in files are ignored.

Create a two-port circuit and plot all of its scattering parameters ($s_{11}$, $s_{21}$, $s_{12}$, and $s_{22}$). Use *Ctrl*-`s` from the *Plot* window to save it to a file. Erase the circuit, and then make a `device` part that will recall the scattering parameters from the previously saved file. Be sure to use the `.puf` extension. What happens if the number of points is different?

## C.6 LAYOUT WINDOW

In the upper portion of the screen is the *Layout* window. The square represents the substrate, and the numbers on the sides represent connectors. Typing an arrow key will draw the selected part in the *Parts* window in the direction of the arrow. The *Message* box will show the change in the $x$ and $y$ coordinates. *Puff* starts out drawing part `a`, but you can select another part in the list by typing the letter for the part desired. The circuit can be grounded at any point by pushing the = key.

If there is already a part in the direction of the arrow key, *Puff* will move to the other end rather than draw over it. If the ends of two parts are closer together than the circuit resolution `r`, *Puff* will connect them together. *Puff* will stop you from drawing a part off the edge, but it will not stop you from crossing over a previous part. You can make a path to a connector by pushing one of the number keys `1`, `2`, `3`, or `4` on the top row of the keyboard. Notice that *Puff* does this by first moving up or down and then right or left. The electrical length of a connector path is not taken into account in the analysis, and it is drawn in a different color to indicate this. You can erase the entire circuit and start over by pushing *Ctrl*-`e`. *Ctrl*-`n` moves the × to the nearest node. This can be useful if you are off the network and want to get back, or if you want to see if two nodes are connected.

The *Shift* key is used for erasing and moving around the layout. *Puff* will erase a part rather than move over it if you hold the *Shift* key down while pushing an arrow key. A ground can be removed with *Shift* =. The path to a connector is erased by holding the *Shift* key down while typing the connector number. If you are not at a connection to a port, this shift-number operation will move you to that port number without drawing the path. This is useful if you would rather start a circuit at a connector than in the center. The shift-arrow operation moves the × when there is no part to erase. The × moves half the length of the currently selected part. Half steps are used rather than full steps to allow centered and symmetrically laid out circuits. To move the full length of the part, step twice. It is important to note that this shift-arrow operation is actually drawing an invisible part. If you later change the size of an invisible part, it will have an impact on the layout, possibly resulting in a part being drawn off the board. If this happens, *Puff* will give you an error message that you may not believe. As a precaution, keep the number of invisible parts to a minimum.

There are special rules for drawing `clines`. Use the arrows to move along the lines and *Ctrl*-`n` to jump from one line to another. When connecting `clines` together, if you draw the second `clines` in the same direction as the first, the new lines will join the previous pair. This is the usual arrangement in a directional coupler. If you change directions, the `clines` will be staggered so that only one of the lines in each pair is connected. This is appropriate for a band-pass filter.

Typing *Ctrl*-`a` from the *Plot* window will activate *Puff*'s photographic artwork routines. The layout produced will be magnified by the photographic reduction ratio (`p`) in the circuit file. The artwork output parameter (`o`) in the circuit file allows you to specify dot-matrix or HP LaserJet printouts or the production of a Hewlett-Packard Graphics Language (HP-GL) file. *Puff* will prompt for titles to be placed atop the printout, or for an HP-GL file name. Only the `tlines`, `qlines`, and `clines` will appear in the artwork, and the corners will be mitered. You may adjust `tline` and `clines` lengths using discontinuity corrections in the parts list. Widths may be adjusted using the artwork width correction factor `a` in the circuit file.

You can inspect *Puff*'s layout calculations from the *Parts* window. Place the cursor in any `tline`, `qline`, or `clines` description and hit the = key. *Puff* will tell you the length and width for these parts, as well as the spacing for `clines`.

Lay out a simple circuit consisting of `tlines` and `clines`. Go to the *Board* window and use the *Tab* key to change the circuit type. Return to the *Layout* window and see how the circuit is affected. Repeat for microstrip, stripline, and Manhattan layouts. Can you make a microstrip circuit that is difficult to realize in stripline? What stripline circuits cannot be realized in microstrip?

Lay out a simple circuit. Go to the *Plot* window and save the circuit using *Ctrl*-`s`. Exit *Puff*; then use an ASCII editor to open the saved *Puff* file. Go to the section of the file that begins `\c{ircuit}`. *Puff* has saved your keystrokes in the *Layout* window in what we call a *keylist*. When reading a new file, *Puff* uses the keylist to redraw the circuit. What keystrokes are not saved? Is it better to erase all the parts using *Ctrl*-`e` or with repeated shift-arrow operations? Can you think of some advantages in keeping the keylist as short as possible?

The most common *Puff* layout errors involve invisible parts. Starting from a blank layout, select a `tline` part and make repeated shift-arrow operations to move the × cursor about the *Layout* window. Go to the *Parts* window, increase the length of the `tline` used, and then return to the *Layout* window. See how long you can make the `tline` before causing a layout error. What happens if you try to delete the `tline` from the parts list?

## C.7 BOARD WINDOW

The relative dimensions of the circuit board in the *Layout* window are specified from the *Board* window. These dimensions set the scale used to draw the distributed components on the screen. Table C.4 gives a brief description of each of the parameters available. Access the *Board* window by pressing function key *F4*. Edit the parameters using the same keys as in the *Parts* window. *F10* brings up a help window explaining the board parameters.

The normalizing impedance `zd` is used to calculate the scattering parameters. It also defines the normalized impedance and admittance values (`z` and `y`) that may appear in the *Parts* window. Paths to connectors are drawn with transmission lines with impedance `zd`. The design frequency `fd` is used to determine the physical lengths of distributed components entered in degrees. The Hz units for `fd` may use any prefix that appears in the Chapter 2 table, although MHz and GHz are generally the most practical. The *Plot* window will inherit the frequency prefix. Change the prefix when you find yourself entering lots of zeros below the log-magnitude plot. Use caution. The prefixes are case sensitive. Also beware of frequency data in `device` and `indef` files that do not coincide with the `fd` prefix (e.g., *Puff* will give meaningless results if you use a device file with GHz frequency data when MHz is used for `fd`). Device scattering parameter data must also match the *Board* window's definition of `zd`.

The *Tab* key toggles between microstrip, Manhattan, and stripline layouts. Resort to the Manhattan mode when `tlines` or `clines` become too long or short. All distributed parts are then drawn with widths 1/20 the board size, and lengths 1/10 the board size. This mode also permits components with unrealizable values, such as `tlines` with negative electrical lengths. However, if *Puff* requires an

**Table C.4.** Description of Parameters that May Be Modified from the *Board* Window.

| Parameter | Description |
|---|---|
| zd | Normalizing or characteristic impedance. Used in the calculation of scattering parameters. Units are Ωs with optional prefix. |
| fd | Design frequency. Used to compute electrical length of parts entered in degrees. Also the frequency used for the component sweep. Prefix given is carried over into the *Plot* window. |
| er | Relative dielectric constant of substrate. Used to calculate dimensions for microstrip and stripline components. Unitless. |
| h | Substrate thickness. One of three parameters that specify the equivalent dimensions for the *Layout* window. Significant in transmission line calculations. |
| s | Board size. Specifies the equivalent length of each side of the square circuit board that makes up the *Layout* window. |
| c | Connector separation. Sets the spacing between ports 1 and 3 and between ports 2 and 4 in the layout. Set to zero to create a centered two-port. |
| *Tab* | Circuit type. Use the *Tab* key to select a microstrip, stripline, or Manhattan layout. |

electrical length calculation, stripline models will be used, and physically unrealizable parameters may not be allowed. If enabled, Manhattan dimensions will appear in the artwork, and artwork corrections will be ignored.

## C.8 PLOT WINDOW

To reach the *Plot* window, push function key *F2*. The circuit in the *Layout* window is analyzed by typing p. If you type *Ctrl*-p, the previous plot will be drawn before the new plot, allowing a comparison of results. After an analysis has been completed, the *Plot* window lists the values of the scattering coefficients at design frequency fd. Use the *PgUp* and *PgDn* keys to move the markers and show the scattering coefficients at the other frequencies. The ↑ and ↓ keys can be used to move the cursor to various parameters. They cycle through a loop that includes the *Plot* window and the *x* and *y* axes on the rectangular plot. You can type over any parameter to change it. *Puff* can plot up to four different scattering parameters simultaneously. To select an *s*-parameter, move the cursor down toward the bottom of the *Plot* window, and a marker will appear, together with the letter s. Then type in the port numbers for the desired *s*-parameters. If you leave a line blank, it will be erased when you move the cursor. Those with EGA and VGA graphics can use the *Tab* key to change the Smith chart from an impedance chart to an admittance chart. Typing *Alt*-s with VGA graphics toggles an enlarged Smith chart.

The *Plot* window allows you to select the number of frequency `Points` to analyze. This must be a positive integer no greater than 500, assuming you have a full complement of memory. *Puff* interpolates between calculated points with a cubic spline. The interpolation is performed by splining the real and imaginary parts of the scattering coefficients separately. The independent variable for calculating the spline curve is the spacing on the Smith chart. This gives better results than using frequency as the splining parameter. If a curve kinks on the Smith chart, or gives erroneous ripples on the rectangular plot, it is an indication that the number of `Points` is too small.

You can plot an impulse response by typing `i`, or a step response by typing `s`. *Puff* will request a frequency interval specified by the ratio `fd/df`. It will then do a 256-point inverse fast Fourier transform of the scattering coefficients, and plot the results on a linear scale. The amplitude of the time-domain plot is the same as the Smith chart radius. The ratio `fd/df` will determine the time axis for the plot, which goes from −1/8`df` to 3/8`df`. The upper and lower limits on the frequency axis of the magnitude plot are used for windowing the Fourier transform. The window is a raised cosine that goes to zero at the upper frequency limit. The scattering coefficients are set to zero outside the window.

A convenient way to see the impulse or step inputs is to draw an open-circuited connector path at an unused port. The reflection coefficient for this port is the input waveform. *Puff* normalizes the input waveforms so that the peak value is 1. The high-frequency limit controls the rise and fall times, and the low-frequency limit affects the ringing. Be aware that the time-domain waveforms are actually periodic, with period 1/`df`, and aliasing from the previous pulse may affect the response. The step input is actually a square wave, and the response to the previous falling edge will affect the rising step that follows.

To save a network in a circuit file, type *Ctrl*-`s` and give a file name. The *Parts* window and the data in the *Plot* window will be saved along with the circuit. The `.puf` extension will be added to the file name if one is not specified. Typing *Ctrl*-`a` from the *Plot* window will activate *Puff*'s photographic artwork routines. The layout produced will be magnified by the photographic reduction ratio (`p`) in the circuit file. Only the `tlines`, `qlines`, and `clines` will appear in the artwork, and the corners will be mitered. The artwork output parameter (`o`) in the circuit file allows you to specify dot-matrix or HP LaserJet printouts or the production of a Hewlett-Packard Graphics Language (HP-GL) file. *Puff* will prompt for titles to be placed atop the printout, or for an HP-GL file name. The HP-GL file will be created with the `.hpg` extension. The *Puff* diskette includes the program `hpg2com`. Use this to dump `.hpg` files to a serial plotter connected at port COM1.

## C.9 COMPONENT SWEEP

The simple optimizer included in *Puff* is called the *component sweep*. Instead of sweeping with respect to frequency, a circuit's scattering coefficients may be swept with respect to a changing component parameter. This feature is invoked by placing a question mark (`?`) in front of the parameter to be swept in the appropriate position of a part's description. For example, to find the optimum value for a tuning capacitor, one could specify a part as `lumped ?5pF`. When plotting in the *Plot* window, the frequency will then be held constant at the design frequency `fd`, and the values specified in the x axis of the rectangular plot will be substituted for (in the above example) capacitance values in picofarads. In this manner, any parameter used in the parts list may be designated as a sweep parameter, but only one parameter may be swept at a time. Swept lumped elements are restricted to single resistors, capacitors, or inductors. A description such as `lumped ?1+5j−5jΩ` is not allowed since it is a series RLC circuit. In addition, the parallel sign ‖ may not be present in the lumped specification. The unit and prefix given in the part's description (following the `?`) is inherited by the component sweep.