Fast algorithm computes histograms
Lukasz Sliwczynski, University of Mining and Metallurgy, Cracow, Poland

A histogram is a convenient method of presenting statistical features of data; you can use a histogram for an estimate of the probability density function of a random phenomenon. To prepare a histogram entails dividing the interval occupied by data values into some number of smaller intervals and then counting the number of occurrences of data in each subinterval. You can easily generate histograms with the aid of standard mathematical packages, such as Matlab. The problem lies in the fact that calculating a histogram using the standard Matlab function `hist()` from a large file of data (for example, 10 million data points) with high accuracy (for example, with 12-bit resolution) may take substantial time—many hours, or even days, depending on computer speed and memory. The problem is that `hist()` treats its input data as “analog” data with almost infinite precision, and the routine must use program loops, which are very slow in Matlab.

However, you often perform measurements with the use of A/D conversion, with the result that the input data for the histogram-calculating procedure are discrete and that the smallest difference between any two data values is one. This fact allows you to write a special function for calculating the histogram. You can use the complete Matlab program for calculating histograms from discrete data (Listing 1). The program is intended for placement into a Matlab m-file, which creates a function from the routine. This function is accessible from the Matlab command window, just as any other function. Figure 1 illus-

```
Listing 1—Matlab Program for Fast Histogram Calculation

function N=fasthist(g,L,W)

N=fasthist(g,L,W)  %

Function calculates histogram of data from D/A converter.
% Range of numbers is from 0 to 2^L-1. Subinterval length is W.
% L is the number of bits of the converter.

g=g(:,1);
gx=sort(g);
gsd=diff(g);
index=find(gsd);
V=gg[index].gg[width(gg)]+1;
N=sum(2^L.V)

if W==L
    for i=1:gg(2^L/W)
        N(i)=sum(N([W*(i-1)+1:L]));
    end

N=N(1:gg(2^L/W));
```

Figure 1 Ordering of the data points in diagrams a through f follows the steps in Listing 1.
trates the operation of the algorithm. The first line (1) of the program prepares input data and gives the output the form of the decision vector g (Figure 4a). The second line (2) performs data sorting and formats the data in ascending order (as in Figure 4b). The decision vector g in Figure 4c is only those points where data changes; elsewhere, the vector has a value of zero. The distances between two consecutive peaks is equal to the number of data samples of the same values, and the height of a peak is equal to the distance between two consecutive numbers.

Next, the routine (4) calculates the indexes of data changes (Figure 4d) and calculates the values of data V (Figure 4e) appearing in input vector g (5). A "1" in this code line is necessary because a vector index cannot equal zero in Matlab. In the following step (7), the step vector V indexes the output vector from the procedure N. This indexing operation is simple and fast because both indexes and data are natural numbers and belong to the same interval [0,2^−1]. At this point, the essential part of the procedure is finished. The numbers N(i) (Figure 4f) are equal to the total number of occurrences of data with the value 1−1 in the input vector g. The last six lines of the program reduce the initial resolution of the A/D converter that you use to acquire data. To obtain faster execution of the program, you switch off this part of the routine only when necessary—for example, when W does not equal 1.

The effect of using this algorithm rather than the standard one in the Matlab package is significant. Using the standard function hist[], the time to compute a histogram from 10,000 points is 10 sec; for the same data file with 12-bit resolution, the algorithm fasthist() requires only 0.35 sec. These figures are based on using a 100-MHz Pentium machine with 33 Mbytes of RAM, running under Windows 95. Using fasthist(), a histogram from 10 million-point data file takes approximately a half-hour as opposed to more than a day using hist(). These results are for Matlab version 4.2. Version 5.2 gives somewhat different results: 68 sec for hist() and 0.38 sec for fasthist(), a 179-times reduction. (DI #2422)

To Vote For This Design, Circle No. 358

Loop powers current transmitter

By N Kannan, Mediatronix Ltd, Triandrum, India

In the circuit in Figure 1, the loop supply is 20 to 30V dc, and the loop current is 10 to 20 mA. IC, operates as a constant-current (Ic) source. IC, R1, and R1 operate as a shunt regulator, which provides 12V VCC. IC, is a TL032 op amp. The sensor circuit could be a resistance-temperature-detector circuit or any other sensor- or signal-conditioning circuit. The IC, combination operates as a current sink, with Ic = V1 / R1. The total loop current is Iloop + Ic. Because of its common-mode input-range limitation, Ic, requires a minimum 2V input for proper operation. Hence, the V1 signal range must be 2 to 10V. With V1 = 2V, you adjust potentiometer P1 to set Iloop at 10 mA. At this operating point, Ic = 4 mA, and the balance of the current (Ic = 6 mA) goes to the shunt regulator, the op amp, and the sensor-circuit load. The maximum load current is typically 5 mA as V1 increases from 2 to 10V, Iloop increases proportionally, solely from the increase in Ic, because Ic is constant.

Inexpensive components provide an efficient loop-powered current transmitter.

<table>
<thead>
<tr>
<th>V1</th>
<th>Vc</th>
</tr>
</thead>
<tbody>
<tr>
<td>2V</td>
<td>2V</td>
</tr>
<tr>
<td>3V</td>
<td>2.4V</td>
</tr>
<tr>
<td>4V</td>
<td>2.8V</td>
</tr>
<tr>
<td>6V</td>
<td>3.6V</td>
</tr>
<tr>
<td>8V</td>
<td>4.4V</td>
</tr>
<tr>
<td>10V</td>
<td>5.2V</td>
</tr>
</tbody>
</table>

For every 1V increase in V1, the change in Vc is 0.4V. The circuit has good accuracy and temperature stability and operates with loop supplies of 20 to 30V. The maximum available sensor-circuit current is approximately 5 mA. Currents greater than this value result in poor Vc regulation. (DI #2421).

To Vote For This Design, Circle No. 359

152 EDN | October 28, 1999

www.ednmag.com
Spice creates time-variant resistor
Vittorio Ricchiuti, Italtel, L’Aquila, Italy

Figure 1 shows a subcircuit that represents the behavioral model of a linearly time-variant resistor in PSpice (OrCAD, www.orscad.com). PSpice libraries include no models of such resistors, which you can use in various applications—for example, to simulate the fallback characteristic of a linear regulator in transient analysis. The suggested model uses the E-device in the PSpice analog-behavioral Modeling parts library. Listing 1 describes the subcircuit in Figure 1 for a time-variant resistor with the characteristic in Figure 2. The function in Figure 2 is the product of an independent voltage source, \( V_{\text{CONTROL}} \). The independent zero-voltage source \( V_{\text{SENSE}} \) senses the current \( I(V_{\text{SENSE}}) \) through the resistor. The voltage \( V_2 \) between nodes 1 and 2 of the time-variant resistor is \( V_2 = f(t) \times I(V_{\text{SENSE}}) \), where \( f(t) \) is a function of \( V_{\text{CONTROL}} \) (DI #2424).

To Vote For This Design, Circle No. 360

Listing 1—Macro Model of Time-Variant Resistor

```plaintext
.SUBckt VAR_RES 12
R1 10
V_Vsense 4 0
+PM5 2 1 \( \frac{2}{1} \) 0 0 0
R_RES_LIM 3 2 VALUE \( \frac{I(V_{\text{Vsense}}) \times (4, 0)}{1} \)
V_Vsense 1 3 DC 0V AC 0V 0V
.END VAR_RES
```

The routine in Listing 1 simulates a linearly time-variant resistor.

VCO supply touts low noise
Ted Henderson, Linear Technology Corp, Milpitas, CA

Many portable RF products use VCOs to generate the RF carrier frequency. These applications often require low-noise VCO power-supply voltages that are higher than the primary battery voltage. Many designs use a dc/dc converter powering a low-noise linear regulator. This approach has several inherent disadvantages. The dc/dc converter produces noise that the regulator may not reject, resulting in regulator-output noise far greater than the thermal-noise.
levels. The linear regulator might require a large output-compensation capacitor with stringent ESR requirements. Finally, the board area for both devices and support components can be large. The circuit in Figure 1 operates from an input range of 2.5 to 4.4V and generates a 4.2V low-noise voltage for a 900-MHz VCO.

The circuit is based on IC1, an LTC1682 charge-pump dc/dc converter. The device includes a charge-pump/linear-regulator tandem that's optimized for minimum regulator-output noise. The linear regulator operates with several types of output capacitors, including small, low-value, low-ESR ceramic capacitors. Figure 2 shows the close-in phase noise of the VCO operating in open-loop mode; Figure 3 shows the peak-to-peak noise voltage at the regulator's output. (DI #2430).

The VCO in Figure 1 exhibits low close-in phase noise.

The supply in Figure 1 generates only approximately 200 mV of peak-to-peak noise.
μC forms FM oscillator

Abel Raynus, Armatron International, Melrose, MA

A project required an inexpensive oscillator whose frequency increased step by step from 200 to 400 Hz and then decreased to 200 Hz. The first step was to design a VCO with a staircase driver. However, this approach entailed at least four ICs and many discrete components. An alternative method (Figure 1) requires only one 16-pin μC (an MC68HC705K1J, costing less than $1) and only a few external components. The process takes place exclusively in software (Listing 1). The μC generates a burst of 256 cycles of a given period. After that, the period decrements or increments, and the μC generates a burst with the new period. The choice of decrement or increment depends on the contents of the flag register. If the flag is zero, it is a sign to decrement; otherwise, to increment. When the set of frequencies reaches completion, the content of the register inverts. So, in one operation, the frequencies grow; in the next, they decrease.

Keep in mind that a linear change in the value of the period causes a nonlinear change in the frequency value (according to a hyperbolic curve). So, you should perform all calculations in the time domain, rather than in the frequency domain. In our application, the non-linear aspect of the frequency is unimportant. However, if necessary, you can realize any kind of FM by organizing the table of frequencies corresponding to each step. The design determines the period resolution, ∆T. In this case, it equals 0.02 msec, because this value allows you to use only one 8-bit register (HPER) to perform all period-changing operations.

For more precision, you could use two or more registers. The number of frequencies in the set is $K = \frac{T_{\text{MAX}}}{\Delta T}$.

![Figure 1](image)

An inexpensive μC replaces complicated VCO circuitry to configure a simple frequency-step generator.

**LISTING 1—μC ASSEMBLY CODE FOR FREQUENCY GENERATOR**

```
;1  * Query FM Oscillator
;2  * range 200 Hz - 400 Hz,
;3  * period discontinuity - 0.02 ms
;4  * number of frequencies - 128
;5  * literal 0
;6  * I/O PORTS
;7  * start
;8  0000 :0
;9  0000 :10 OUT equa 7 pta pina
;10  0000 :11 CONSTANTS
;11  0000 :12 HMAX equa 25077 Hz; frequency = 200 Hz
;12  0000 :13 *VARIABLES
;13  0000 :14 org RN
;14  0000 :15 clk mbs 1 sound duration counter
;15  0000 :16 clk mbs 1 sound frequency counter
;16  0000 :17 HPER mbs 1 half of period register
;17  0000 :18 flag mbs 1 flag register
;18  0000 :19 *INITIALIZATION
;19  07F1 ;20 org RN
;20  07F1 20 fcb 100010000 00c.parall.resistor
;21  0300 22 org RN
;22  0300 A6FF ;23 init lda 0
;23  0302 8704 ;24 sta dds a 60 ms per output
;24  0304 3FC0 ;25 clrs R1 / 0 -> R1
;25  0306 3FC1 ;26 clrs R2 / 0 -> R2
;26  0308 3FC3 ;27 clrs flag
;27  030A 6FPA ;28 lda HMAX
;28  030C 87C2 ;29 sta HPRF ; set T = 5 ms, f = 200 kHz
;29  030E C032D ;30 main jsr burst ; generate 256 cycles
;30  0311 3CC3 ;31 main flag
;31  0313 2714 ;32 main bex mL
;32  0315 3CC2 ;33 main inc HPER ; HPRF = 1
;33  0317 3CC1 ;34 main inc R1 / R2 + 1
;34  0319 86C1 ;35 main lda R2 / R2 -> ACC
;35  031B A170 ;36 main cmp 12575 ; R2 < 125 kHz
;36  031D 2FEP ;37 main bra main
;37  031F 86C3 ;38 main lda flag ; inverting flag
;38  0321 A011 ;39 main bor #1 /
;39  0323 87C3 ;40 main sta flag
;40  0325 3FC1 ;41 main clrs R2 / 0 -> R2
;41  0327 20DE ;42 main bca main
;42  0329 3CC2 ;43 main mL dec HPER ; HPRF = 1
;43  032B 80EA ;44 main bra mL
;44  032D 1ED0 ;45 main burst bext out.pra ; P0 -> out
;45  032F 86C2 ;46 main lda HPER ; HPER = x
;46  0331 C0342 ;47 main jsr dly01x ; half period delay
;47  0334 LF00 ;48 main bcl out.pra ; P0 -> out
;48  0336 86C5 ;49 main lda HPER ; HPER = x
;49  0338 C0342 ;50 main jsr dly01x ; half period delay
;50  033B 3CC0 ;51 main inc R1 / R1 + 1
;51  033D 3CC0 ;52 main lda R2 / R2 -> ACC
;52  033F 26EC ;53 main bra dly01x ; return from burst
;53  0341 A402 ;54 main dly01x lda #2 ; delay 0.01 x ms
;54  0342 4A02 ;55 main rep0 deba
;55  0344 24FD ;56 main bca rep0
;56  0347 5A ;57 main dca
;57  0348 207F ;58 main dly01x ; return from dly01x
;58  034A A1 ;59 main sta
;59  034C 3CC2 ;60 main org VECTORS + 6
;60  034E 0300 ;61 main lbdn init
```

www.ednmag.com
T_{\text{MIN}})/\Delta T$. You can determine any frequency is the set as:

Flag = 0: \( f_c = 1/(T_{\text{MAX}} - \Delta T \cdot K2) \).
Flag = 1: \( f_c = 1/(T_{\text{MIN}} + \Delta T \cdot K2) \).

Table 1 shows some of the frequencies. The duration of the set is \( t = \frac{1}{f_c}(T_{\text{MAX}} + T_{\text{MIN}} + K1_{\text{MAX}} \cdot K2_{\text{MAX}}) \). In this case, \( t = 2 \) msec (DI #2432).

<table>
<thead>
<tr>
<th>Flag</th>
<th>K2</th>
<th>T (msec)</th>
<th>( \frac{1}{T_c} ) (msec)</th>
<th>HPER</th>
<th>F (Hz)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>5.56</td>
<td>2.50</td>
<td>250</td>
<td>200.0</td>
<td>( T_{\text{MAX}} )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>4.88</td>
<td>2.49</td>
<td>249</td>
<td>200.8</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>4.76</td>
<td>2.48</td>
<td>248</td>
<td>201.6</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>4.64</td>
<td>2.47</td>
<td>247</td>
<td>202.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
<td></td>
<td>298.7</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>123</td>
<td>2.54</td>
<td>1.27</td>
<td>127</td>
<td>391.7</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>124</td>
<td>2.52</td>
<td>1.26</td>
<td>126</td>
<td>396.8</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>2.50</td>
<td>1.25</td>
<td>125</td>
<td>400.0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2.52</td>
<td>1.26</td>
<td>126</td>
<td>396.8</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2.54</td>
<td>1.27</td>
<td>127</td>
<td>393.7</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2.50</td>
<td>1.28</td>
<td>128</td>
<td>396.5</td>
<td></td>
</tr>
</tbody>
</table>

To Vote For This Design, Circle No. 362

Circuit converts pulse width to voltage

Ron Hegler, Picker International, Cleveland, OH

The circuit in Figure 1 stems from a radio-controlled modeling application, which requires a voltage proportional to the width of the incoming servo pulses. The circuit is optimized for a positive-going pulse width of 1 to 2 msec, repeating at intervals of approximately 17 msec. The output produces a voltage of 0.95V for a 1-msec pulse to 2.25V for a 2-msec pulse. The circuit operates similarly to a PLL, but it locks onto the pulse width, rather than to the frequency, of the incoming signal. IC2a is a one-shot multivibrator with its time constant a function of \( R_2 \), the FET's on-resistance, and \( C_1 \). IC2a is a pulse-width comparator that compares the reference pulse from IC1b with the incoming pulse. Upon the rise of the incoming pulse, IC3a's Q output clocks high and drives IC2a's D input high. IC3a times out before the input pulse falls, IC3a's Q output goes low, driving the D input of IC2a low. This action drives the Q of IC1b high when the input pulse falls. The Q output connects to the FET through \( D_1 \) and \( R_1 \). \( C_2 \) filters the Q output to adjust the time constant of one-shot IC3a to match the incoming pulse. The voltage across \( C_3 \) indicates the incoming pulse width. (DI #2431).

To Vote For This Design, Circle No. 363
Restore dc to NRZ signals

Jay Kirschbaum, Columbia University, New York, NY

For NRZ signals, ac amplification is preferable to dc amplification, because ac amplification is usually more economical and has greater immunity to drift. However, ac processing has the disadvantage that ac coupling removes the dc reference level of the digital data stream. As Figure 1 shows, if a long stream of ones or zeros appears at the amplifier's input, the output droops after a period determined by the RC time constant at the input. This problem makes it necessary to include a dc-restoration circuit following the ac amplifier.

The circuit in Figure 2 restores the dc level that the amplifier removes and eliminates droop, even if long strings of ones or zeros appear in the data stream. The circuit also provides additional amplification, increasing the signal to ECL levels. Synchronous clocking gives the circuit good immunity to noise in the data channel. Tests reveal that the circuit op-

---

This circuit eliminates the droop from the data stream in Figure 1 and simultaneously amplifies the digital signal to ECL levels.
erates satisfactorily at clock speeds from dc to 1 GHz.

IC is an MC10F416 five-channel line receiver, which boosts digital signals from a minimum of 50 mV p-p to ECL levels. The input circuit of each channel of IC is a differential amplifier, which responds when the signal level at the true input swings past the signal at the complementary input. In Figure 2, the input data stream goes to the D input. The complementary input D connects to the Vpp pin, which is at -1.3V. The true input D connects to bias voltage Vd through 50Ω. Vd is also nominally -1.3V, but you can tweak it to compensate for device characteristics. IC is an MC10EL31 flip-flop. Both ICs are specified to 2 GHz. The logic level at D transfers to Q at the positive transition of the clock signal at CLE. Resistor R, and the 50Ω pull-down resistor, R, form a voltage divider that puts D above or below √D by half the expected input-voltage swing. The value of R is a function of the swing of the input signal. For example, assume D has the value 100 mV p-p. If R is 370Ω, and Q is at logic zero (-1.75V), the voltage divider sets D above 50 mV below D and Q stays at logic zero as long as these conditions prevail.

Now suppose that logic one appears at D. This condition simultaneously puts D above 50 mV above D, and Q switches to logic one (-0.9V) at the next clock transition. When this condition happens, the R,-R, voltage divider holds D five 50 mV above D, and Q stays at logic one until the next transition to logic zero. Voltage droop disappears, and the dc reference of the digital signal is re-established itself. Moreover, the circuit amplifies the 100-mV swing of the digital signal to ECL levels. Because the circuit triggers only when the clock edge appears, it's highly immune to noise in the data channel. At power-up, one transition from logic zero to logic one and one transition from logic one to logic zero are necessary at the amplifier input to put the circuit in a defined state. After that sequence, the circuit faithfully follows the input.

Proper timing between the data and the clock is important to reach the maximum operating speed of the circuit. Ideally, the clock edge should arrive at IC, as soon as the data-setup time is over, switching Q and pulling up D as close to the data transition as possible. The theoretical upper limit on the operating (clock) frequency is a function of the propagation delay through the circuit (825 psec at 25°C). Data transitions must be spaced at least this far apart, so the maximum clock frequency is 1.2 GHz. Tests show that the circuit operates successfully at clock frequencies of dc to 1 GHz and data frequencies of dc to 500 MHz. The tests used a square wave for the clock input and a synchronous square wave of a lower frequency at the data input. The circuit restores the dc level and amplifies the signal to ECL levels over the entire test range. (DI #2434).

To Vote For This Design,
Circle No. 364

---

**TelCom Offers Highly Integrated, Very-High-Voltage Off-Line Switching Regulators that Simplify Your Power Converter Designs**

**Power Switch**

<table>
<thead>
<tr>
<th>Device</th>
<th>On Resistance (Ω)</th>
<th>Peak Current (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC33370</td>
<td>12.8</td>
<td>0.9</td>
</tr>
<tr>
<td>TC3371</td>
<td>6.8</td>
<td>1.5</td>
</tr>
<tr>
<td>TC3372</td>
<td>4.8</td>
<td>2.0</td>
</tr>
<tr>
<td>TC3373</td>
<td>3.8</td>
<td>2.7</td>
</tr>
<tr>
<td>TC33574</td>
<td>3.0</td>
<td>3.9</td>
</tr>
</tbody>
</table>

**TC3370 Series Features**

- Simplified Interface allows On/Off Control by a Microcontroller or other Circuitry
- Lower System Cost with Reduced Component Count for a Total Converter System Solution
- Improved Reliability with Internal Thermal Shutdown Protection, Cycle-by-Cycle Current Limiting, and Input Undervoltage Lockout with Hysteresis
- Programmable State Controller

**TC3370 Series Applications**

- Consumer Electronics
- Office Automation
- Personal Computers
- Wireless Communications
- Industrial Systems

**Order Your 1999 Data Book Now!**

**Literature: 1-800-888-5806 - www.telcom-semi.com**

Order Your EV Kit Today!