Data Synchronization Simulation Using The MATHWORKS Communications Toolbox.

Darrel R. Judd
Harman-Motive Inc.

Abstract

This paper demonstrates the use of the Communications Toolbox[i] to simulate and test a data synchronizer using an early-late gate technique. The simulation is done in SIMULINK. A two tone FSK signal is generated, passed through an AWGN channel, down converted to baseband and passed to an FM detector. The signal is then synchronized so that the binary information can be extracted from the signal. The utility of the Communications Toolbox to simulate this task is then evaluated.

Introduction

When a communications signal is received, it must be synchronized to the system that receives it. Depending on the type of demodulation scheme used, it may be important to know the correct carrier frequency and phase. This is called carrier synchronization. If the signal is a data signal, it is also important to sync up to the symbol periods of the data. This is called data synchronization. There are many ways that the synchronization problem has been solved: pilot tones, preambles, phase locked loops, etc. This paper assumes that the signal of interest is a 2 tone FSK signal. It is also assumed that the baud rate is known, but the starting point of each symbol is not known. An early-late gate technique[ii] is used to baud sync to the instantaneous frequency output of an FM detection process.

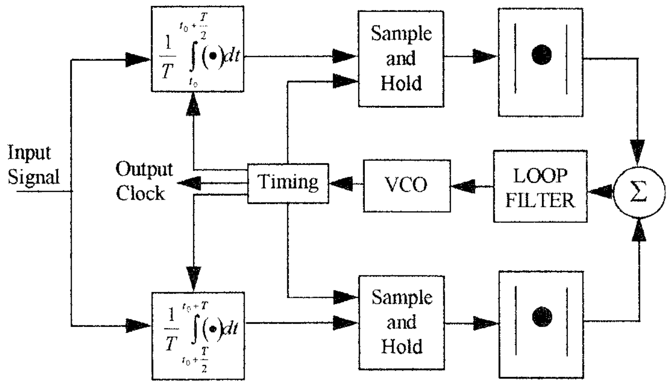

Input Signal
$\frac{1}{T}\int_{t_0}^{t_0+\frac{T}{2}}(\bullet)dt$
$\frac{1}{T}\int_{t_0+\frac{T}{2}}^{t_0+T}(\bullet)dt$
Sample and Hold
Sample and Hold
Output Clock
Timing
VCO
LOOP FILTER
Σ

Figure 1: Block diagram of Early-Late Gate data synchronizer. This version works well with rectangular pulses.

Early Late Gate Theory

The early-late gate synchronizer is a very popular, but older technique that has been around for a long time. A block diagram of the algorithm is shown in Figure 1. The basic principle for this data synching technique is to compare the energy in the first half of a symbol period to the energy in the last half of a symbol period. Thus "Early" and "Late" gate, respectively.

As the energy levels in these gates are compared to each other, there are four conditions that can happen:

1) The loop has locked to the symbol period. The energy levels are equal. The symbol transition ocurs between the early and late gate. No timing correction is made for this condition.
2) Two identical symbols have been transmitted. The energy levels are equal. This means that a symbol transition was not found in this symbol period. No timing correction is made for this condition.
3) The energy in the early gate is greater than the energy in the late gate. This means that there was a symbol transition in the last half of a symbol period. A timing correction can now be made to sync up to the symbol boundary.
4) The energy in the late gate is greater than the energy in the early gate. This means that there was a symbol transition in the early gate. A timing correction can now be made to sync up to the symbol boundary.

These four cases are shown graphically in Figures 2 thru 5.

Figure 2: The shaded areas show equal energy in both gates. Gates match symbol transitions, no timing correction is necessary.

The basic steps of the algorithm shown in Figure 1, are listed below:

1) Given a data signal s(t), choose an arbitrary point in time, t_0. This point lies somewhere within a symbol boundary. Let this point be $s(t_0)$.
2) Let the symbol period be T, where 1/T is the symbol rate.
3) Now define a window in time as $t_0 \le t \le t_0 + \frac{T}{2}$. This window is called the early gate.
4) Now define a window in time as $t_0 + \frac{T}{2} \le t \le t_0 + T$. This window is called the late gate.
5) There are now two adjacent windows that cover one symbol period, but the starting point of the symbol is not known. If

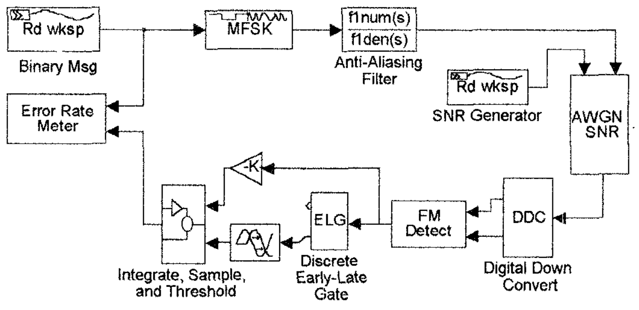
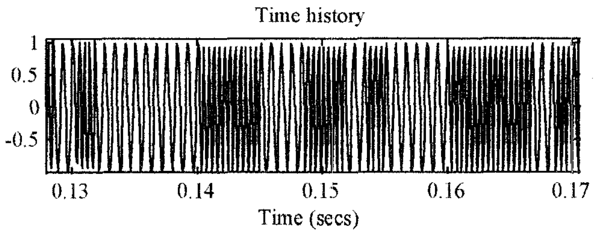


Figure 6:Signal simulation, demodulation and synching with the Early-Late Gate synchronizer.



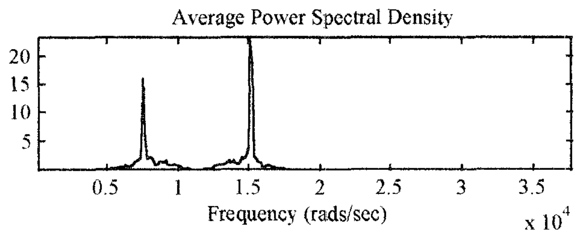


Figure 7: Time domain plot of 2 tone FSK signal, and its power spectral density.

Signal Demodulation

Digital Down Converter

At this point, the signal processing is best described by the following equations:

$$s(t) = s(nT)$$
$$s(nT) = A\cos((\omega_c \pm \omega_s)nT)$$
$$I(nT) = s(nT)\cos(\omega_c nT)$$
$$Q(nT) = -s(nT)\sin(\omega_c nT)$$
$$I(nT) = A/2\cos(\pm\omega_s nT) + A/2\cos((2\omega_c \pm \omega_s)nT)$$
$$Q(nT) = A/2\sin(\pm\omega_s nT) - A/2\sin((2\omega_c \pm \omega_s)t)$$

Once the signal has passed through the communications channel, it must be sampled and down converted to baseband. This is done with the `DDC` block. The analog input is first filtered to insure that it is band limited. Then it is sampled with a `Zero-Order Hold` block.
The overall process for the DDC is shown in Figure 8.

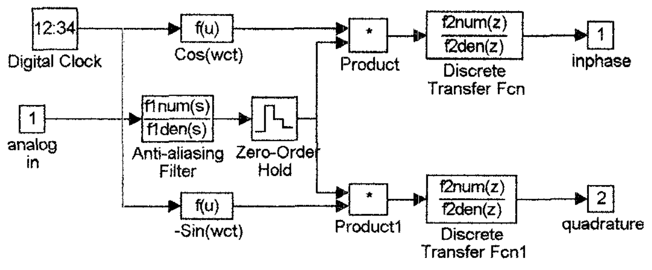
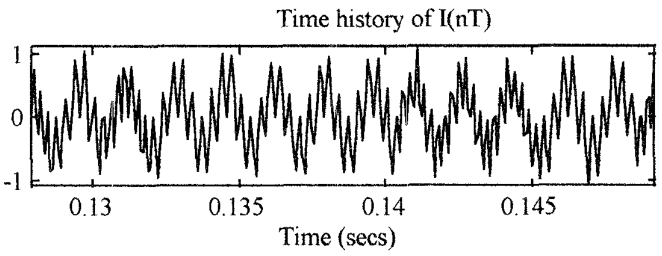


Figure 8: The Digital Down Converter block diagram. The analog input is sampled and frequency down converted to baseband.

I(nT) is shown in Figure 9, note the $2\omega_c \pm \omega_s$ peaks in the power spectral density.



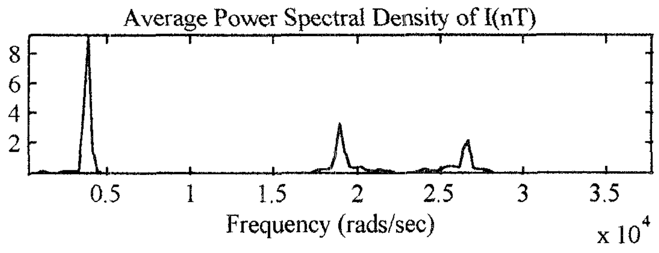


Figure 9: I(nT) and PSD before intermods are filtered out.

The $2\omega_c \pm \omega_s$ terms are filtered out of I and Q with identical low pass filters. The cut off frequency for these filters is determined by the symbol rate used. An example of how this is done is shown in Figure 10.

we look at the amount of energy in each gate, we can determine if t_0 is at a symbol boundary. If the energy in the early gate is greater than the energy in the late gate, this means that the signal transitioned during the late gate. Our point of reference, t_0, could now be decreased and we would be one step closer to knowing the symbol boundary.

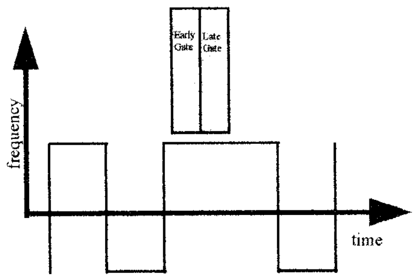


Figure 3: The shaded areas show equal energy in both gates. Gates do not match symbol transitions, but because the energy levels are equal for both gates, no timing correction can be made.

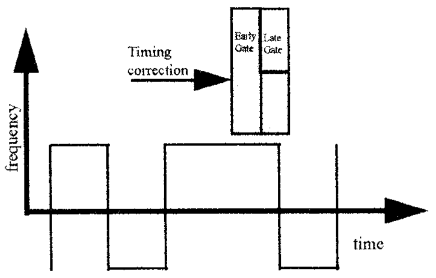


Figure 4: The shaded areas show unequal energy levels. Gates do not match symbol transitions, but the energy levels are not equal for both gates, therefore a timing correction is made.

The first step in the simulation is to generate a binary message with the block `Rd wksp`. Next the binary message is encoded into a 2 tone FSK signal with the block `MFSK`. The FSK signal is filtered and noise is added to the signal with the block, `AWGN SNR`. On the receiving side, the signal is first down converted to remove a carrier and separated into quadrature components with the block `DDC` (Digital Down Converter). The instantaneous frequency of the baseband signal is then calculated using the block `FM Detect`. Now the processed signal is input into the `Discrete Early-Late Gate` block. The outputs of this block are two timing signals. These signals are synchronized to the symbol boundaries of the detected FSK signal. The rest of the blocks are used to display the results of the simulation.

The signal parameters used in the simulation are:

- Sample rate = 12000 samples per second.
- Baud rate = 600 symbols per second.
- The number of FSK tones = 2.
- The carrier frequency = 1200 Hz.
- The frequency spacing between the tones = 1200 Hz.

Signal Simulation

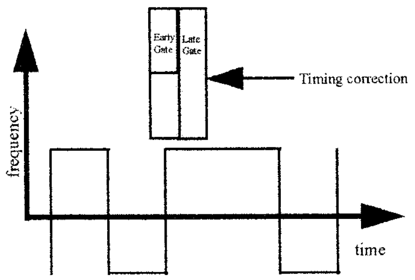


Figure 5: The shaded areas show unequal eneryg levels. Gates do not match symbol transitions, therefore a timing correction is made.

The binary message for the two tone FSK signal is generated with a `Sampled Read From Workspace` block. The parameters for the Sampled Read block are set to generate random numbers ranging between 0 and 1. The output of this block is sent to the `MFSK mod` block. The binary message is also sent to the `Error Rate Meter` block so that the original data can be compared with the demodulated data.

The `MFSK mod` block modulates the digital data stream according to the parameters listed above in the paragraph Early Late Gate Simulation.

Because rectangular pulse shaping is used for this simulation, and because a rectangular pulse is time limited , the resulting signal will have infinite bandwidth in the frequency domain. In order to solve this problem, an anti-aliasing filter is added to limit the bandwidth of the signal. The bandwidth is chosen to be 4000 Hz. The FSK signal and its Power Spectral Density are shown in Figure 7.

Early Late Gate Simulation

The SIMULINK block diagram for the overall simulation is shown in Figure 6.

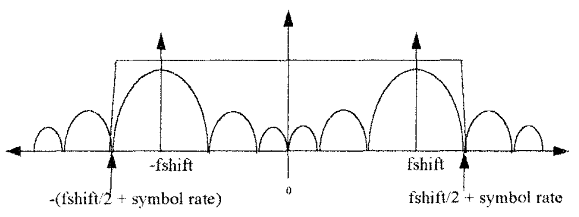


Figure 10: Relationship of low pass filter cutoff frequency to the shift frequency Fshift and the symbol rate. This filter removes the twice carrier frequencies.

The cutoff frequency is chosen such that the main lobes of the Mark and Space tones are included in the pass band. If the frequency shift between the two tones is large enough, a band pass filter could be placed around each main lobe. This would help to eliminate intersymbol interference. Another solution to the intersymbol interference problem is to use a pulse shaping filter. Pulse shaping filters are also included in the Communications Toolbox, but were not used for this simulation. The results of the filtering process are shown in Figure 11.

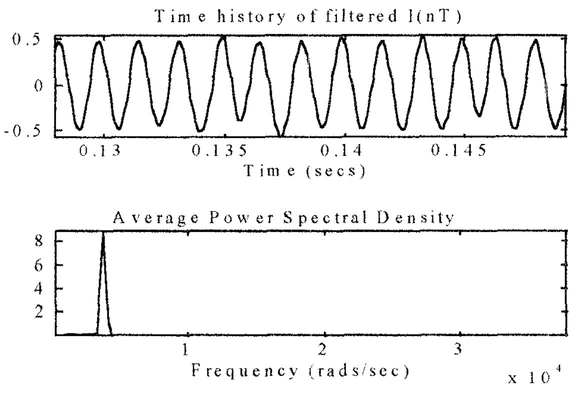


Figure 11: Signal and PSD after intermods are filtered out.

Note that the twice carrier terms have been filtered out. The quality of this filtering will determine the spur free dynamic range of the DDC receiver.

FM Detect

The instantaneous frequency is calculated from the complex signal according to the following equation:

$$freq(nT) = \frac{d\left(a\tan\left(\frac{Q(nT)}{I(nT)}\right)\right)}{dt}.$$

This method works well for high SNR conditions. The `FM Detect` block is shown in Figure 12.

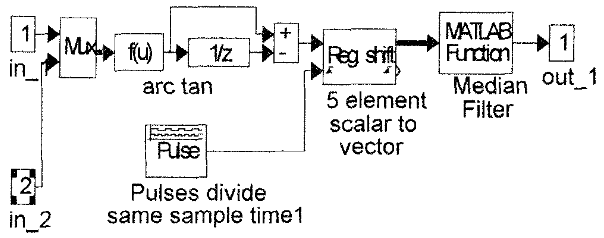


Figure 12: FM Detection followed by median filtering to remove noise spikes caused by atan2 function.

The derivative is simulated with a first order difference consisting of a unit delay and subtract. A problem with this technique is that when the phase of the signal moves from quadrant 3 to quadrant 2 or 4 in the complex plane, large noise spikes are generated in $freq(nT)$. This is shown in Figure 13.

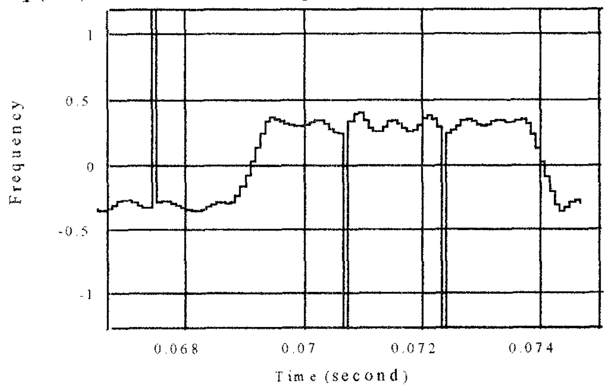


Figure 13: Instantaneous frequency with noise spikes, before median filtering.

These spikes are easily removed with a 5 tap median filter. The median filter is implemented with the `Register Shift` block and the `MATLAB Function` block. The results of the filtering are shown in Figure 14.

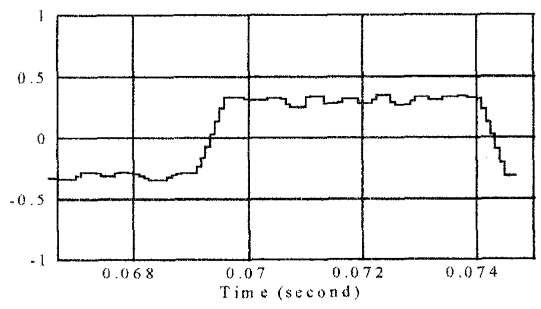


Figure 14: Instantaneous frequency after median filter.

Discrete Early-Late Gate

The diagram for the Early-Late Gate block is shown in Figure 15. For this implementation the loop filter in Figure 1 is unity and therefore is not shown. There are two outputs for the block:

1) A square wave representing the early gate,
2) A square wave for the late gate.

The symbol boundary is at the falling edge of the early gate and at the rising of the late gate.

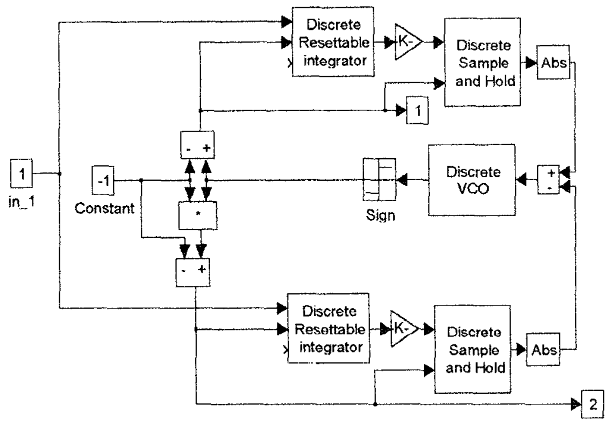


Figure 15:Discrete Early Late Gate Block

This technique for data synchronization can be used for other kinds of demodulation techniques. For example the FM Detection process could be replaced with matched filters. The energy output from each filter could then be used to generate the error function.

Simulation Results

The results of this simulation are graphically displayed in Figure-16.

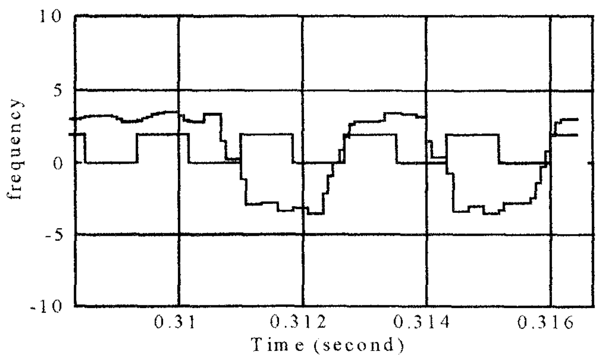
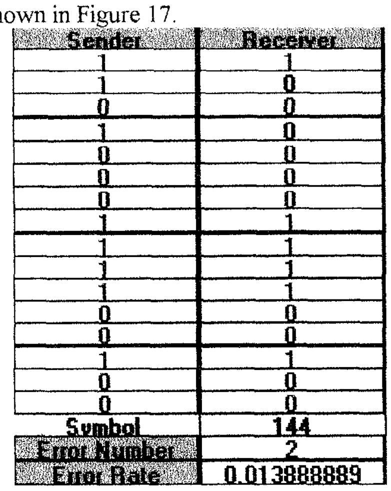


Figure 16: Graph of simulation block shown in Figure 3. The rising edge of the square wave is aligned with the symbol boundaries.

The early and late gate square wave is aligned with the symbol boundaries. The outputs from this block are used to trigger an integrate and dump over each symbol period. The output of the integrator is compared to a threshold to determine each symbol.

The output symbols are compared to the input symbols, and the results are shown in Figure 17.



Sender	Receiver
1	1
1	0
0	0
1	0
0	0
0	0
0	0
1	1
1	1
1	1
1	1
0	0
0	0
1	1
0	0
0	0
Symbol	144
Error Number	2
Error Rate	0.013888889

Figure 17: Error Meter display. After 144 symbols where transferred at an SNR of 5 dB, 2 errors occurred.

Conclusions

The purpose of this paper was not to introduce a new data synchronizing scheme, but to test the utility of The Communications Toolbox. There is a learning curve for SIMULINK and the Communications Toolbox. Once the fundamentals are mastered, the process of creating a simulation block are quite easy. One feature that I really liked was the ability to have both analog and discrete simulations in the same block, for example the DDC block has an analog input and a discrete output. The toolbox worked very well in simulating the early-late gate data synchronizer. The number of SIMULINK blocks and their functionality allows a straight forward approach to most communication problems. This simulation was run with a pre-alpha version of the toolbox. By the time of this publication, some of the block names may have changed. One improvement that could be made to the toolbox is to add a channel block that would allow noise power to specified in terms of SNR instead of noise variance. One was created for this simulation, but is not part of the toolbox. In all, I found the toolbox to be a very useful simulation tool.

---

[i] The Communications Toolbox for MATLAB and SIMULINK. The MATHWORKS Inc. 24 Prime Park Way, Natick, MA 01760

[ii] F.M. Gardner. Phase-locked Loops pages 235-240. John Wiley & Sons, 1979.