



Ingenius. Revista de Ciencia y  
Tecnología

ISSN: 1390-650X

revistaingenius@ups.edu.ec

Universidad Politécnica Salesiana  
Ecuador

Arévalo Bermeo, Germán

Extracción del jitter de una señal binaria de alta velocidad utilizando MatLab

Ingenius. Revista de Ciencia y Tecnología, núm. 1, 2007, pp. 32-38

Universidad Politécnica Salesiana

Cuenca, Ecuador

Disponible en: <http://www.redalyc.org/articulo.oa?id=505554804009>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

# Extracción del jitter de una señal binaria de alta velocidad utilizando MatLab

Germán Arévalo Bermeo  
Ingeniería Electrónica  
UPS-QUITO

Enter



## ¿Qué es el jitter?

Uno de los principales problemas a los que está sometida una señal binaria transmitida a alta velocidad es lo que conocemos como jitter.

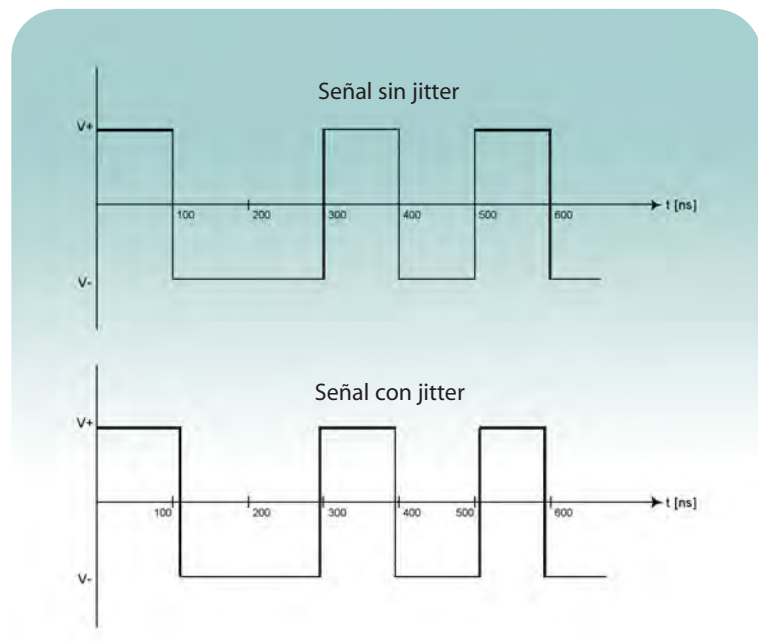
La Unión Internacional de Telecomunicaciones (UIT) en su norma G. 701 define el jitter como *las variaciones de corta dura-*

*ción no cumulativas de una señal digital con respecto a sus posiciones ideales en el tiempo.* El punto de referencia tomado para la observación de dichas variaciones puede ser cualquier punto conveniente y fácilmente identificable, como por ejemplo, el borde izquierdo o derecho de un pulso o cualquier instante de muestreo.

Es decir, el jitter (cuya traducción al español es *temblor*) se genera cuando el cruce por cero de una señal no se produce en el instante exacto en el que debería producirse. Por ejemplo, el tiempo de bit,  $t_b$ , de una señal transmitida a 100 Mbps (ó Mb/s) es de 100 ns (nanosegundos), el mismo que puede calcularse simplemente al evaluar el inverso del ancho de banda,  $AB$ :

$$t_b = \frac{1}{AB} = \frac{1}{10 \text{ [Mb/s]}} = \frac{1}{10 \cdot 10^6 \text{ [bit/s]}} = 100 \cdot 10^{-9} \left[ \frac{s}{\text{bit}} \right] = 100 \left[ \frac{ns}{\text{bit}} \right]$$

de modo que, el cambio de un bit de uno a cero o viceversa debería idealmente producirse exactamente en múltiplos de 100 ns, pero si no es así, los múltiples adelantos o retardos en las transiciones generan el fenómeno conocido como jitter, mismo que puede provocar una degradación en el rendimiento de la transmisión al introducir errores en la lectura de los bits y desfases incontrolados en las señales digitales.



**Uno de los principales problemas a los que está sometida una señal binaria transmitida a alta velocidad es lo que conocemos como jitter.**

El jitter es usualmente medido en *Intervalos de Unidad* (UI-Unit Intervals), donde 1 UI corresponde a un período completo de reloj o, correspondientemente, a un tiempo de bit. La amplitud del jitter puede ser pequeña, de manera que típicamente se emplean las milésimas de UI (mUI) en las mediciones.

El empleo de UI permite independizar las mediciones de jitter de la velocidad de transmisión, pues todo tiempo de bit, independientemente de su valor, equivale a 1 UI. De este modo se puede por ejemplo comparar directamente la generación de jitter de un sistema que opera a 10 Mbps con otro que opera a 1 Gbps.

## Métodos para medición del jitter

Existen principalmente tres métodos para la medición del jitter de una señal digital:

1. Uso de Osciloscopios
2. Empleo de Detectores de fase
3. Procesamiento digital de señales

Los dos primeros métodos emplean dispositivos electrónicos de medición que son generalmente costosos, pero el uso de técnicas de procesamiento digital de señales puede hacerse empleando no sólo hardware especializado, sino incluso software de bajo costo instalado en cualquier computador personal. El presente artículo ilustra una

forma de medición del jitter que emplea el procesamiento de una señal digital empleando el software MatLab.

## Datos empleados para probar el programa

Para comprobar el desempeño del método de extracción del jitter que se describirá a continuación se emplearon los datos muestreados de una trama digital generada por un sistema SONET operante a 10 Gbps (OC-192). Dichos datos están originalmente ordenados en una matriz de dos columnas: una columna de los valores del tiempo (t) y una segunda columna con los correspondientes valores de voltaje (v) como se muestra a continuación.

```
BEGIN TIMEDATA
% t v
      0      0
1.68e-009 -15.2305
1.785e-009  4.23319
1.89e-009  15.6513
1.995e-009 -14.6717
2.1e-009   -5.79758
2.205e-009  3.07659
2.31e-009  11.9508
2.415e-009  2.12189
2.52e-009 -11.861
2.625e-009  16.7937
2.73e-009   3.756
2.835e-009 -14.7537
2.94e-009  -3.78152
.
.
.
2.999976e-005 -18.9297
2.9999865e-005 -1.23846
2.999997e-005  1.9292e-308
END
```

El jitter es usualmente medido en *Intervalos de Unidad* (UI -Unit Intervals), donde 1 UI corresponde a un período completo de reloj o, correspondientemente, a un tiempo de bit.

## Extracción del jitter de una señal binaria empleando MatLab

Para comenzar, el programa debe tomar los datos y buscar los

cruces por cero, esto se realiza mediante el análisis de los signos generados por el producto de cada par de muestras de amplitud de voltaje sucesivas (cada  $i$  muestra por la inmediata-

mente siguiente  $i+1$  muestra) así, si el resultado es negativo se concluye que existe un cruce por cero entre las dos muestras consideradas.

```
load datos.dat; % carga de los datos almacenados en un archivo llamado
                  datos.dat
t=datos(:,1); % separación de los datos de tiempo en una variable llamada t
v=datos(:,2); % separación de los datos de voltaje en una variable llamada v
productos = [v' 0].* [0 v']; % productos de las muestras v(i) por las v(i+1) empleando vectores
productos <=0; % análisis de los signos de los valores generados por los productos
cruces = ans(2:length(ans)-1); % descarte del primer y último valor (el resultado final es un vector
                               % que contiene unos donde hay cruces por cero y ceros donde no hay)
```



El uso de vectores para la realización de los productos consume muchísimo menos ciclos de máquina que el uso de lazos de repetición.

Luego, tomando como referencia el primer cruce por cero, se evalúan las diferencias entre el tiempo en el cual un cruce por cero (una transición binaria) ocurre y

el tiempo óptimo en el cual debía ocurrir, lo cual constituye precisamente el jitter presente en la señal (en el caso de una señal SONET OC-192 se sabe que las transiciones binarias deben ocurrir idealmente en múltiplos exactos de 100 ps).

Es apropiado emplear la ecuación de la recta que une un par de muestras entre las que existe un

cruce por cero para determinar un valor más exacto del punto de cruce de la señal que se está analizando. Además, al tomar como punto de partida el primer cruce por cero se genera un desplazamiento positivo o negativo que debe ser compensado al final, de manera que exista un comportamiento simétrico en los valores positivos y negativos del jitter.

```
min(find(cruces));
```

```
to=t(ans+1)-v(ans+1)/((v(ans)-v(ans+1))/(t(ans)-t(ans+1)));
```

```
[t' 0]-[v' 0]./([(0 v')-[v' 0])./([0 t']-[t' 0])))-t0;
```

```
jitter=ans(2:length(ans)-1).*cruces
```

% encuentra la posición del primer cruce por cero

% encuentra el primer valor del cruce por cero (uso de la ecuación de la recta)

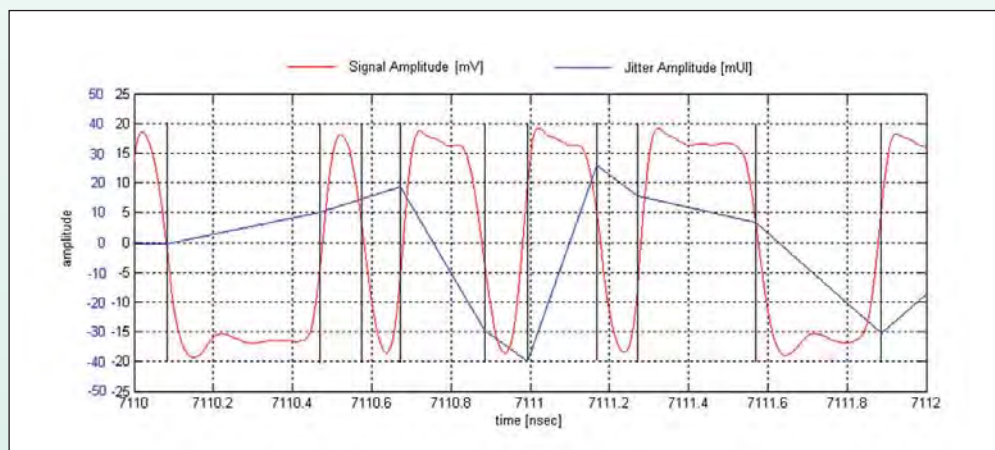
% cálculo del jitter en todas las transiciones binarias

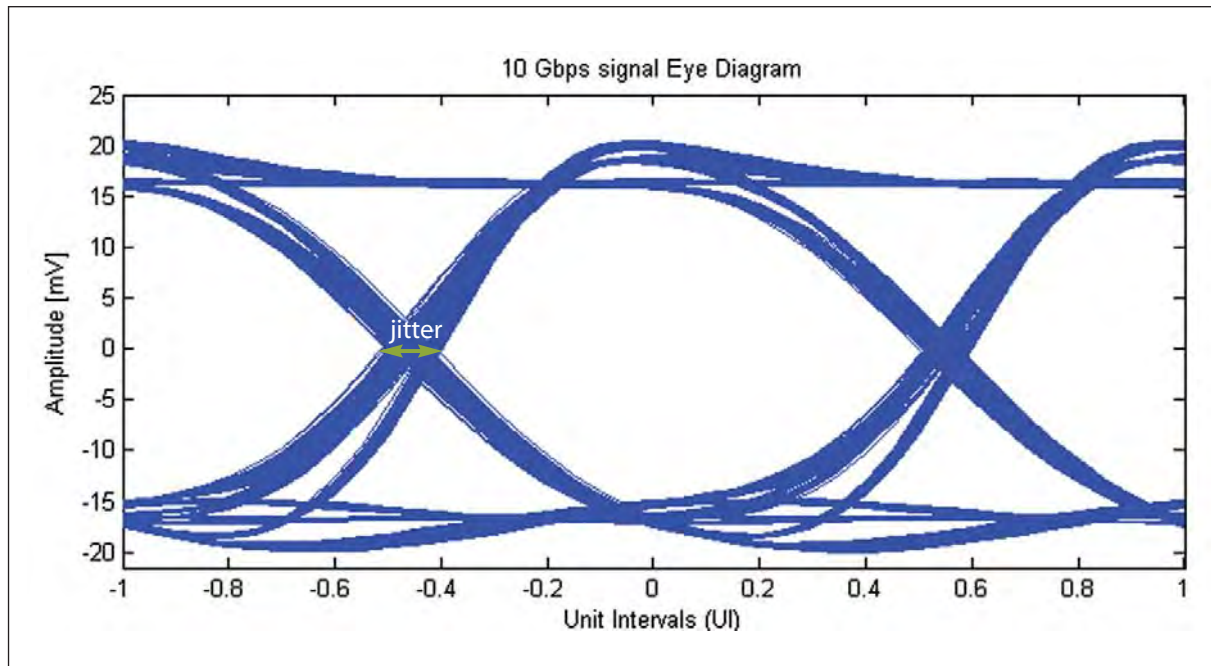
% descarte del primer y último valor (el resultado final es un vector que contiene los valores de jitter de la señal)

En la figura a continuación se grafican simultáneamente con respecto al tiempo, una sección de una trama SONET OC-192 (en rojo) y el jitter (en azul) presente

en las transiciones binarias existentes en dicha sección. Nótese que se toma como punto de referencia el primer cruce por cero. El valor pico-pico del jitter encontrado

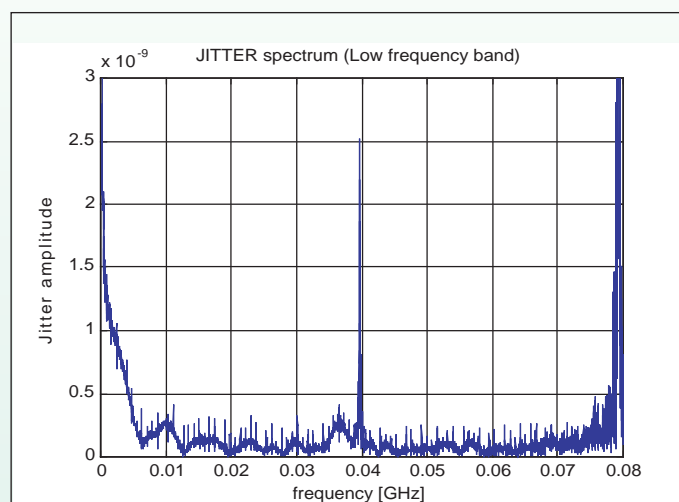
tiene un valor de alrededor de 100 mUI, lo cual concuerda con el jitter que puede apreciarse someramente en el diagrama del ojo de la señal.





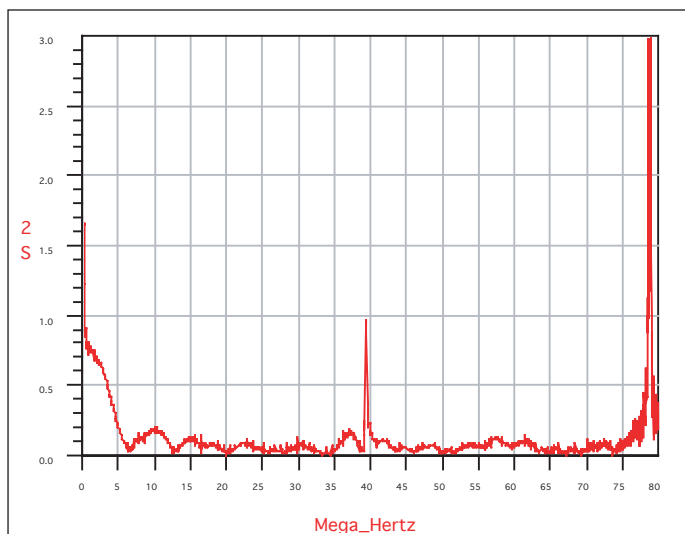
**Finalmente, se puede comprobar que la obtención del jitter empleando MatLab da resultados confiables al comparar el espectro de frecuencia del jitter calculado...**

Finalmente, se puede comprobar que la obtención del jitter empleando MatLab da resultados confiables al comparar el espectro de frecuencia del jitter calculado, el cual se obtiene empleando la transformada rápida de Fourier en MatLab, y el espectro de frecuencia medido directamente de la señal (con un osciloscopio o un detector de fase) y visualizado con un analizador de espectros óptico.



**Espectro frecuencial del jitter obtenido con MatLab**





**Espectro frecuencial del jitter medido directamente de la señal digital**

**...los espectros frecuenciales compaginan casi perfectamente, lo que demuestra la fiabilidad del método para la determinación del jitter...**

Como puede observarse, los espectros frecuenciales compaginan casi perfectamente, lo que

demuestra la fiabilidad del método para la determinación del jitter empleando el procesamiento

digital de la señal con un software que permita tal análisis como lo es MatLab.

## Referencias bibliográficas

- Ing. Roberto Bucher, *Introduzione a Matlab*, 7 aprile 2003.
- *MatLab 6.0 Users Guide*, 2003.
- *MatLab 7.0 Users Guide*, 2006.
- *Analyzing Jitter Using a Spectrum Approach*, Techtronix Inc. 2002.
- *Frequency Agile Jitter Measurement System*, Application Note 1267, Agilent Technologies, 2002.
- *Jitter Analysis Techniques for High Data Rates*, Application Note 1432, Agilent Technologies, 2003.
- *Measuring Jitter in Digital Systems*, Application Note 1448-1, Application Note 1432, Agilent Technologies, 2003.
- *Jitter Generation and Jitter Measurements with the Agilent 81134A Pulse Pattern Generator & 54855A Infiniium Oscilloscope*, Agilent Technologies, 2003.
- *Jitter Solutions for Telecom, Enterprise, and Digital Designs*, Agilent Technologies, 2005.