

“Impulsímetro Digital para Canal de Arranque de Reactores de Investigación”

***CARRERA: ESPECIALIZACIÓN EN REACTORES
NUCLEARES Y SU CICLO DE COMBUSTIBLE***

Alumno: Gloria E. Rios
Director: Claudio A. Verrastro

Marzo 2012



UNSAM
UNIVERSIDAD
NACIONAL DE
SAN MARTÍN

Índice

Resumen.....	4
1. Capítulo 1	5
1.1 Introducción.....	5
1.2 Objetivo.....	6
1.3 Antecedentes.....	6
2. Capítulo 2	7
2.1 Caracterización de los parámetros estadísticos de la señal a medir... 7	
2.1.1 Función densidad de probabilidad del flujo	7
2.1.2 Media del flujo.....	8
2.1.3 Varianza del flujo	9
2.1.4 Función densidad de probabilidad del flujo corregido.....	10
2.1.5 Media del flujo corregido.....	11
2.1.6 Varianza del flujo corregido.....	11
2.1.7 Desvío estándar del promedio	12
2.2 Procedimiento de cálculo	13
2.2.1 Descripción	13
2.2.2 Generación de datos de entrada para simulación del procedimiento de cálculo y elección de los parámetros de medición.....	15
2.2.3 Simulación del procesamiento de cálculo.....	15
2.2.4 Constantes y aproximaciones de funciones utilizados	18
2.2.5 Generación de tablas para multiplicación en hardware	20
2.3 Resultados de las simulaciones.....	21
2.3.1 Verificación del cálculo para tasas distintas de cero.	30
2.3.2 Análisis de salto negativo de flujo o ausencia de pulsos	32
3. Capítulo 3	34
3.1 Descripción del equipo	34
3.1.1 Procesamiento: FPGA	35
3.1.2 Alimentación	37
3.1.3 Generación de clock	37
3.1.4 Entrada de pulsos	37
3.1.5 Salidas analógicas proporcionales al flujo lineal y logarítmico.....	38
3.1.6 Display.....	39
3.1.7 Salida de datos serie	40
3.1.8 Frente	40
4. Capítulo 4	42

4.1	Implementación en VHDL.....	42
4.1.1	Diagrama en bloques.....	42
4.1.2	Diagrama funcional del Proceso de Cálculo.....	44
4.1.3	Depuración del código VHDL.....	46
4.1.4	Utilización de recursos de la FPGA.....	47
5.	Capítulo 5	48
5.1	Verificación utilizando el simulador de pulsos aleatorios.....	48
5.1.1	Error por tiempo muerto	48
5.1.2	Software utilizado en la adquisición de datos.....	50
5.1.3	Procedimiento de adquisición de datos.....	52
5.1.4	Tablas de Perfiles	54
5.2	Resultado de las simulaciones.....	54
6.	CONCLUSIONES.....	57
7.	Referencias bibliográficas	58
8.	Anexos.....	58

Resumen

Se diseñó un equipo electrónico, denominado Impulsímetro Digital, para la medición del flujo de cuentas que proporcionan los detectores del canal de arranque en régimen de pulsos de un reactor de investigación. El procesamiento es completamente digital y está basado en tecnología FPGA: "Arreglo Lógico Configurable". Estos dispositivos consisten en un conjunto muy grande de compuertas lógicas, elementos de memoria y los recursos de interconexión para poder implementar funciones lógicas complejas.

Los rangos de flujo a medir en el canal de arranque van de 100 cuentas/segundo hasta alrededor de 10^5 cuentas/segundo, pudiendo el equipo medir desde 1 hasta 10^6 cuentas/segundo. El error para flujos altos depende del tiempo de respuesta del detector y del ancho de pulso del conformador de pulsos del detector.

En el equipo se implementa un procedimiento de cálculo que utiliza un esquema de medición de cuentas prefijadas variable, donde se consigue un error estadístico controlado que es además filtrado para suavizar la curva del resultado. Basándose en el conocimiento del error estadístico se realizan tests de hipótesis para determinar si se debe o se puede cambiar el parámetro del filtro. El objetivo es mantener un error máximo dado para flujos bajos, bajar el error cuando el aumento de flujo lo permita y adaptarse rápidamente a un cambio de flujo.

Al utilizar un esquema de cuentas prefijadas resulta muy simple la capacidad de adaptación a los cambios de flujo. Si el flujo aumenta, rápidamente se acumulan cuentas y se obtiene un resultado con menor error estadístico. El método de cálculo propuesto no limita la cantidad de cuentas, sino el tiempo máximo en que éstas se acumulan a algunos milisegundos. Por lo tanto, si el valor de flujo es alto, la próxima medición se obtiene en algunos milisegundos. Si el flujo es bajo, se toma el tiempo necesario para acumular una cantidad dada de cuentas que aseguren un cierto error estadístico.

La puesta en marcha del algoritmo se hizo en Matlab. La implementación en VHDL se depuró utilizando resultados del cálculo en Matlab y se las comparó con la simulación que provee la herramienta de desarrollo mediante la aplicación Modelsim. Finalmente el hardware se verificó con el módulo Simulador de Flujo desarrollado en el sector Instrumentación y Control de CNEA.

1. Capítulo 1

1.1 Introducción

Los canales de arranque de los reactores de investigación miden en las primeras 4 a 5 décadas de potencia. La información de flujo neutrónico es proporcionada por contadores proporcionales de BF3 o de fisión, que se encuentran dentro del reactor. La señal generada es un tren de pulsos de corriente, cuya tasa de ocurrencia es proporcional al flujo neutrónico. Para cumplir con el criterio de redundancia los canales están triplicados o duplicados, como es el caso del RA1. Cada canal cuenta con su detector e instrumentación asociados independiente.

En el arranque de un reactor de baja potencia, el flujo neutrónico se mide en modo pulso con detectores muy sensibles como son los de trifluoruro de boro, que generan un pulso de corriente por cada neutrón que interactúa con el gas de BF3. Como la información de interés es la cantidad de pulsos por unidad de tiempo y no la energía de la radiación incidente, no se colecta toda la carga generada en el detector, sino que se acorta el ancho de pulso haciendo que la constante del circuito preamplificador menor que el tiempo de colección [1]. Para flujos altos se producirá un solapamiento entre los pulsos que dependerá del ancho de pulso que se logre en el preamplificador, y que llevará a una medición de flujo menor que la real. El error de la medición dependerá del método de medición del flujo.

La instrumentación requerida consiste en un preamplificador de la señal de corriente. Éste genera un pulso de algunas centenas de nanosegundos de ancho de pulso, de amplitud constante de algunos volts. Esta señal es leída por los instrumentos que miden flujo $\varphi(t)$ y tasa de variación $Tita$, donde:

$$\varphi(t) = \varphi_0 \cdot e^{Tita \cdot t}$$

Tradicionalmente se generan señales analógicas proporcionales al flujo y a la tasa de variación que se comparan con niveles de disparo por alto y/o por bajo para detectar condición de scram o inhibición de subida de barras.

En este trabajo se presenta un módulo medidor de flujo cuya única entrada es la señal de pulsos conformados de ancho y amplitud fijos que genera el amplificador de corriente, y que genera dos señales analógicas, una proporcional al flujo con rango lineal por décadas y otra proporcional al logaritmo del flujo. La señal de flujo lineal va acompañada por 6 señales digitales que indican la escala. El valor de flujo lineal y la escala se muestran en un display 7 segmentos de 4 dígitos y una indicación del error estadístico del valor mostrado.

El rango de medición es de 1c/s a 10^6 c/s porque se prevé su utilización en un canal de arranque-marcha que utiliza el método Campbell para la medición de flujo en el rango intermedio entre modo pulso y modo corriente. Para el uso en un sistema Campbell se requiere que los instrumentos que miden en los distintos 'modos de medición', se solapen en al menos 1 década para asegurar que no exista un punto ciego de medición.

El método de cálculo de flujo utilizado se basa en la acumulación de una cantidad de pulsos conocida, lo que permite una mejor adaptación a los cambios de flujo, que a su vez resulta en tiempos de respuesta ajustados al valor de flujo con errores estadísticos acotados. El método resuelve automáticamente el compromiso entre precisión y tiempo de respuesta con una velocidad que un usuario, utilizando controles externos, no puede lograr.

1.2 Objetivo

El objetivo del trabajo es desarrollar un equipo medidor de flujo neutrónico por conteo de pulsos para el canal de arranque de reactores de investigación, que facilite y mejore la operación, resolviendo automáticamente los cambios de parámetros necesarios para medir el flujo con menor error estadístico, y con un tiempo de respuesta menor a 50ms a partir de un cierto valor de flujo a determinar.

1.3 Antecedentes

El desarrollo del módulo impulsímetro digital surge del planteo de modernización del canal de arranque del RA1, que está compuesto por los equipos desarrollados también en el sector de instrumentación y control:

- Mod CNEA 031: amplificador conformador
- Mod CNEA 041: amplificador monocanal
- Mod CNEA 051: escalímetro
- Mod CNEA 061: impulsímetro
- Mod CNEA 073: comparador

Para reemplazar a los amplificadores CNEA 031 y CNEA 041 se desarrolló en el sector de Instrumentación y Control un amplificador discriminador – conformador modelo K660, que cumple con las dos funciones. Este nuevo amplificador es el que provee la señal de pulso para el nuevo impulsímetro digital. Éste a su vez reemplaza a los módulos CNEA 051 y CNEA 061, para la generación de las señales proporcionales al flujo logarítmico y al flujo lineal.

El módulo CNEA 061 genera una señal proporcional a la tasa de variación del flujo. En el nuevo canal de arranque esta señal se obtiene de un módulo impulsímetro digital configurado como medidor de tasa de variación. El módulo se diseñó con el hardware necesario para proveer ambas funcionalidades, cambiando la configuración de la FPGA.

La primera versión del impulsímetro digital fue desarrollada con un microcontrolador MC68705U3, en el que se utilizaba el mismo esquema de cuentas prefijadas y promediación variable según el valor del flujo que se utiliza en el módulo presentado en este trabajo. El rango de medición era de 1c/s a 10^6 c/s. Estos dispositivos presentan la desventaja de utilizar interrupciones, aunque se limite al máximo su utilización. Con la tecnología FPGA en cambio, se dispone de hardware dedicado para cada función implementada, las que se realizan paralelamente. Además existen para éstos dispositivos de lógica programable, herramientas y métodos que facilitan el proceso de verificación y validación, pudiéndose alcanzar niveles de confiabilidad mucho mejores que para desarrollos basados en software.

2. Capítulo 2

2.1 Caracterización de los parámetros estadísticos de la señal a medir

Dado que la señal a medir es una variable aleatoria y el método de medición requiere del conocimiento de la distribución estadística del flujo en función del parámetro de medición “K : cuentas acumuladas”, se requiere determinar la distribución del flujo calculado, y las expresiones de los parámetros estadísticos necesarios.

2.1.1 Función densidad de probabilidad del flujo

Los pulsos generados en los detectores corresponden a un fenómeno Poisson con parámetro μ . El tiempo entre pulsos, que es el parámetro que se mide, tiene una función distribución de probabilidad (fdp en adelante) exponencial con parámetro $\beta=\mu$. En el proceso se suman $x = K$ tiempos entre pulsos para cada medición. Esta suma de tiempos tiene distribución Gamma con parámetros $\alpha = K$ y $\beta = \mu$, cuya función densidad de probabilidad es la siguiente:

$$f(x) = \frac{1}{\mu^K \cdot \Gamma(K)} \cdot x^{K-1} \cdot e^{-\frac{x}{\mu}}, \text{ para } x > 0 \quad (1.)$$

donde

K : es la cantidad de tiempos entre pulsos sumados, y

μ : el tiempo medio entre pulsos

El flujo es calculado como la inversa de suma de tiempo dividida por la cantidad de cuentas acumuladas, K:

$$\varphi = \frac{K}{x} \quad (2.)$$

La función densidad de probabilidad de la variable aleatoria φ , $g(\varphi)$ se calcula como ([3], Página 186):

$$g(\varphi) = f(w(\varphi)) \cdot |J_W| \quad (3.)$$

donde:

$$w(\varphi) = \frac{K}{\varphi} \quad (4.)$$

$$|J_W| = \frac{K}{\varphi^2} \quad (5.)$$

Reemplazando (4) en (1) y multiplicando por el Jacobiano se obtiene la distribución de probabilidad del flujo calculado como la inversa de la acumulación de una cantidad K de tiempos entre pulsos dividida por K:

$$g(\varphi) = \frac{1}{\mu^K \cdot \Gamma(K)} \cdot K^K \cdot \left(\frac{1}{\varphi}\right)^{K+1} \cdot e^{-\frac{K}{\mu \cdot \varphi}} , \text{ para } \varphi > 0 \quad (6.)$$

2.1.2 Media del flujo

La media del flujo calculado se calcula como:

$$E[\varphi] = \int_0^{\infty} \varphi \cdot g(\varphi) \partial \varphi \quad (7.)$$

$$E[\varphi] = \frac{1}{\mu^K \cdot \Gamma(K)} \int_0^{\infty} \left(\frac{K}{\varphi}\right)^K \cdot e^{-\frac{K}{\mu \cdot \varphi}} \partial \varphi \quad (8.)$$

Haciendo un cambio de variable:

$$z = \frac{K}{\varphi} , \quad \partial z = -\frac{K}{\varphi^2} \partial \varphi \quad (9.)$$

$$\partial \varphi = -\frac{\varphi^2}{K} \partial z = -\frac{K}{z^2} \partial z \quad (10.)$$

Reemplazando (10) en (8):

$$E[\varphi] = \frac{K}{\mu^K \cdot \Gamma(K)} \int_0^{\infty} z^{K-2} \cdot e^{-\frac{z}{\mu}} \partial z \quad (11.)$$

Esta expresión coincide con la fórmula 15.76 de la tabla de fórmulas [1]:

$$\int_0^{\infty} y^n \cdot e^{-a \cdot y} \partial y = \frac{\Gamma(n+1)}{a^{n+1}} \quad (12.)$$

Entonces:

$$E(\varphi) = \frac{K}{\mu^K \cdot \Gamma(K)} \cdot \frac{\Gamma(K-1)}{\left(\frac{1}{\mu}\right)^{K-1}} \quad (13.)$$

Por propiedad reproductiva de la función Gamma:

$$\Gamma(n) = \frac{\Gamma(n+1)}{n} \quad (14.)$$

Reemplazando (14) en (13) se obtiene la media del flujo calculado con (2.). Se observa que la media tiene un sesgo que depende de K.

$$E(\varphi) = \frac{K}{K-1} \cdot \frac{1}{\mu} \quad (15.)$$

2.1.3 Varianza del flujo

La varianza del flujo calculado es:

$$\sigma_{\varphi}^2 = \int_0^{\infty} (\varphi - E(\varphi))^2 \cdot g(\varphi) \partial \varphi \quad (16.)$$

Expandiendo el binomio:

$$\sigma_{\varphi}^2 = \underbrace{\int_0^{\infty} (\varphi)^2 \cdot g(\varphi) \partial \varphi}_I - 2 \cdot E(\varphi) \underbrace{\int_0^{\infty} \varphi \cdot g(\varphi) \partial \varphi}_{II} + E(\varphi)^2 \underbrace{\int_0^{\infty} g(\varphi) \partial \varphi}_{III} \quad (17.)$$

Cálculo de I:

$$I = \frac{K}{\mu^K \cdot \Gamma(K)} \int_0^{\infty} \left(\frac{K}{\varphi}\right)^{K-1} \cdot e^{\mu \cdot \varphi} \partial \varphi \quad (18.)$$

Haciendo el cambio de variable (9) y utilizando la (12) obtenida de tabla, resulta:

$$I = \frac{K^2}{\mu^K \cdot \Gamma(K)} \frac{\Gamma(K-2)}{\left(\frac{1}{\mu}\right)^{K-2}} \quad (19.)$$

Utilizando la propiedad (14):

$$I = \frac{K^2}{(K-2) \cdot (K-1)} \frac{1}{\mu^2} \quad (20.)$$

Cálculo de II: la integral coincide con el cálculo de la media, entonces:

$$II = 2 \cdot E(\varphi)^2 = \frac{2 \cdot K^2}{(K-1)^2 \cdot \mu^2} \quad (21.)$$

Cálculo de III: la integral da '1' por propiedad de la función densidad de probabilidad, entonces:

$$III = E(\varphi)^2 = \frac{K^2}{(K-1)^2 \cdot \mu^2} \quad (22.)$$

Reemplazando (20), (21) y (22) en (17) se obtiene la varianza del flujo:

$$\sigma_{\varphi}^2 = \frac{K^2}{(K-1)} \frac{1}{\mu^2} \left(\frac{1}{(K-2)} - \frac{1}{(K-1)} \right) \quad (23.)$$

Es interesante notar que para los valores de $K=2$ y $K=1$ la varianza no está definida.

Valores bajos de K podrían ser útiles en la medición del flujo en el arranque de una central nuclear, donde al no haber fondo gamma proveniente de productos de fisión, no hay fotoneutrones, por lo que la medición de fondo puede llegar a ser de 10^{-3} cuentas/segundo.

En la Figura 1 se grafican las funciones densidad de probabilidad de flujo calculadas con (6), para K igual a 2, 4, 8, 16, 32 y 64. Las barras horizontales están centradas en la media de los datos calculada con (15), e indican los valores ± 1 sigma, calculados con (23). Se calcula para flujo medio igual a 1 c/s.

Se observa en la Figura 1 que la fdp se vuelve cada vez más simétrica a medida que aumenta K , y que el sesgo de la media se va haciendo despreciable. Sin embargo, como se prevé usar valores bajos de K , como ser $K = 16$ o $K = 32$, se debe corregir el valor de flujo obtenido para eliminar el error que introduce la forma de cálculo. Como se verá en 2.1.6, utilizar el flujo corregido simplifica el cálculo del desvío estándar, lo que es beneficioso para la implementación en hardware.

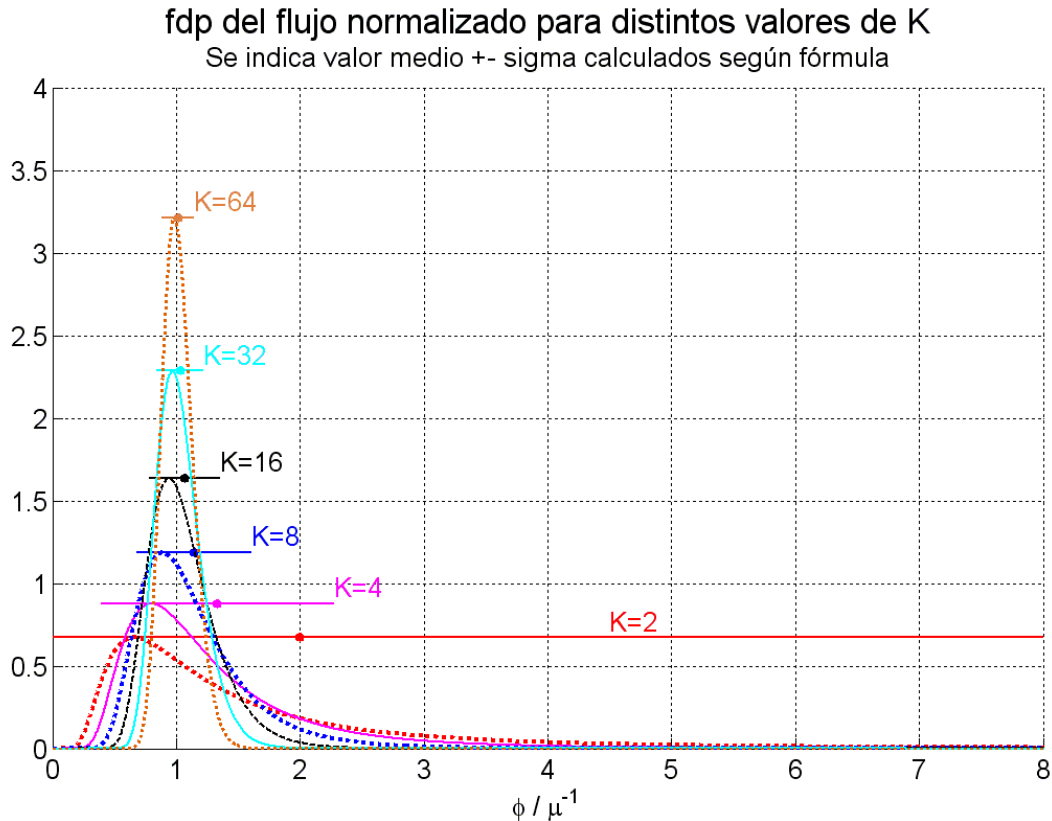


Figura 1. . FDP del flujo calculado con (6.) y sigma calculada a partir de la (23.) para distintos K .

2.1.4 Función densidad de probabilidad del flujo corregido.

Para eliminar el sesgo de la media (15.), el flujo corregido es calculado como:

$$\varphi_{corr} = \varphi \cdot \frac{K-1}{K} \quad (24.)$$

La función densidad de probabilidad de la variable aleatoria φ_{corr} , se calcula como:

$$g(\varphi) = f(w(\varphi)) \cdot |J_W| \quad (25.)$$

donde:

$$w(\varphi_{corr}) = \frac{K}{K-1} \cdot \varphi_{corr} \quad (26.)$$

y

$$|J_W| = \frac{K}{K-1} \quad (27.)$$

Entonces reemplazando (26.) en (1.) y multiplicando por la (27.):

$$g(\varphi_{corr}) = \frac{K}{K-1} \frac{K^K}{\mu^K \cdot \Gamma(K)} \cdot \left(\frac{1}{\frac{K}{K-1} \cdot \varphi_{corr}} \right)^{K+1} \cdot e^{-\frac{K}{\mu \cdot \frac{K}{K-1} \cdot \varphi_{corr}}} \quad (28.)$$

Simplificando:

$$g(\varphi_{corr}) = \left(\frac{K-1}{K} \right)^K \frac{K^K}{\mu^K \cdot \Gamma(K)} \cdot \left(\frac{1}{\varphi_{corr}} \right)^{K+1} \cdot e^{-\frac{K}{\mu \cdot \varphi_{corr}}} \cdot e^{\frac{1}{\mu \cdot \varphi_{corr}}} \quad (29.)$$

2.1.5 Media del flujo corregido

Por propiedad de la media [3]:

$$E(aX + b) = aE(X) + b \quad (30.)$$

Entonces:

$$E(\varphi_{corr}) = \frac{K-1}{K} E(\varphi) = \frac{1}{\mu} \quad (31.)$$

2.1.6 Varianza del flujo corregido

Por propiedad de la varianza [3]:

$$\sigma_{aX+b}^2 = a^2 \sigma_X^2 \quad (32.)$$

Entonces:

$$\sigma_{\varphi_{corr}}^2 = \left(\frac{K-1}{K} \right)^2 \sigma_{\varphi}^2 \quad (33.)$$

$$\sigma_{\varphi_{corr}}^2 = \frac{1}{(K-2)} \frac{1}{\mu^2} \quad (34.)$$

2.1.7 Desvío estándar del promedio

Se utiliza un promedio móvil con pesos exponencialmente decrecientes (EWMA) que da más peso a las muestras recientes que a las antiguas [8]. La fórmula utilizada para el cálculo del promedio es:

$$P_i = P_{i-1} + \frac{\varphi_i - P_{i-1}}{N} \quad (35.)$$

donde φ_i es el flujo calculado con un cierto 'K', P_{i-1} en el promedio anterior y P_i es el promedio actual. Se utiliza la fórmula recursiva dado que simplifica la implementación en hardware. Además este promedio tiene mayor efectividad en bajar la dispersión comparado con un promedio móvil estándar [8].

La ecuación del promedio se puede escribir como una ecuación en diferencias con coeficientes constantes:

$$Y_n + \left(\frac{1}{N} - 1 \right) Y_{n-1} = \frac{1}{N} X_n \quad (36.)$$

Donde Y es el promedio y X es el nuevo valor de flujo calculado. La varianza del promedio se obtiene utilizando la respuesta al impulso del filtro. Si se considera $\sigma_{\varphi_K}^2$ constante resulta:

$$\sigma_{\varphi_{i,N}}^2 = \sigma_{\varphi_{i,K}}^2 \cdot \sum_i h(i)^2 \quad (37.)$$

De la fórmula (36.), la transferencia resulta:

$$H(e^{j\omega}) = \frac{\sum_{i=0}^M b_n \cdot e^{-j\omega i}}{\sum_{i=0}^L a_n \cdot e^{-j\omega i}} = \frac{1/N}{1 - \left(1 - \frac{1}{N}\right) e^{-j\omega}} \quad (38.)$$

La respuesta al impulso correspondiente es [5]:

$$h(n) = \frac{1}{N} \left(1 - \frac{1}{N}\right)^n \quad (39.)$$

Reemplazando en (37):

$$\sigma_{\varphi_{i,N}}^2 = \sigma_{\varphi_{i,K}}^2 \cdot \sum_i \frac{1}{N}^2 \left(1 - \frac{1}{N}\right)^{2 \cdot i} \quad (40.)$$

La sumatoria es la serie geométrica, que converge para $r < 1$:

$$\sum_i a \cdot r^i = \frac{a}{1-r} \quad (41.)$$

Entonces la ecuación (40) converge a:

$$\sigma_{\varphi_{i,N}}^2 = \sigma_{\varphi_{i,K}}^2 \cdot \frac{1}{2N-1} \quad (42.)$$

El desvío estándar del promedio resulta:

$$\sigma_{\varphi_{i,N}} = \frac{\sigma_{\varphi_{i,K}}}{\sqrt{2N-1}} \quad (43.)$$

2.2 Procedimiento de cálculo

2.2.1 Descripción

Para poder dar una medición con un error aceptable se acumulan cuentas de modo de bajar el error estadístico. Es decir se pone un límite mínimo de cuentas que se debe acumular para tener lo que se llamará '*flujo instantáneo*', la medición más rápida de flujo en el cálculo. Este valor mínimo de cuentas se determinará en el párrafo 2.2.2.

El asegurar un cierto valor del error estadístico implica aumentar el tiempo de respuesta para los flujos más bajos, donde es imposible cumplir con el requerimiento de tener una medición en menos de algunas decenas de milisegundos. Esto no supone un riesgo, dado que si el flujo aumenta, rápidamente se acumularán las cuentas necesarias y se tendrá una nueva medición.

Ahora si el flujo es suficientemente alto, se podrán juntar más cuentas que las mínimas, y así bajar el error estadístico. Entonces lo que se establece es un tiempo máximo durante el cual acumular cuentas. Este tiempo de acumulación, denominado '*TACUM*' se determinará en el párrafo 2.2.2.

Una vez obtenido el flujo instantáneo con un error máximo dado, se promedia este valor, ya que el mismo tiene una gran dispersión como se muestra en las gráficas de distribución de flujo en función de las cuentas acumuladas (Figura 1).

El parámetro de promediación '*N*' cambia adaptándose a las variaciones de flujo para asegurar una medición más estable y con menor error para mostrar al usuario, y a la vez poder seguir a las variaciones cuando éstas sucedan. Para lograr esto se realiza un test de hipótesis cada vez que se obtiene un nuevo valor de flujo instantáneo, para decidir si se debe bajar la promediación o si se puede aumentar, bajando o subiendo el parámetro *N* del filtro.

Para decidir si se debe disminuir el *N* se plantea un test de hipótesis que prueba si el promedio $\bar{\varphi}_{i,N}$ es el valor medio del flujo instantáneo $\varphi_{i,K}$. La hipótesis nula H_0 es entonces:

$$H_0 : E(\varphi_{i,K}) = \bar{\varphi}_{i,N} \quad (44.)$$

El intervalo de confianza que se utiliza es una cantidad de $\pm N$ SIGMAS veces el desvío estándar del flujo instantáneo. El valor final de N SIGMAS que se utilizará se determina en el párrafo 2.2.2. Si el promedio $\bar{\varphi}_{i,N}$ resulta fuera del intervalo de confianza, se considera que se está alejando de la media del flujo actual, y se decide bajar N a, $N_{i+1} = \frac{N_i}{4}$.

La potencia de la prueba para los valores más bajos de K utilizados varía según el flujo aumente o disminuya, debido a la asimetría de la función densidad de probabilidad. En la Figura 2 se grafican un salto de flujo por 2 y a la mitad, y se muestra la potencia del test para el caso $K = 32$, utilizando N SIGMAS = 3. Se ve que en este caso es más probable detectar una disminución del flujo que un aumento en la misma proporción.

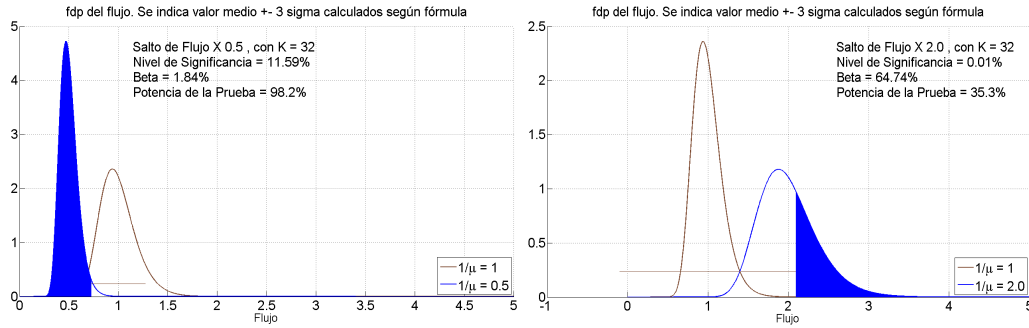


Figura 2. Potencia de prueba de hipótesis para $K = 32$ para salto de flujo a la mitad y al doble

En la Figura 3 se observa que la potencia de la prueba aumenta para mayor cantidad de cuentas acumuladas y no hay diferencia para saltos mayores o menores a 1 dado que la curva se aproxima a una gaussiana y se corrige la asimetría.

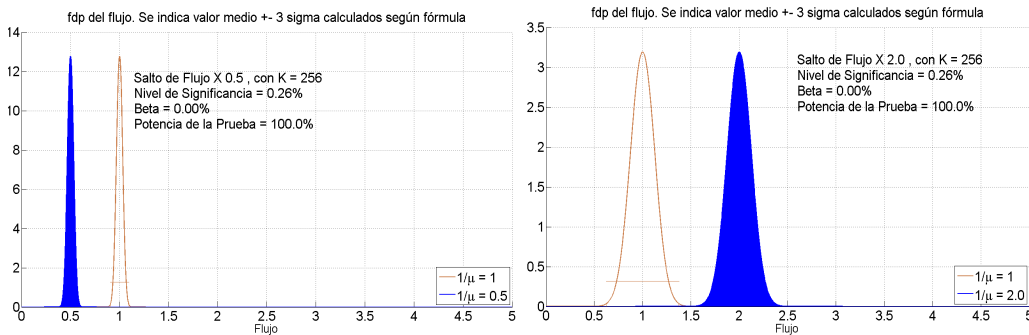


Figura 3. Potencia de prueba de hipótesis para $K = 256$ para salto de flujo a la mitad y al doble

Si no es necesario bajar N , se plantea subirlo para mostrar al usuario una medición con menor error y más estable. Para eso se lleva el promedio doble

$\bar{\varphi}_{i,2N}$. Si éste resulta dentro del intervalo de confianza del $\pm \frac{\sigma \bar{\varphi}_{i,N}}{4}$, se considera que el promedio $\bar{\varphi}_{i,2N}$ no está atrasado respecto al $\bar{\varphi}_{i,N}$ y se aumenta N al doble, $N_{i+1} = 2 \cdot N_i$, hasta un máximo de 256.

2.2.2 Generación de datos de entrada para simulación del procedimiento de cálculo y elección de los parámetros de medición

Para evaluar los parámetros de medición más convenientes se utiliza el software Matlab, con él que se generan conjuntos de datos de tiempos entre pulsos correspondientes a un proceso Poisson con parámetro $\lambda = \varphi(t)$, para diferentes flujos y tasas de variación. La distribución de probabilidad de los tiempos es una exponencial con parámetro $\mu = \frac{1}{\varphi(t)}$, variando el flujo teórico con la tasa de variación 'Tita' según la fórmula:

$$\varphi(t) = \varphi_0 \cdot e^{Tita \cdot t} \quad 45.$$

Se generaron un conjunto de simulaciones para flujos entre 1 y 10^6 c/s de a décadas, con tasa de variación igual a cero y salto por 2 aproximadamente en la mitad del tiempo de simulación, sin simulación de ancho de pulso real. Todos los conjuntos de datos se simularon con un salto del flujo para evaluar el retardo de la respuesta del flujo calculado. Se eligió un salto pequeño ya que es el peor caso para el test de hipótesis.

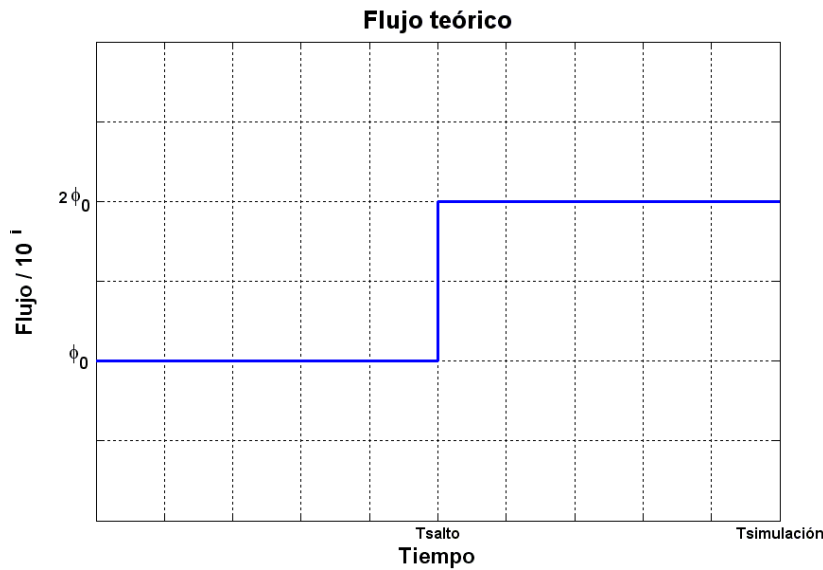


Figura 4. Perfil teórico generado en cada corrida con $\varphi_0 = 10^i$ para $i = 1$ a 6.

Se simuló el procedimiento de cálculo para distintos valores de los parámetros a determinar generando 100 corridas para cada caso de combinación de los parámetros. Se calcularon algunas variables de interés del flujo resultado, y en función de éstas se eligieron los valores de los parámetros más convenientes como se describe en 2.3.

2.2.3 Simulación del procesamiento de cálculo.

Para la simulación del procesamiento de cálculo se utilizó el Matlab. Se implementaron todas las aproximaciones de cálculo que serán necesarias para el traspaso al hardware, párrafo 2.2.4. De este modo los resultados de simulación de cálculo servirán luego para la puesta en marcha del código

VHDL por comparación de los vectores resultado y de las variables internas del algoritmo de cálculo.

En lo que sigue de la sección se describen los pasos del algoritmo escrito para Matlab, donde aparecen las variables y constantes que luego se utilizan en el código VHDL. Como se verá, todas las variables se escriben como potencias de 2 para facilitar la implementación en hardware, donde las divisiones y multiplicaciones se realizan como desplazamiento de bits.

Se acumulan los tiempos entre pulsos de a $K = 16 \cdot 2^j$ pulsos, con $j \in [0; 13]$. El valor mínimo de K es 16, por lo que se expresa a la cantidad como un múltiplo de $K_j = 16$.

El flujo se calcula con la fórmula corregida [24.]:

$$\varphi_{corr} = \frac{\text{Pulsos contados}}{\text{Tiempo acumulado}} \cdot \frac{K-1}{K} \quad 46.$$

El criterio para determinar la cantidad de pulsos a acumular tiene 2 condiciones.

La primera es acumular pulsos durante al menos T_ACUM segundos.

La segunda condición es mantener el error del flujo sin promediar debajo de un cierto valor, que es el parámetro de simulación 'ERROR_INFERIORF'. El flujo es una variable aleatoria cuya desviación estándar tiene la siguiente expresión:

$$\sigma_\varphi = \sqrt{\frac{\varphi^2}{(K-2)}} \quad 47.$$

Este segundo criterio es el determinante para flujos bajos. Para flujos grandes (> 1000), es necesario acumular más pulsos para superar los 10 milisegundos, entonces el error será siempre menor a 'ERROR_INFERIORF'.

Para simular el cálculo en el dispositivo de hardware se realiza el cálculo como se implementará allí, esto es aplicando logaritmos para convertir las divisiones en restas, y luego obteniendo el antilogaritmo. Para simular las funciones con precisión de 8 bits que se utilizarán, se generaron las funciones 'logaritmo2' y 'antilogaritmo2' en Matlab.

El flujo instantáneo entonces se calcula con la siguiente fórmula:

$$\varphi_K = \text{antilogaritmo2} \left(\logaritmo2 \left(\frac{K}{T_{acumulado}} \cdot \frac{K-1}{K} \right) \right) \quad 48.$$

$$\varphi_K = \text{antilogaritmo2} (\logaritmo2 (K) - \logaritmo2 (T) + \logaritmo2 (F_{Correccion}))$$

Dado que el tiempo en el hardware se lleva con una base de tiempo que tiene un cierto 'periodo', el tiempo acumulado se expresa en cuentas de esa base de tiempo : $\frac{T}{\text{Periodo}}$. Entonces en la ecuación anterior se multiplica y divide por el 'periodo':

$$\varphi_K = \text{antilogaritmo2} \left(\logaritmo2 \left(\frac{K_i \cdot 2^j}{\text{Periodo}} \right) - \logaritmo2 \left(\text{floor} \left(\frac{T}{\text{Periodo}} \right) \right) \right) + \logaritmo2(F_{\text{Correccion}})$$

$$\varphi_K = \text{antilogaritmo2} \left(\logaritmo2 \left(\frac{K_i}{\text{Periodo}} \right) + j - \logaritmo2 \left(\text{floor} \left(\frac{T}{\text{Periodo}} \right) \right) \right) + \logaritmo2(F_{\text{Correccion}})$$

49.

El primer término es una constante y el tercer término es una tabla de constantes, que se determinan con 8 bits decimales, en el párrafo 2.2.4.

Una vez obtenido el valor del flujo instantáneo, se hallan el promedio N y $2N$, con $N = 2^C$, con las siguientes fórmulas:

$$\bar{\varphi}_N(i) = \bar{\varphi}_N(i-1) + \frac{\varphi_{i,K} - \bar{\varphi}_N(i-1)}{N} \quad 50.$$

$$\bar{\varphi}_{2N}(i) = \bar{\varphi}_{2N}(i-1) + \frac{\varphi_{i,K} - \bar{\varphi}_{2N}(i-1)}{2 \cdot N} \quad 51.$$

Finalmente, el valor de N se modifica si se cumplen los siguientes criterios:

- Si el nuevo valor de $\varphi_{i,K}$ se aparta más de NSIGMAS sigmas del valor promedio, N baja: $N = \frac{N}{4} = 2^{C-2}$.
- Sino, si el promedio $2N$ está dentro de $\pm \frac{\sigma_{\bar{\varphi}_{i,N}}}{4}$, N sube: $N = 2 \cdot N = 2^{C+1}$. El sigma del promedio se calcula con [45].

$$\sigma_{\bar{\varphi}_{i,N}} = \frac{\sigma_{\varphi_{i,K}}}{\sqrt{2N-1}} \quad 52.$$

Se hace disminuir N más rápido para poder seguir más rápidamente el valor de flujo si éste tiene variaciones. Se fijó $N = 256$ ($C_{\text{MAX}} = 8$) como valor máximo.

Finalmente se obtiene el flujo lineal y la escala. El flujo lineal se obtiene con la fórmula [53.] con 12 bits de resolución, que es la resolución del DAC que se usa para generar la señal analógica proporcional al flujo promedio en el equipo. Cada escala va de $1 \cdot 10^{\text{Escala}}$ a $10 \cdot 10^{\text{Escala}}$. Este cálculo se introduce en la simulación para hacer la verificación del código VHDL.

$$\varphi_{\text{lineal}} = \frac{\varphi_{\text{Pr om}} \cdot \frac{2^{12}}{10}}{10^{\text{Escala}}} \quad 53.$$

La escala se maneja con histéresis. Debido al cero vivo que está sumado a la señal de salida, no se permite el valor de fondo de escala para no sobrepasar los 10V de salida, y el cambio se hace en un valor menor.

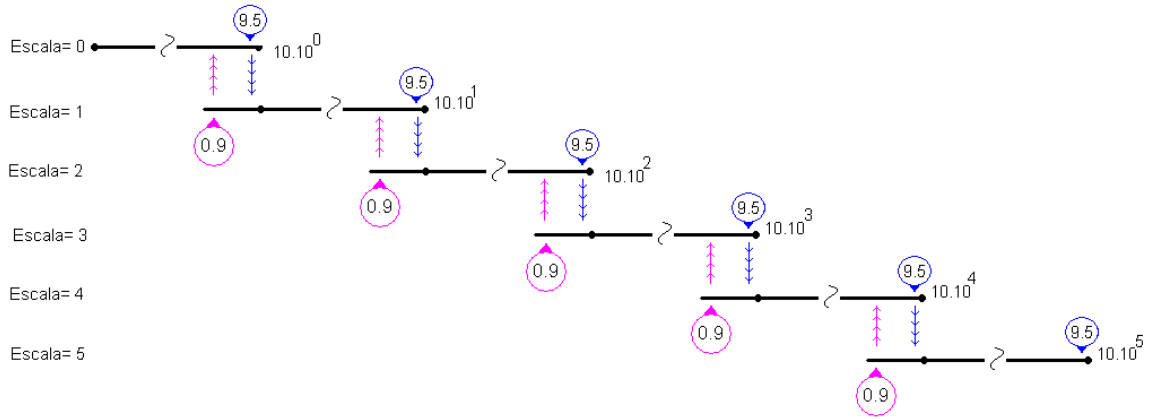


Figura 5. Cambio de escala.

Para el cálculo de la fórmula [53], se recurre a logaritmos nuevamente.

$$\varphi_{lineal} = \text{Antilogaritmo2}(\text{logaritmo2}(\varphi_{Pr om}) + \text{logaritmo2}(409.6) - \text{logaritmo2}(10^{Escala})) \quad 54.$$

Los 2 últimos términos son constantes que se determinan con 8 bits decimales como el resultado del logaritmo calculado por tabla en el párrafo 2.2.4.

En el hardware también se calcula el flujo logarítmico con la fórmula [55.], que es una señal de 0 a 10V de 12 bits de resolución. Cada década del valor de flujo corresponde a 1V de esta señal de salida, por lo tanto el rango de tensión cubre un rango mayor que el de medición del equipo.

$$\varphi_{log} = \text{Log}_{10}(\varphi_{Pr om}) \cdot \frac{2^{12}}{10} \quad 55.$$

Para el cálculo en hardware se utilizan logaritmos base 2. La fórmula anterior se calcula entonces como:

$$\varphi_{log} = \frac{\text{logaritmo2}(\varphi_{Pr om})}{\text{logaritmo2}(10)} \cdot \frac{2^{12}}{10} \quad 56.$$

$$\varphi_{log} = \text{antilog2}(\text{logaritmo2}(\varphi_{log})) \quad 57.$$

$$\varphi_{log} = \text{antilog2} \left(\text{logaritmo2}(\text{logaritmo2}(\varphi_{Pr om})) + \text{logaritmo2} \left(\frac{2^{12}}{10 \cdot \text{logaritmo2}(10)} \right) \right) \quad 58.$$

El último término es una constante que se determina con 8 bits decimales como el resultado del logaritmo calculado por tabla en el párrafo 2.2.4.

2.2.4 Constantes y aproximaciones de funciones utilizados

Dado que se simula emulando al código VHDL, se aproximan todas las constantes a 8 bits decimales. A continuación se calculan todas las constantes necesarias para el cálculo y se las deja expresadas en notación hexadecimal.

- I. Constantes para cálculo del flujo instantáneo utilizados en la fórmula [49.]

$$CTE_PERIODO = \text{logaritmo2}\left(\frac{K_i}{Periodo}\right) = \text{logaritmo2}\left(\frac{16}{8\mu\text{seg}}\right) \quad 59.$$

$$CTE_PERIODO = 14. EE_H \quad 60.$$

K	CTE_CORREGIDO = $\text{logaritmo2}(F_{Correccion}) = \text{logaritmo2}\left(\frac{K-1}{K}\right)$ (el valor de la constante es negativo, se guarda el valor absoluto y luego se resta)
16	0.18 _H
32	0.0C _H
64	0.06 _H
128	0.03 _H
256	0.01 _H
512	0.01 _H
>512	0.00 _H

Tabla 1. Constante de factor de corrección del flujo instantáneo.

- II. Constantes para cálculo del flujo lineal utilizados en la fórmula [54.]

$$LOG2_ESC = \text{Logaritmo2}(409.6) = 8. AE_H \quad 61.$$

Escala	LOG2_10D = $\text{logaritmo2}(10^{Escala})_H$
0	0.00 _H
1	3.52 _H
2	6.A5 _H
3	9.F7 _H
4	D.4A _H
5	10.9C _H

Tabla 2. Constantes para cambio de escala.

- Constantes para cálculo del flujo logarítmico utilizados en la fórmula [58.]

$$CTE_PHILOGARITMICO = \text{logaritmo2}\left(\frac{2^{12}}{10 \cdot \text{logaritmo2}(10)}\right) = 6.F2_H$$

- III. Cálculo del Logaritmo Base 2

El cálculo de logaritmo base 2 se realiza mediante una tabla que da el resultado en 1 byte entero más 1 byte decimal. La entrada a la función es un valor entero al cual se le calcula la característica, luego se obtiene

logaritmo base 2 de los 9 bits más significativos y finalmente se suman ambos valores.

$$\text{Característica} = \text{floor}(\log_2(x)) \quad 62.$$

$$\text{logaritmo2} = \text{Característica} + \underbrace{\left[\log_2 \left(\frac{x}{2^{\text{Característica}}} \right) \right]_{0.8 \text{ bits}}}_{\text{Valor guardado en tabla en el hardware}} \quad 63.$$

El valor de salida de la función tiene 8 decimales, lo que equivale a un error máximo de -0.07% , decreciente con x .

IV. Cálculo del Antilogaritmo base 2

El cálculo del antilogaritmo recibe como entrada un número con 8 bits decimales. Se obtiene el valor de 2 elevado a la parte decimal del número. Luego se hace una multiplicación por 2^N , donde N es la parte entera del número. En la implementación esta multiplicación se hace mediante N desplazamientos de bits a la izquierda.

$$\text{Antilog}_2(x) = 2^x = 2^{(N + b_{-1} \cdot 2^{-1} + b_{-2} \cdot 2^{-2} + \dots + b_{-8} \cdot 2^{-8})} \quad 64.$$

$$\text{Antilog}_2(x) = 2^N \cdot \underbrace{2^{b_{-1} \cdot 2^{-1} + b_{-2} \cdot 2^{-2} + \dots + b_{-8} \cdot 2^{-8}}}_{\text{Valor guardado en tabla en el hardware}} \quad 65.$$

donde:

- N : parte entera de x ,
- $b_{-1} \cdot 2^{-1} + b_{-2} \cdot 2^{-2} + \dots + b_{-8} \cdot 2^{-8}$: parte decimal de x
- $b_i \in [0;1]$: dígitos binarios

El valor de salida de la función tiene 8 decimales, lo que equivale a un error máximo = 0.19% .

2.2.5 Generación de tablas para multiplicación en hardware

Para la determinación de los intervalos de decisión para la modificación de N , se calculan los coeficientes que multiplican al flujo a partir de la fórmula del desvío estándar, (34) y (43). Estos coeficientes se aproximan por valores que se pueden expresar con sólo 2 dígitos binarios, para poder hacer la multiplicación por la suma de 2 desplazamientos de bits del valor del flujo. El intervalo de confianza para con el que se evalúa bajar el ' N ' es:

$$NSIGMAS \cdot \sigma_{\varphi_{i,K}} = NSIGMAS \cdot \frac{\varphi_{i,K}}{\sqrt{K-2}} \approx \left(\frac{1}{2^a} + \frac{1}{2^b} \right) \varphi_{i,K} \quad 66.$$

Se genera el vector de coeficientes con índice ' j ' asociado a la expresión:
 $K = 16 \cdot 2^j$.

Para el caso del desvío estándar del promedio, se obtiene la fracción de sigma promedio que se utiliza en la decisión a partir del valor ' $NSIGMAS \cdot \sigma_{\varphi}$ ' obtenido antes, obtiene la fracción de sigma promedio que se utiliza en la

decisión. También se genera un vector de coeficientes aproximados por números de 2 dígitos binarios, con índice 'c' asociado a la expresión: $N = 2^c$.

2.3 Resultados de las simulaciones

Se simuló el procesamiento haciendo un barrido de los parámetros que luego serán fijos en la implementación en hardware. Éstos son :

- **NSIGMAS** : cuantos desvíos estándar se acepta que resulte el flujo instantáneo alejado del promedio sin bajar el promedio. Se analizan NSIGMAS = 2 y NSIGMAS = 3.
- **TACUM** : es el tiempo mínimo en que se acumulan pulsos. No puede ser mayor a 50ms. Se analizan TACUM = 5mS, TACUM = 10mS y T_ACUM = 20mS.
- **ERROR_INFERIORF**: es el máximo error que se acepta que tenga el flujo instantáneo, afecta directamente a la mínima cantidad de cuentas acumuladas. Se analizan los valores indicados en la Tabla 3.

ERROR_INFERIORF	K mínimo asociado
5%	512
10%	128
20%	32
50%	4

Tabla 3. Valores de ERROR_INFERIORF analizados.

Se deciden los parámetros en función de los valores obtenidos de la media de las 100 corridas de los siguientes indicadores:

- **Proporción de valores atípicos en el flujo promedio**: se calcula la cantidad de mediciones que resultan con un valor de flujo promedio que se desvía más del 5% del valor anterior obtenido. En los gráficos se muestra como una proporción de tiempo. Se calcula este parámetro como un estimador que evidencia un comportamiento no deseado del valor de flujo promedio que es la salida del módulo tanto en display como en forma analógica. Estos valores atípicos son muy visibles cuando se grafica el resultado, como se muestra en la Figura 7.
- **Valor cuadrático medio (RMS) del error % del flujo promedio**: es la raíz cuadrada del valor medio de los cuadrados del valor del error porcentual respecto al valor de simulación teórico. La media se calcula sobre las mediciones que corresponden al primer tramo del flujo teórico para $\phi(t) = 10^i$ hasta la medición anterior al salto.
- **Tiempo de refresco**: es el tiempo de refresco entre mediciones. Se calcula el valor medio en el tiempo que corresponde al primer tramo del perfil teórico hasta la medición anterior al salto
- **Retraso al 90%**: es el tiempo entre el instante real del salto y el instante que el flujo promedio llega por primera vez al 90% del valor final de salto.

Se descartan las primeras 20 mediciones, ya que en ellas los valores de los parámetros variables se están ajustando desde sus valores iniciales y los resultados en esos casos no son representativos de la capacidad de método.

El tiempo de simulación no es el mismo para todos los valores de flujo, debido a la limitación en el tamaño de la matriz de tiempos entre pulsos necesaria, mucho más grande cuanto mayor es el flujo. Sin embargo se aseguró un tiempo de simulación de al menos 100 valores de medición para hacer los promedios de tiempo de respuesta y RMS del error del flujo promedio.

A cada combinación de los parámetros NSIGMAS, TACUM y ERROR_INFERIORF se le asignó un número de caso según se muestra en la Tabla 4. El número de caso es el que aparece como eje de ordenadas en las figuras siguientes.

Número de caso	NSIGMAS	TACUM	ERROR_INFERIORF
1	2	5 ms	5 %
2			10%
3			20%
4			50%
5		10 ms	5 %
6			10%
7			20%
8			50%
9		20 ms	5 %
10			10%
11			20%
12			50%
13	3	5 ms	5 %
14			10%
15			20%
16			50%
17		10 ms	5 %
18			10%
19			20%
20			50%
21		20 ms	5 %
22			10%
23			20%
24			50%

Tabla 4. Listado de la combinación de parámetros simulado y su número de caso asociado.

En la Figura 7 se grafica el valor del primer indicador analizado. Se grafica el valor medio de la proporción de mediciones que resultaron más del 5% distintas a la medición anterior. En la Figura 6 se muestra un ejemplo para flujo igual a 1000c/s calculado según el caso 4.

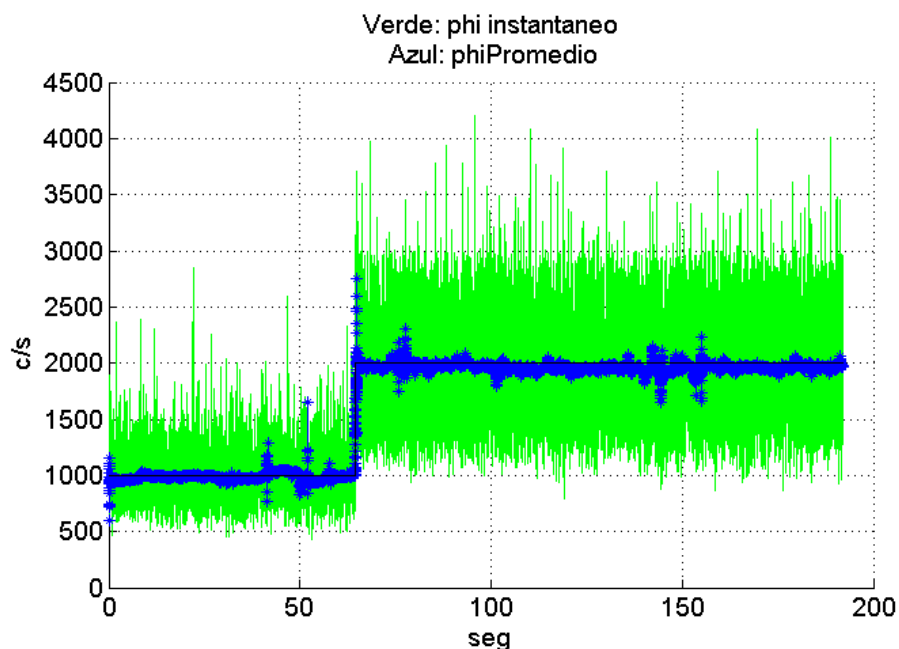


Figura 6. Ejemplo de una corrida para flujo = 1000c/s y NSIGMAS =2. Se ven los picos en el flujo promedio.

En la Figura 7 se observa que para los casos 1 a 12, en los que NSIGMA es igual a 2 se produce una mayor proporción de valores atípicos del flujo promedio, especialmente para flujos igual a 1000 c/s, 100c/s y 10c/s.

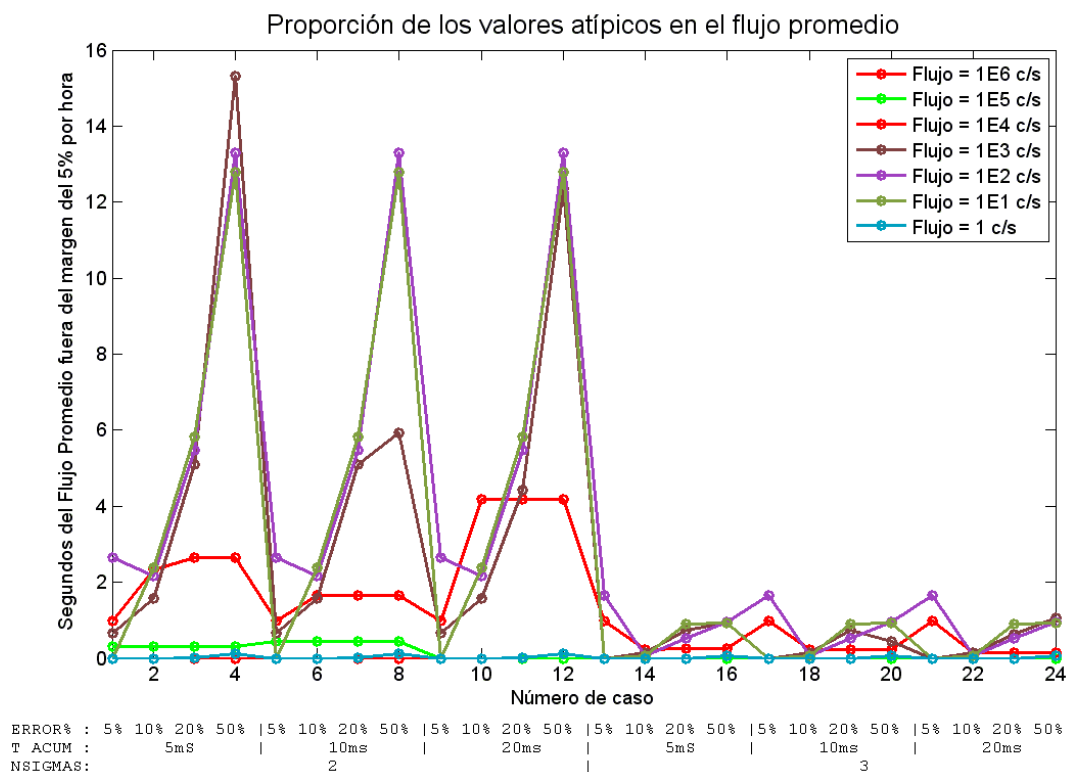


Figura 7. Resultados del indicador proporción de valores atípicos en el flujo promedio

Por esta razón se descarta usar el valor de parámetro de NSIGMA igual a 2.

Desde la Figura 8 a la Figura 14 se muestran los resultados obtenidos para los otros 3 indicadores: RMS del error, tiempo de refresco y retraso al 90%, para los casos de simulación 13 a 24, en los que NSIGMA es igual a 3. En la parte de abajo del gráfico se indica a qué combinación de los parámetros de medición corresponde cada punto.

En estos gráficos se hace evidente es el compromiso entre tiempo de respuesta y error, que es mayor para flujos menores a 10^4 c/s. Para flujos mayores el error depende del tiempo de acumulación. Se descarta usar TACUM = 5 ms ya que con tiempo de acumulación mayor se asegura RMS de error menor al 1% a partir de flujo 10^5 c/s y en los demás es menor o no afecta.

Se descarta ERROR_INFERIORF = 5% porque aumenta el tiempo de refresco a más del doble para flujos menores a 10^4 c/s, y en particular para ese valor de flujo asegura un tiempo de refresco de 50ms. Se descarta usar ERROR_INFERIORF = 50% porque aumenta en más de 3 veces el tiempo de refresco a para flujos iguales a 10c/s y 100c/s, y para los demás flujos no afecta.

Entre TACUM = 10 ms y TACUM 20ms, siendo que no afecta a las variables analizadas, se elige el menor tiempo de acumulación, TACUM = 10ms.

Finalmente entre los casos 18 y 19, se tiene el compromiso de menor tiempo de refresco o menor error. Se elige el caso 19 que da menor tiempo de refresco:

Caso : 19	NSIGMAS = 3	TACUM = 10ms	ERROR_INFERIORF = 20%
-----------	-------------	--------------	-----------------------

Tabla 5. Parámetros de cálculo elegidos.

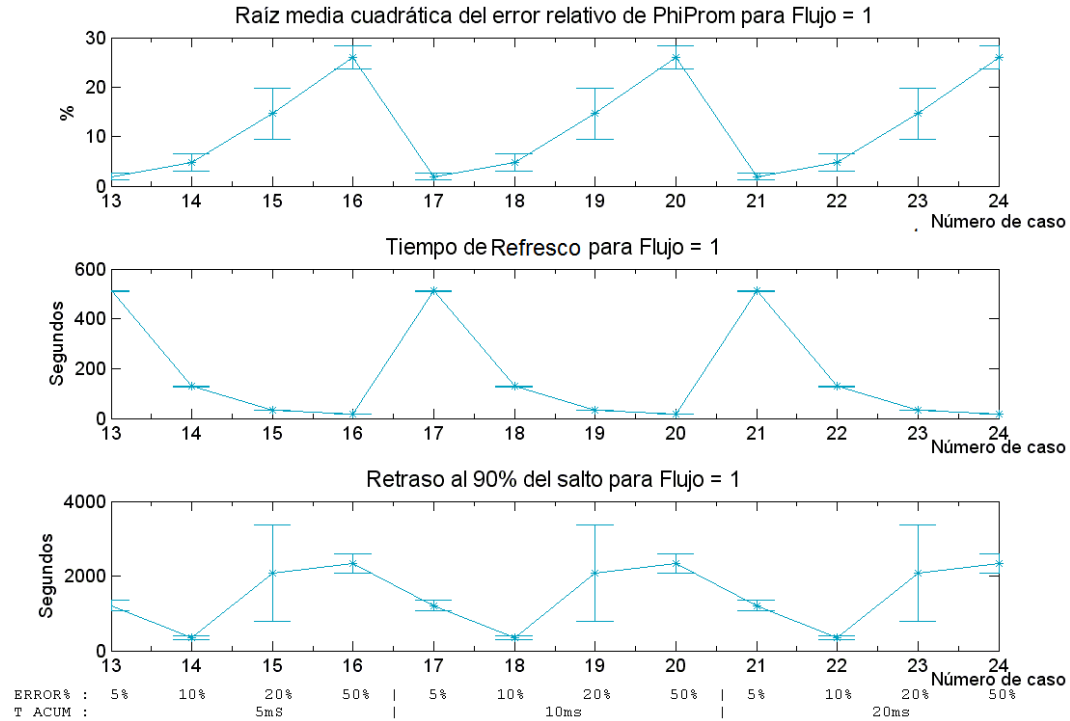


Figura 8. Promedio y desviación estándar de 100 corridas de las variables de decisión para flujo igual a 1 c/s de los casos de simulación 13 a 24.

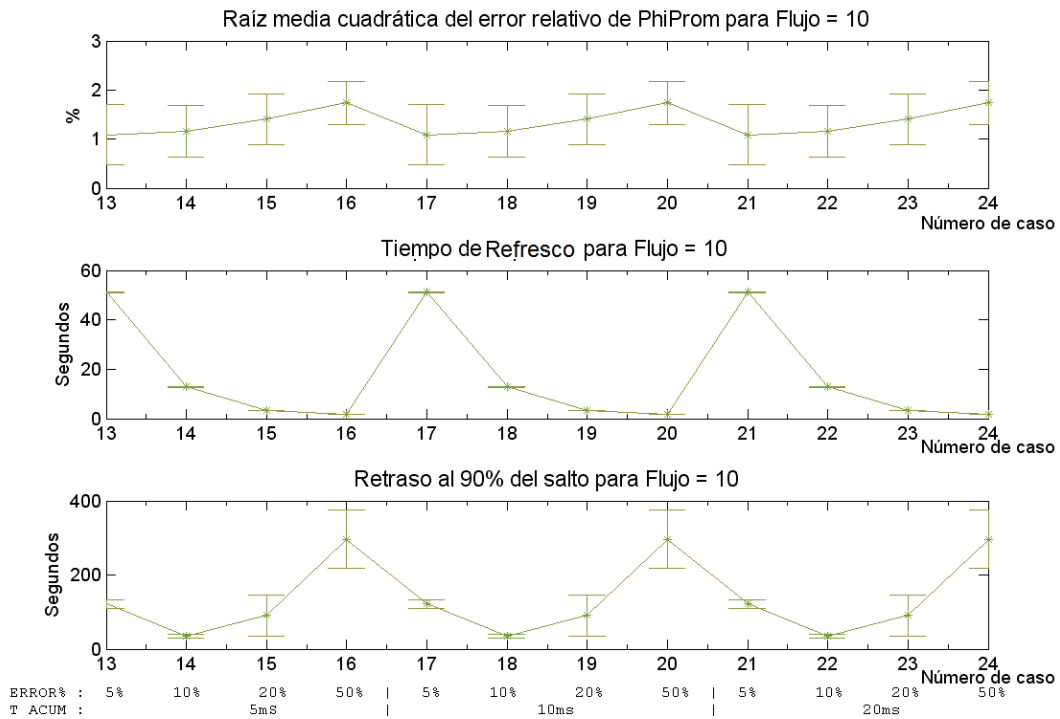


Figura 9. Promedio y desviación estándar de 100 corridas de las variables de decisión para flujo igual a 10^1 c/s de los casos de simulación 13 a 24.

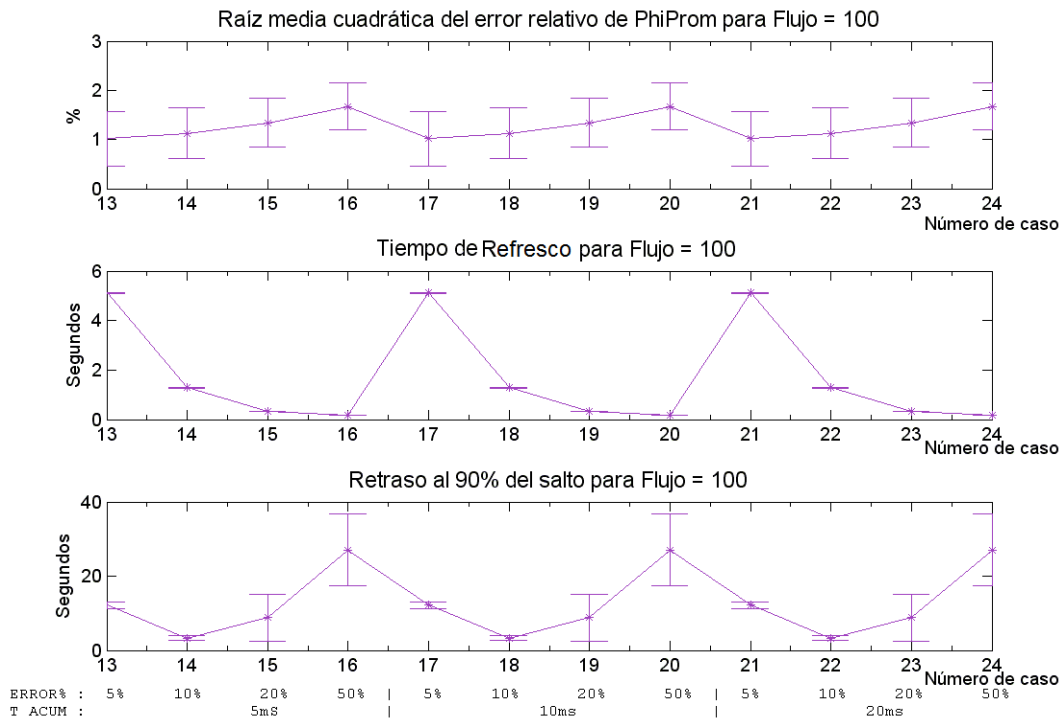


Figura 10. Promedio y desviación estándar de 100 corridas de las variables de decisión para flujo igual a 10^2 c/s de los casos de simulación 13 a 24.

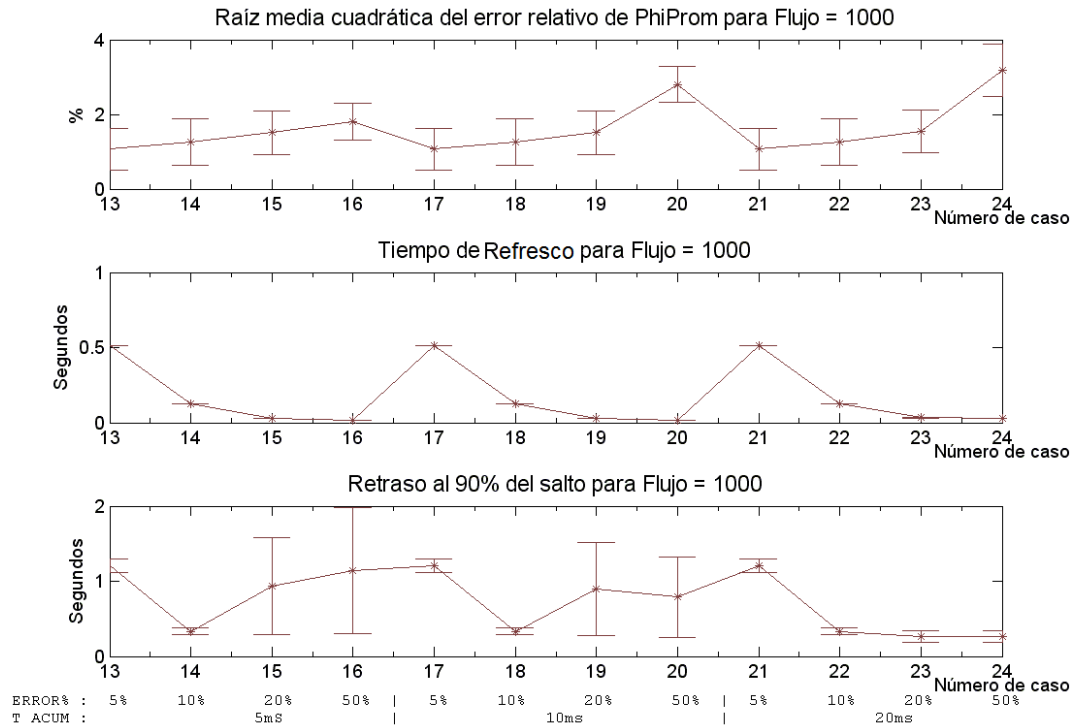


Figura 11. Promedio y desviación estándar de 100 corridas de las variables de decisión para flujo igual a 10^3 c/s de los casos de simulación 13 a 24.

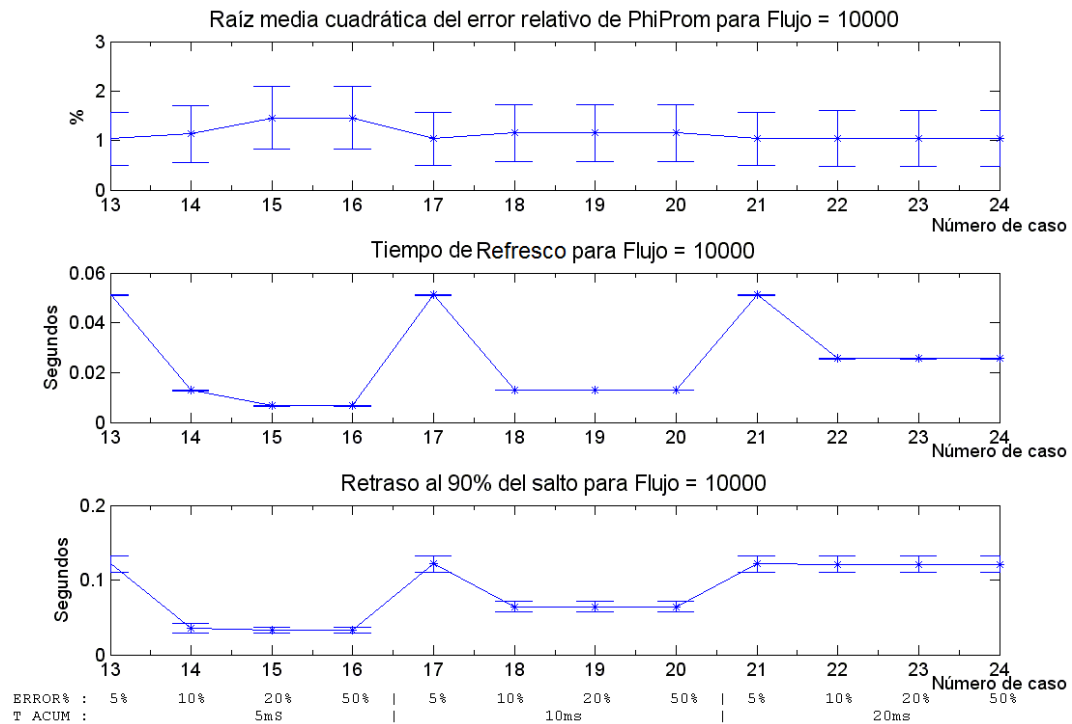


Figura 12. Promedio y desviación estándar de 100 corridas de las variables de decisión para flujo igual a 10^4 c/s de los casos de simulación 13 a 24.

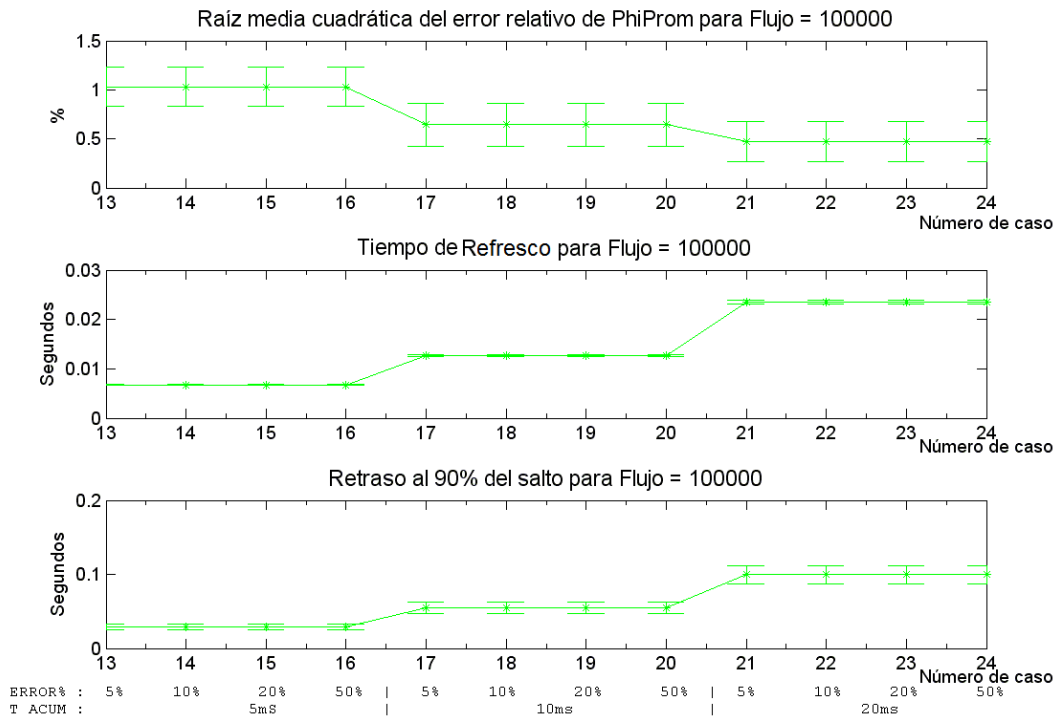


Figura 13. Promedio y desviación estándar de 100 corridas de las variables de decisión para flujo igual a 10^5 c/s de los casos de simulación 13 a 24.

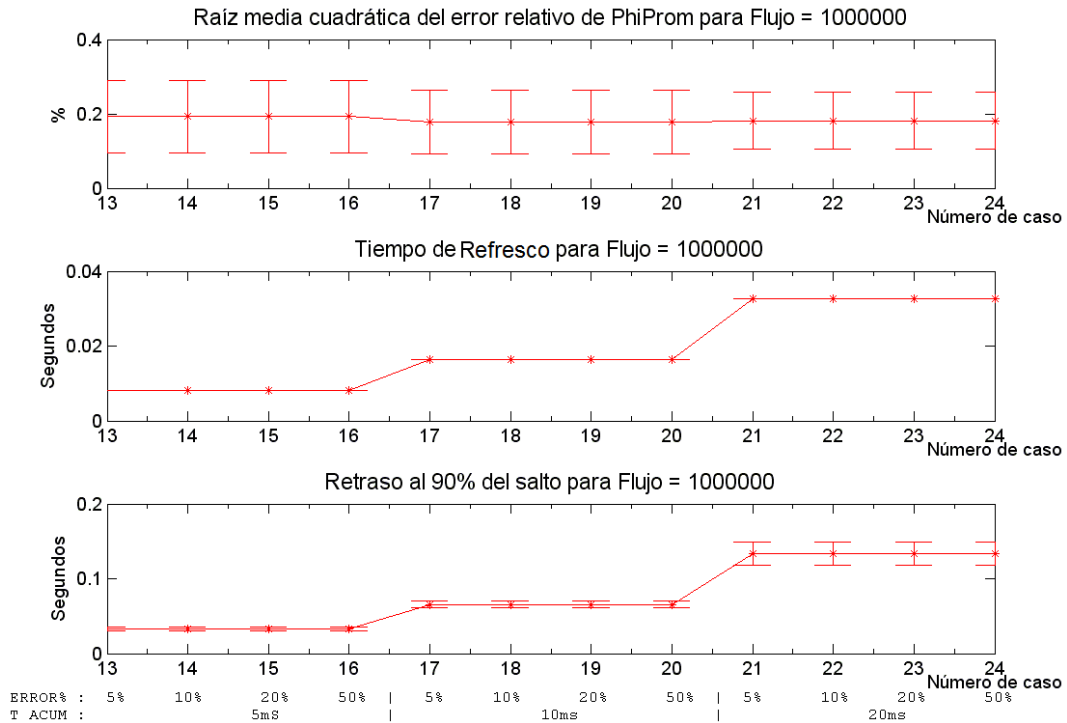


Figura 14. Promedio y desviación estándar de 100 corridas de las variables de decisión para flujo igual a 10^6 c/s de los casos de simulación 13 a 24.

En la Figura 15 se grafica la salida del cálculo utilizando los parámetros fijos elegidos. El flujo promedio se grafica superpuesto al flujo instantáneo, junto al valor de los parámetros variables K y c , que da el parámetros del filtro 'N'. También se grafica el tiempo de refresco.

Se observa como varían los valores de N y K punto a punto debido a la dispersión del flujo instantáneo, graficado en verde. En el momento del salto, el valor de c baja hasta 2, lo que permite seguir al flujo. El tiempo de refresco se mantiene acorde al valor que corresponde a cada valor de flujo.

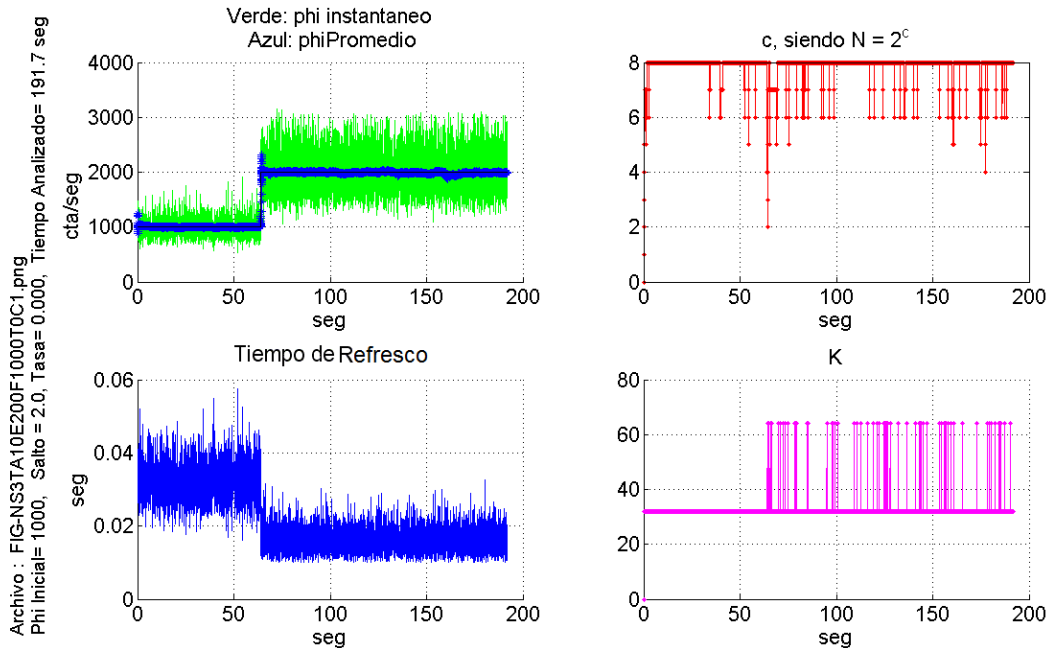


Figura 15. Ejemplo de resultado del cálculo y valores de los parámetros K y c para flujo igual a 10^3 c/s.

En la Figura 16 se grafica el valor de tiempo de refresco que es esperable para el caso elegido, en función del valor de flujo.

A partir de flujo igual a 1000c/s resulta menor a 50ms, con muy baja variabilidad como se observa en las Figura 11 a Figura 14.

En la Figura 17 se grafica el valor de RMS del error del flujo promedio que es esperable para el caso elegido, en función del valor de flujo. El valor está alrededor del 1.5% para flujos mayores a 10 c/s, y menor al 1% para valores mayores, aunque como se verá en el Capítulo 5, el error debido al ancho de pulso afectará notablemente en la última década.

En la Figura 18 se grafica el valor de tiempo de retraso en llegar al 90% del valor final para un salto por 2 del flujo, que es esperable para el caso elegido, en función del valor de flujo. A partir de flujo igual a 10^3 c/s el tiempo es menor a 1 segundo.

Los valores mostrados en estas gráficas se utilizarán para comparar los valores medidos en el proceso de verificación en el Capítulo 5.

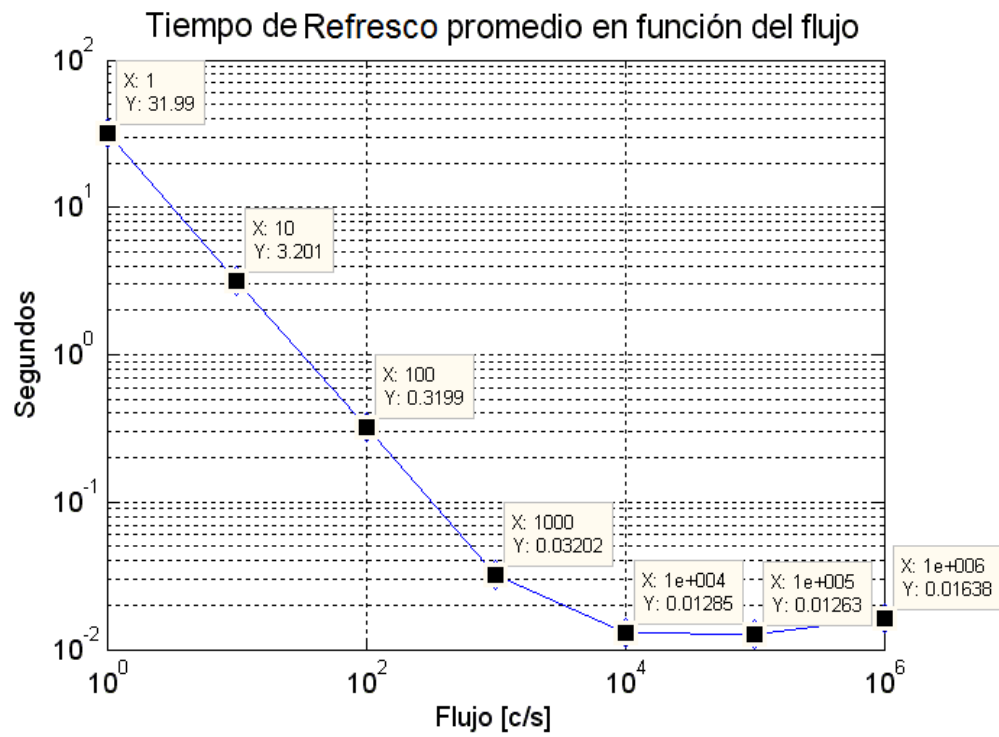


Figura 16. Valor medio de 100 corridas del tiempo de refresco en función del flujo.

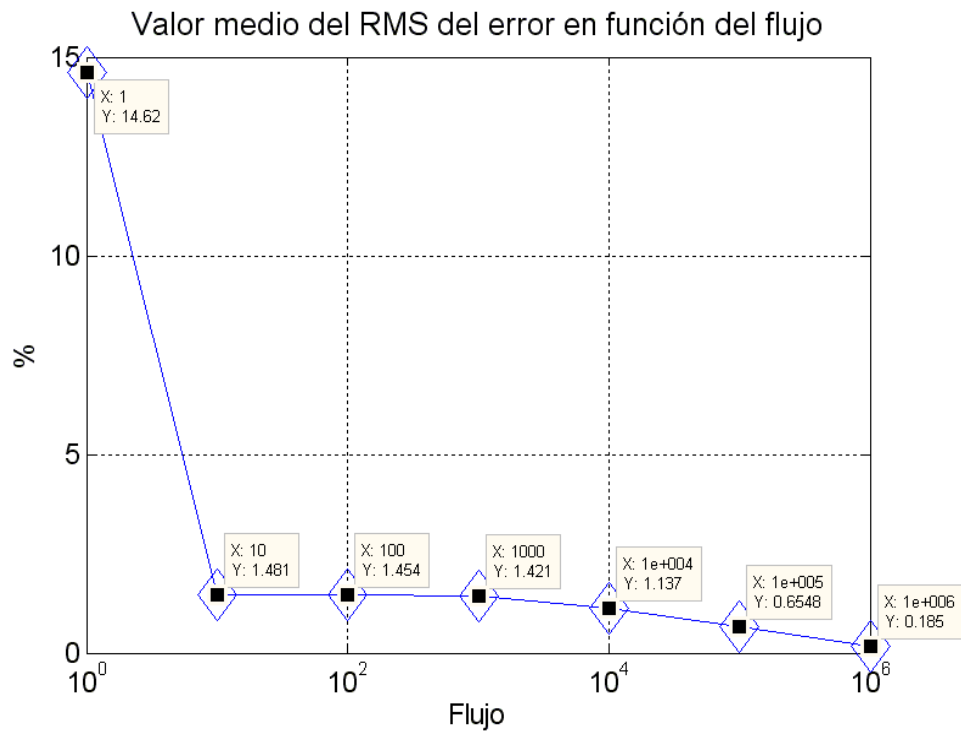


Figura 17. Valor medio de 100 corridas del RMS del error del flujo promedio en función del flujo.

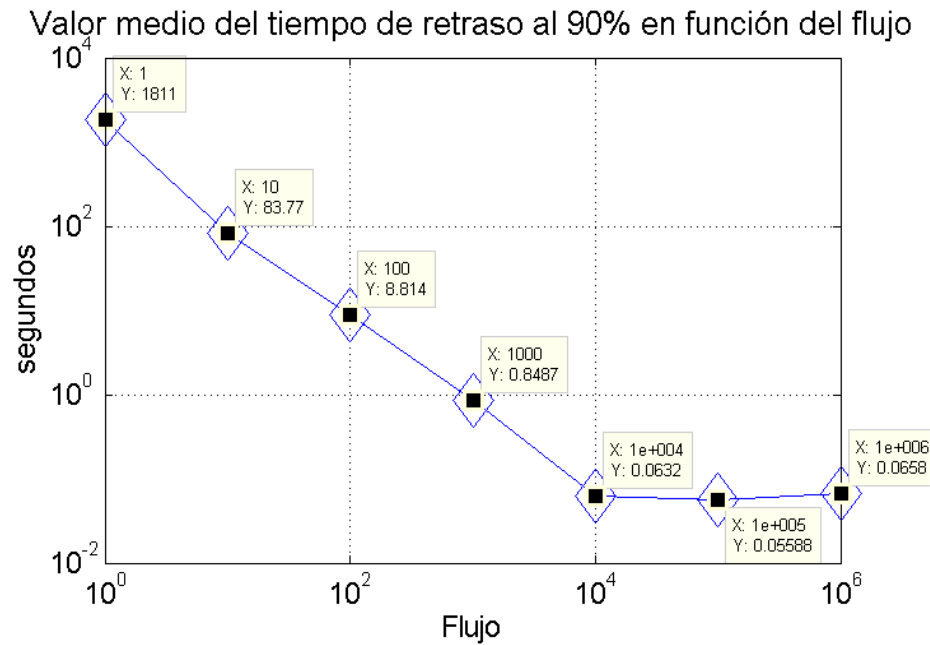


Figura 18. Valor medio de 100 corridas del tiempo de retraso al 90% para salto por 2 en función del flujo.

2.3.1 Verificación del cálculo para tasas distintas de cero.

Se verificó que el resultado del cálculo para tasas de variación distintas de cero se mantenga con errores dentro de los observados para flujo constante, lo que indica que no se retrasa en seguir un flujo permanentemente variable.

En la Figura 19 se observa el resultado del cálculo con flujo variable con tasa de 7% desde flujo igual a 10 c/s, y en la Figura 20, para tasa de 3% desde flujo igual a 10^3 c/s. Se ve como el valor de K se va adaptando, subiendo a medida que el flujo sube, ya que tarda menos de TACUM=10ms en acumular las cuentas indicadas.

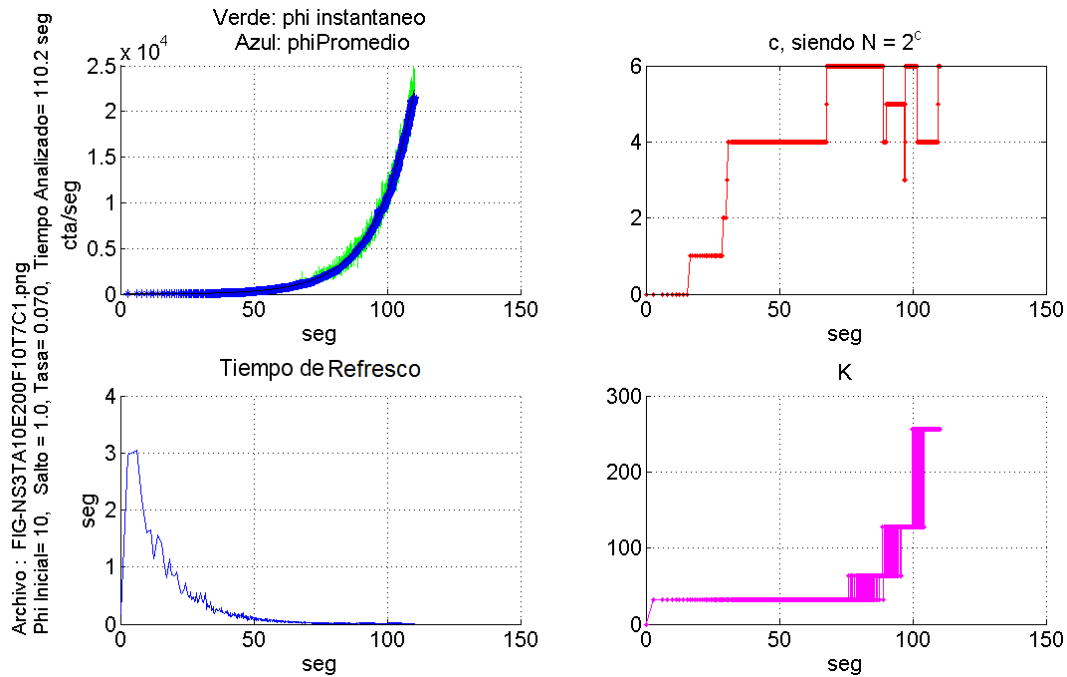


Figura 19. Simulación con tasa de variación de 7% desde flujo inicial =10 c/seg.

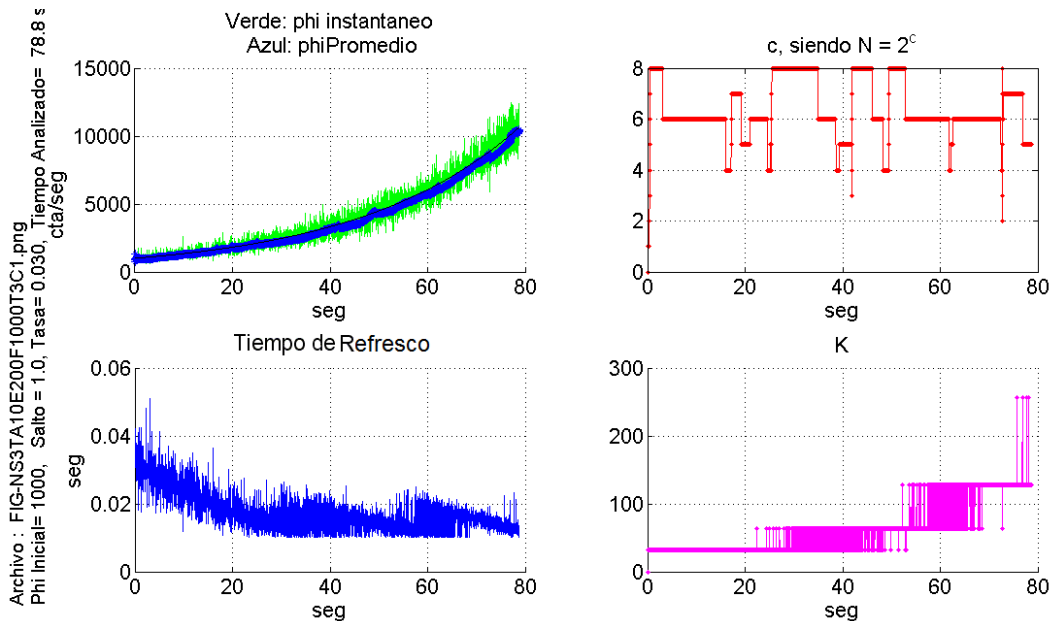


Figura 20. Simulación con tasa de variación de 3% desde flujo inicial =1000 c/seg.

Para obtener un valor representativo del error del flujo para tasas de variación 3% y 7%, y se hicieron 100 corridas de excursiones de flujo desde 10c/s hasta 10^6 c/s y calculó el error del flujo promedio respecto al teórico correspondiente a cada punto, y se promediaron 10 tramos por década. Se hizo así porque el punto en la curva teórica se toma para cada medición varía con la aleatoriedad de la suma de tiempos entre pulsos. Tomando tramos se caracteriza por sectores al error, como se observa en la Figura 21, donde se grafica el valor medio de las 100 corridas en cada tramo.

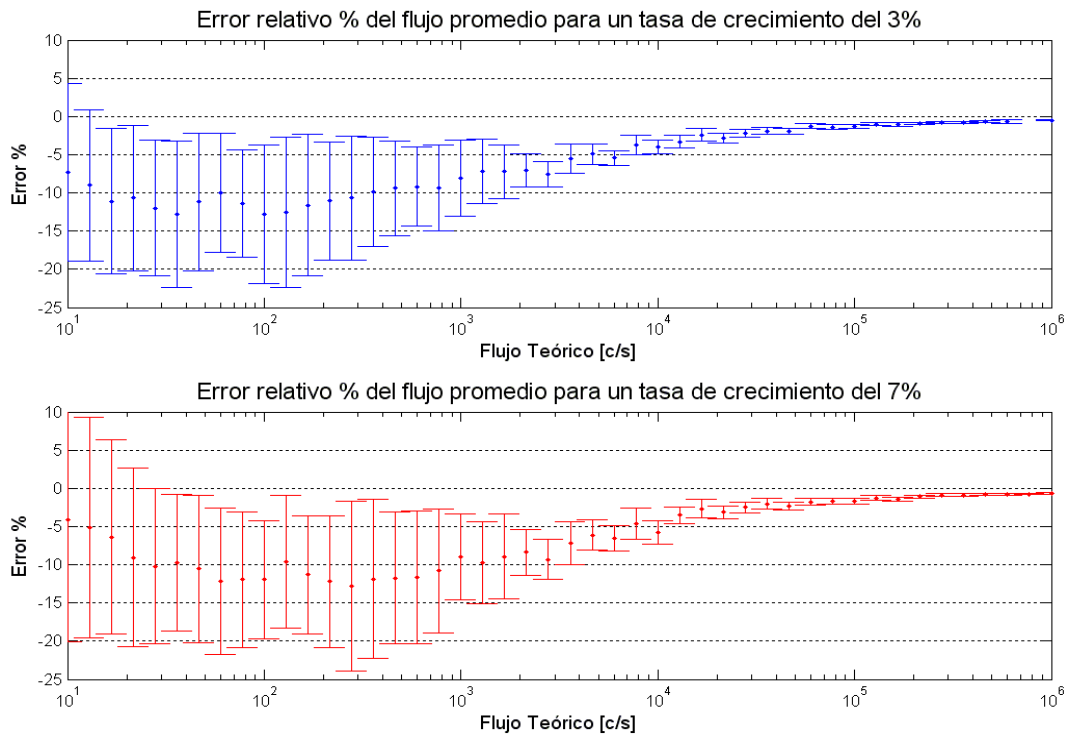


Figura 21. Media del error del flujo promedio para tasas de variación de 3% y 7%.

Para flujos bajos el error para tasa de variación 3% y 7% se mantienen iguales, estando debajo del 15%, con un desvío mayor a $\pm 5\%$.

Para tasa de variación de 3%, a partir de flujo igual a 400c/s el error es menor al 10%, y la dispersión importante pero siempre decreciente.

Aún para tasas de variación altas, el flujo promedio sigue al teórico con errores menores al 15% para flujos bajos de 1c/s, y errores menores al 5% para flujos altos.

2.3.2 Análisis de salto negativo de flujo o ausencia de pulsos

En casos en que la falta de pulsos de entrada produzca un overflow de la cuenta de tiempo, o que una disminución muy abrupta del flujo produzca un retardo en la respuesta del equipo que resulta excesivamente larga, se espera que los tiempos de refresco sean del orden que corresponda al flujo existente, o detectar la condición de falta de pulsos e indicarla en el equipo.

En la Figura 22 se ve el caso de un salto de flujo = 10^5 c/s a flujo = 100 c/s. En este caso acumular K cuentas sin controlar el tiempo que esto toma, implica un tiempo de refresco varias veces mayor al esperado para el flujo = 100 c/s.

Se implementa un control de tiempo que monitorea que:

$$\text{Tacumulado}_{\text{actual}} < 8 \cdot \text{Tacumulado}_{\text{anterior}} \quad 67.$$

Si no se cumple, toma el Tacumulado para ya guardado de $K/2$. En el caso que K sea igual al mínimo, no es posible tomar $K/2$. Para esos casos que se

dan cuando el flujo es muy bajo, se indicará **BajoRango** ya que se debería bajar el K, pero no es posible.

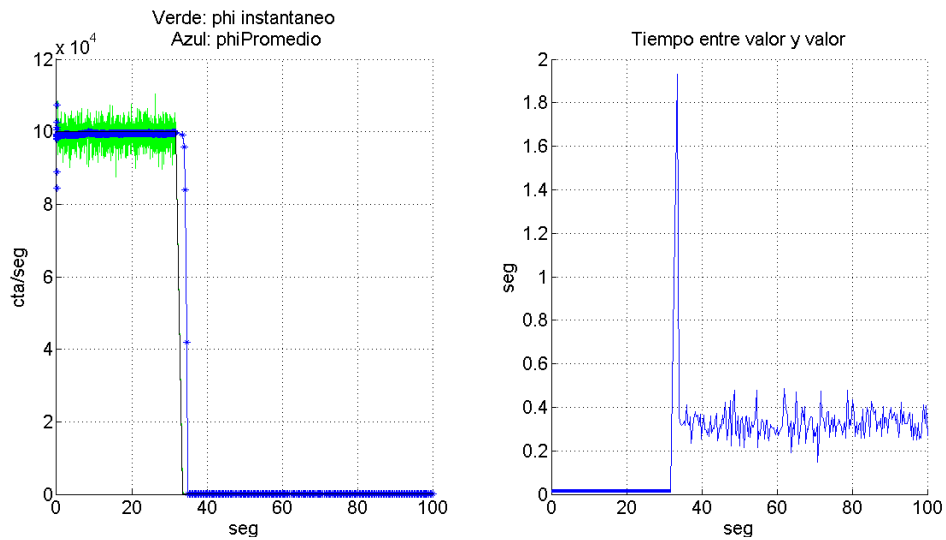


Figura 22. Caso sin control del tiempo. Tiempo de refresco varias veces mayor al normal.

Además se incluye el control de overflow del tiempo acumulado. El contador de tiempo acumulado debe detenerse al llegar a la máxima cuenta, y el cálculo se hará con ese valor. Se indicará **BajoRango** ya que no es posible medir correctamente el valor de flujo, según se indica en la Figura 34.

Si se controla el tiempo, se tiene el siguiente resultado para el mismo caso anterior:

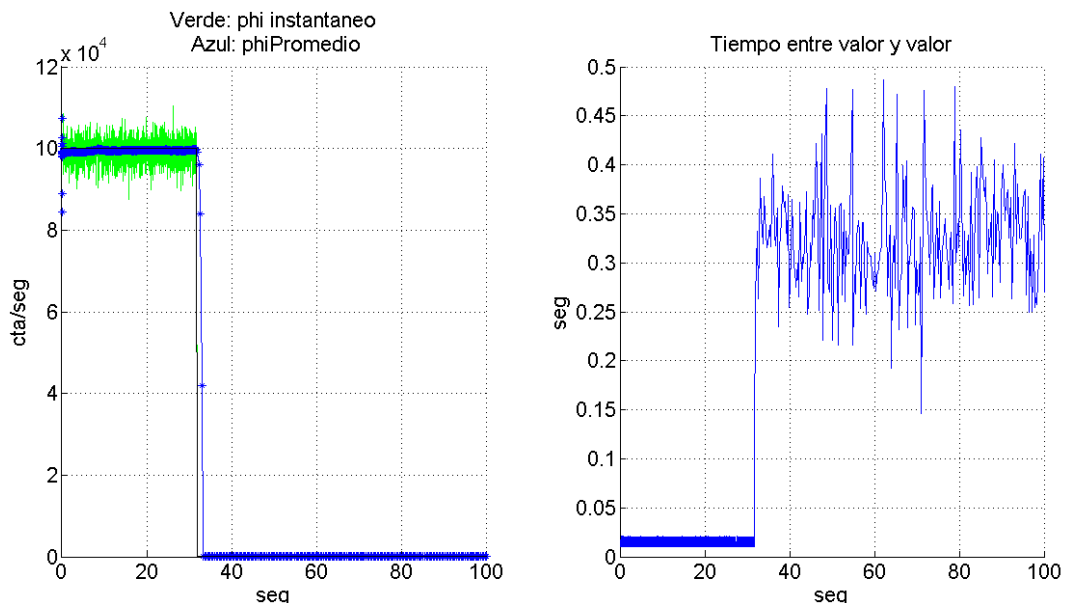


Figura 23. Caso con control de tiempo. Tiempo de refresco del orden del correspondiente al flujo presente.

3. Capítulo 3

3.1 Descripción del equipo

El equipo consiste en un módulo eurocard de 160x100mm con conector eurocard tipo C de 64 pines. El frente del módulo es de 128mm de alto y 4U (=20.3mm) de ancho.

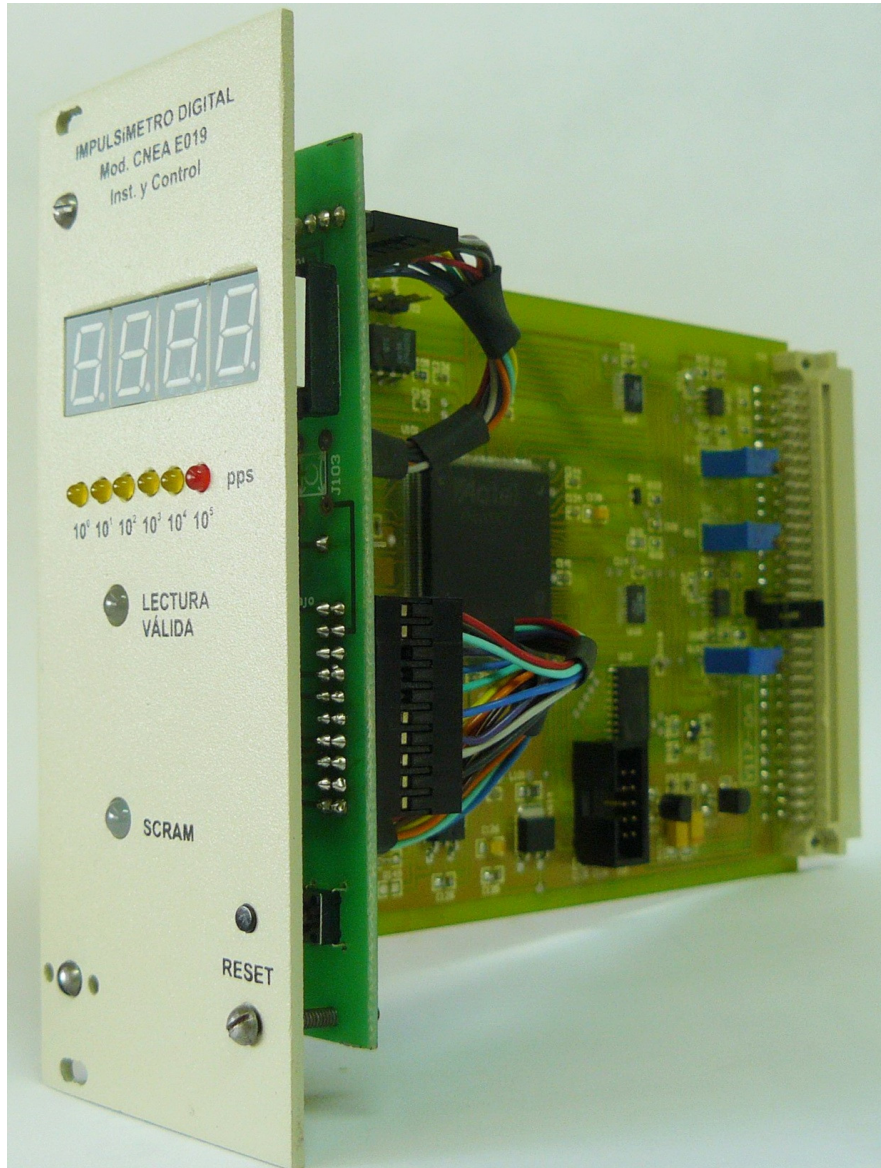


Figura 24. Foto del módulo impulsímetro.

En la Figura 25 se muestra el diagrama en bloques del circuito.

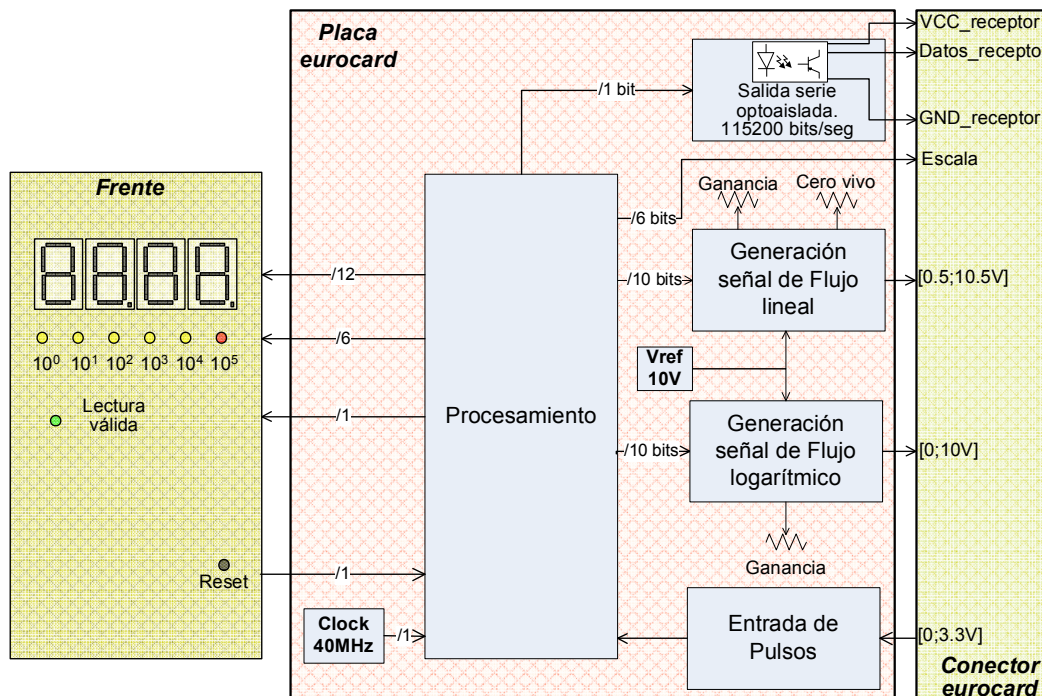


Figura 25. Diagrama en bloques del circuito eléctrico.

3.1.1 Procesamiento: FPGA

El procesamiento, que consiste en hacer los cálculos descritos en el capítulo 2 y el control de los indicadores del frente, lo realiza una FPGA: “Arreglo lógico configurable”. Este dispositivo consiste en un array de celdas básicas, con recursos típicos como tablas lógicas configurables de, típicamente, 3 ó 4 entradas; y 1 ó 2 flip-flops D. A cada celda llega una línea de clock. Las celdas se repiten miles de veces y cada una está interconectada con muchas otras celdas. Con estos recursos es posible realizar las funciones lógicas necesarias, y al tener elementos de memoria, hacer lógica secuencial.

Se programa utilizando lenguajes de descripción de hardware, y en particular el utilizado en este desarrollo es el VHDL. En la descripción se crean componentes, describiendo sus entradas y salidas y la función lógica que realizan. Luego se instancian todas las veces necesarias y se especifican las interconexiones entre ellos, en forma similar a cuando se dibuja un circuito eléctrico, pero utilizando texto. Permite describir funciones lógicas combinacionales y secuenciales.

Una propiedad interesante de estos dispositivos es que permiten el procesamiento paralelo, debido a que no existe un programa ejecutándose, sino que cada componente descrito realiza su función independientemente de cualquier componente que no esté conectado con él. El grado de paralelismo es muy grande, sólo limitado por la cantidad de tablas y flip-flops (elementos de memoria) disponibles.

Otra propiedad que los hace muy versátiles en comparación con un microcontrolador, es que se pueden instanciar la cantidad de componentes que se necesiten. Por ejemplo si se requiere recibir datos de ‘N’ líneas serie de

recepción, se instancian 'N' UARTs (Universal Asynchronous Receiver /Transmitter) que trabajan en paralelo. En un microcontrolador este recurso está definido de fábrica, limitando las aplicaciones o haciéndolas más complejas para poder compartir los recursos. Otro ejemplo de versatilidad se aprecia cuando se comparan los contadores disponibles en un microcontrolador que además tienen un tamaño máximo. En una FPGA se crean todos los contadores necesarios y de la cantidad de bits necesarios, sin desaprovechar recursos lógicos. Estas características de paralelismo y versatilidad resultan en mayores velocidades de funcionamiento y capacidades de procesamiento.

Los componentes son fácilmente reutilizables en otras implementaciones, cuando se utilizan descripciones estructuradas y jerárquicas.

El lenguaje VHDL permite también escribir tests de simulación. Es posible introducir vectores de estímulo temporizados a las entradas del sistema a simular. La salida se puede comparar con la salida esperada del sistema, pudiéndose hacer un reporte de las diferencias o detener la simulación ante un valor erróneo.

Los distintos fabricantes de FPGAs ofrecen sus ambientes de desarrollo, con una cantidad importante de facilidades gratuitas, incluyendo el simulador.

Para este equipo se eligió una FPGA de la familia ProAsic3 de la firma ACTEL, que se caracteriza por fabricar dispositivos de tecnología flash. Las características del producto que llevaron a su elección son las siguientes:

- *Fuente de alimentación única*, esto permitió desarrollar la placa de circuito impreso en sólo 2 capas, bajando el costo, la complejidad del circuito impreso y la cantidad de componentes.
- *Un solo chip*, la configuración está guardada en el mismo dispositivo, por lo tanto no hay un tiempo de carga de la configuración típico de las FPGA basadas en tecnología RAM. También reduce la cantidad de componentes necesarios en el circuito.
- *'Live at power up'*, significa que el dispositivo está funcional antes que la alimentación llegue a la tensión regulada requerida. Una vez pasado el nivel de activación que es aproximadamente la mitad de la tensión de alimentación, los pines de salida están activos.
- *'Neutron immune'*, esta característica se refiere a que la tecnología flash es más inmune a los efectos de partículas cargadas y neutrones de alta energía que pudiesen colisionar con el dispositivo. El fabricante se especializa en desarrollar integrados para uso aeroespacial y militar en las que los equipos están más expuestos a estas radiaciones.

LA FPGA elegida es la A3P1000 que tiene el equivalente a 24576 D-Flip-Flops, 144Kbits de memoria RAM, 1Kbit de ROM, 1 PLL integrado y en 300 pines de entrada/salida. El encapsulado de la FPGA es PQ-208, siendo el único que no es BGA (Ball Grid Array) con esa cantidad de recursos. La separación entre pines es de 0.5mm.

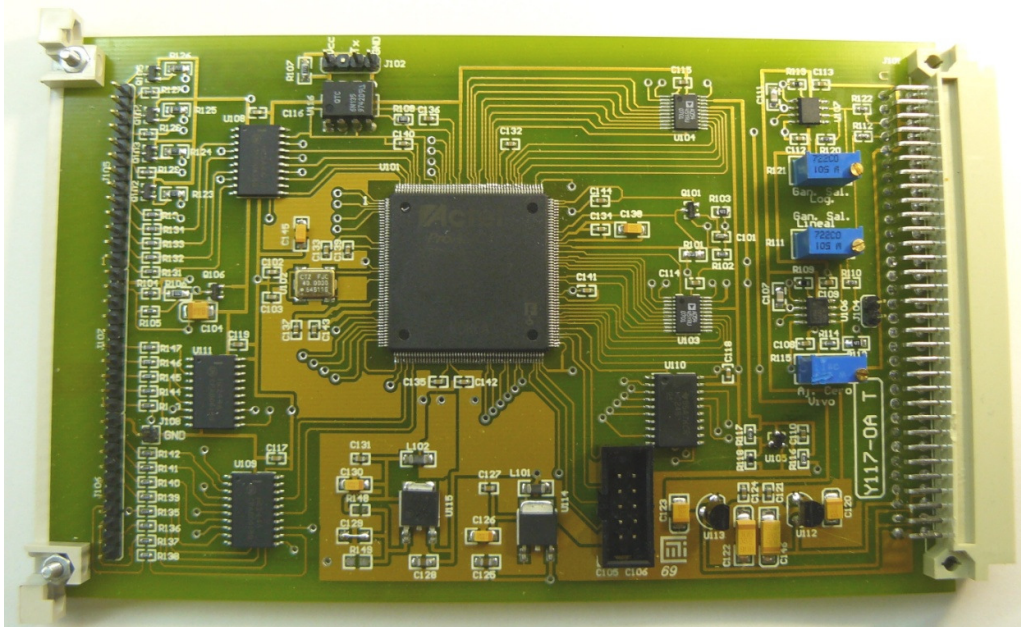


Figura 26. Fotografía de la placa eurocard. El integrado central es la FPGA.

3.1.2 Alimentación

El módulo requiere alimentación de 9V para generar las tensiones que necesitan los componentes digitales, y $\pm 15V$ para generar las tensiones que requieren los amplificadores de salida y los DACs.

3.1.3 Generación de clock

Se utiliza un circuito generador de clock, que genera una señal de 40MHz y es la entrada a un PLL interno de la FPGA. El PLL genera una señal de 'LOCK' que se utiliza como reset interno, el cálculo se inicia cuando la señal de clock es estable. El nivel de los pines de salida es asegurado a partir de que la alimentación alcanza los 0.85V y se mantienen mientras la señal de LOCK mantiene al dispositivo en RESET.

3.1.4 Entrada de pulsos

La señal de pulsos de entrada debe ser tener nivel LVTTTL, siendo 3.3V equivalente a un "1" lógico. En la Figura 27 se muestra la parte correspondiente del circuito. La impedancia de entrada es de 1K Ω . El ancho de pulso mínimo es de 40ns. Se cuenta en el flanco positivo del pulso.

Figura 27. Circuito de entrada de pulsos,

3.1.5 Salidas analógicas proporcionales al flujo lineal y logarítmico

Las salidas de flujo lineal y flujo logaritmo son generadas por sendos DACs de 12 bits. Se utilizó el integrado AD5445 que permite una referencia de 10V y un ancho de banda de salida 10Mhz, que es mucho mayor que el ancho de banda requerido. Los niveles lógicos son de 3.3V, pudiéndose conectar directamente a la FPGA.

Figura 28. Circuito de salida del flujo lineal.

La salida de escala asociada a la salida de flujo lineal está disponible en el conector eurocard. Las mismas son señales de 3.3V. En caso de ser necesario, se puede reconfigurar la FPGA para que acepte la escala desde el exterior, que las señales pasan por un buffer bidireccional controlado por la FPGA.

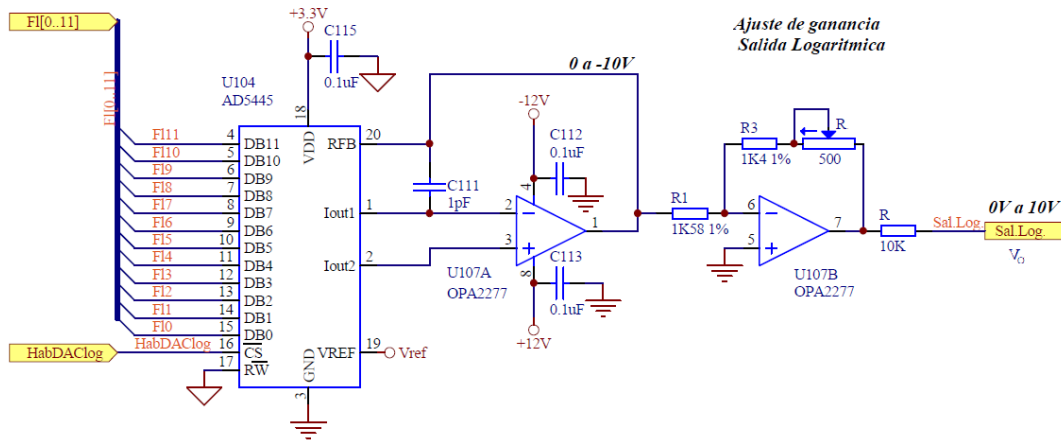


Figura 29. Circuito de salida del flujo logarítmico.

3.1.6 Display

El display de 4 dígitos 7 segmentos se controla desde la FPGA donde se implementó un componente que multiplexa la escritura de cada dígito, otro que hace la conversión a BCD y otro componente que convierte de BCD a 7 segmentos. Es un buen ejemplo de la versatilidad de la FPGA, se ahorra el espacio, y la compra, de 3 circuitos integrados.

En la Figura 30 se muestra la parte del circuito eléctrico que controla los displays. Éstos se encuentran en la placa de circuito impreso del frente. Los displays son cátodo común, cada uno se conecta a 4 transistores que hacen de habilitación. Se utiliza también el punto en 2 de los dígitos (señal Dot-Seg).

El ritmo de multiplexado es de 1ms, es decir que cada segmento se prende cada 4 ms durante 1ms.

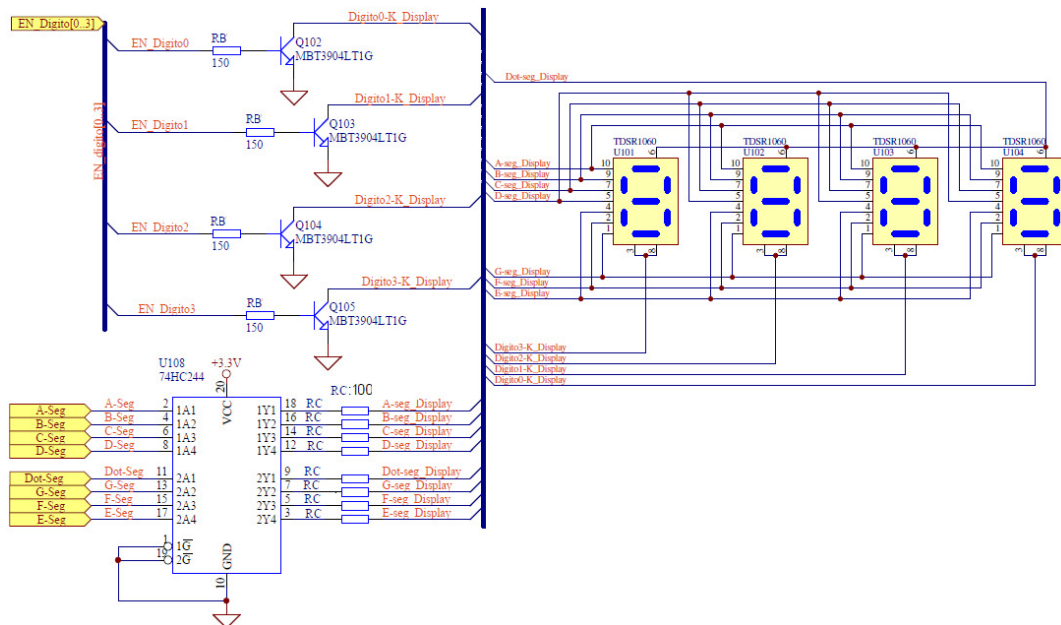


Figura 30. Circuito de manejo de display.

3.1.7 Salida de datos serie

El equipo provee una salida de datos optoacoplada por la que se transmite en forma serializada el estado interno de las variables más importantes del cálculo. La velocidad de transmisión es de 115200 bits/seg. Como se puede ver en la Figura 31, el equipo que lea esta señal debe proveer la alimentación, que determinará el nivel de tensión de la señal de salida y una línea de masa, que estará aislada de la masa interna. La señal Tx_out podrá ser de hasta 15V y la alimentación de 15V máximo. El valor de la resistencia R107 se determina en función del nivel de tensión requerida.

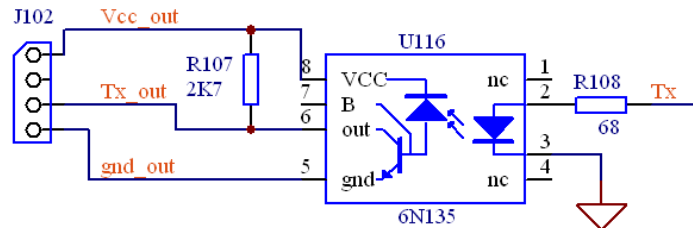


Figura 31. Circuito de salida de datos serie optoacoplada.

La transmisión se inicia luego de finalizada cada medición automáticamente sin que la FPGA reciba ningún tipo de comando desde el exterior. El mínimo tiempo entre mediciones es de 10ms, por lo que se pueden llegar a transmitir hasta 115 bytes por cada una.

En la fase de verificación del código VHDL se utilizó esta salida para adquirir los resultados, graficar las variables de medición y determinar el valor final de error y tiempo de respuesta del equipo.

3.1.8 Frente

En la Figura 32 se describe la funcionalidad de cada indicador en el frente del módulo. Los componentes se montan sobre una placa de circuito impreso del tamaño del frente, que se atornilla al frente de aluminio.

La indicación de 'overflow', se hace prendiendo los segmentos superiores de los displays. El overflow se produce por medirse un flujo mayor a $2.097 \cdot 10^6$ c/s o si se quiere representar un número mayor a 99.99 en el display.

La indicación de 'underflow', en caso de medirse flujo menor a 1c/s, se hace prendiendo los segmentos inferiores de los displays. La limitación está en la tabla de logaritmos que devuelve como mínimo el logaritmo de 1.

En el conector eurocard se tienen disponibles las señales listadas en la Tabla 6.



Figura 32. Indicadores del frente del módulo.

Señal	Pin	Descripción
Loop	c2,a2	Lazo de inserción del módulo
Sal. Log.	c3	Salida de flujo logarítmico
AGND	a3, c5	Masa analógica
Sal.Lineal	c4	Salida de flujo lineal
Tx_out	a10	Salida de datos serie optoaislada
gnd_out	a11	Masa aislada del transmisión de datos serie.
Vcc_out	a12	Alimentación externa para transmisión de datos serie
GND	c12	Masa digital
Entrada	c13	Entrada de pulsos
Reset	c14	Entrada de reset externo TTL 3.3V, activo alto
Scram	c16	Salida opcional no utilizada
Sal_Escala5	c19	Bit indicador de escala por 10^5 , activo alto
Sal_Escala4	c20	Bit indicador de escala por 10^4 , activo alto
Sal_Escala3	c21	Bit indicador de escala por 10^3 , activo alto
Sal_Escala2	c22	Bit indicador de escala por 10^2 , activo alto
Sal_Escala1	c23	Bit indicador de escala por 10^1 , activo alto
Sal_Escala0	c24	Bit indicador de escala por 10^0 , activo alto
+8.5V	c26,a26	Entrada de alimentación de +8.5V
+24V	c29,c30	Entrada de alimentación de +24V
-24V	a29,a30	Entrada de alimentación de -24V
GND	c32	Masa digital

Tabla 6. Lista de pines de entrada/salida del conector eruocard del módulo.

4. Capítulo 4

4.1 Implementación en VHDL

El cálculo presentado en el Capítulo 2 se transcribió en código VHDL. Para el desarrollo del código se usó el entorno de desarrollo 'Liberio 9' que provee el fabricante de FPGAs ACTEL en forma gratuita. Estos entornos proveen todas las herramientas necesarias para escribir el código, simularlo, sintetizarlo, hacer la implementación interna y la asignación de recursos; y finalmente grabar el dispositivo, facilitando el intercambio entre las distintas aplicaciones.

4.1.1 Diagrama en bloques

En la Figura 33 se ve el diagrama en bloques principal del código. Los componentes están descritos en VHDL, formando una estructura jerárquica. Cada componente puede ser instanciado varias veces, como es el caso del 'COUNTER_NBITSE'. En un caso se implementa un contador de 24 bits para llevar el tiempo total acumulado; y en otro, 17 bits, para contar la cantidad de pulsos detectados.

Al ser este el nivel superior de la estructura, las señales de entrada y salida indicadas en verde, son los pines de salida físicos del dispositivo.

Los bloques en amarillo, el PLL y el bloque de memoria RAM son recursos de hardware que no utilizan celdas lógicas. El PLL recibe la señal del generador del clock externo 'ClkIN' de 40MHZ-50% de duty cycle, y genera una señal de clock 'Clk' que se distribuye por todo el dispositivo por recursos de interconexión de clock globales. La señal 'LOCK' se mantiene en '1' mientras no se enclave el clock interno en fase y frecuencia con el clock externo, y activa la señal de 'ResetInterno', junto con el reset externo generado por un pulsador en el frente del equipo.

Una instancia del 'COUNTER_NBITSE' se utiliza para contar los pulsos externos. La señal de pulsos entra en el componente 'BORRA_FLANCO', que genera un pulso único de ancho igual al período de 'clk', cada vez que detecta un flanco positivo en su entrada. Este pulso habilita durante un único ciclo de clock el enable del contador, aumentando la cuenta en '1'.

El componente 'SETEABLE_CLOCKE' es un divisor de frecuencia de clock, escrito con parámetros genéricos de la frecuencia de entrada y de salida deseada, y se las especifica con constantes cada vez que se instancia uno de estos componentes, que son varias dentro de 'DISPLAY_DRIVER' para hacer el barrido de los enables de los 4 displays de salida y dentro de 'UART_TX' para fijar la velocidad de transmisión. En el caso mostrado en el diagrama en bloques se obtiene un clock de 125KHz, que produce un pulso de 25ns cada 8us.

La otra instancia del componente 'COUNTER_NBITSE' cuenta pulsos de la base de tiempo y genera la variable 'Ti' que corresponde al tiempo acumulado.

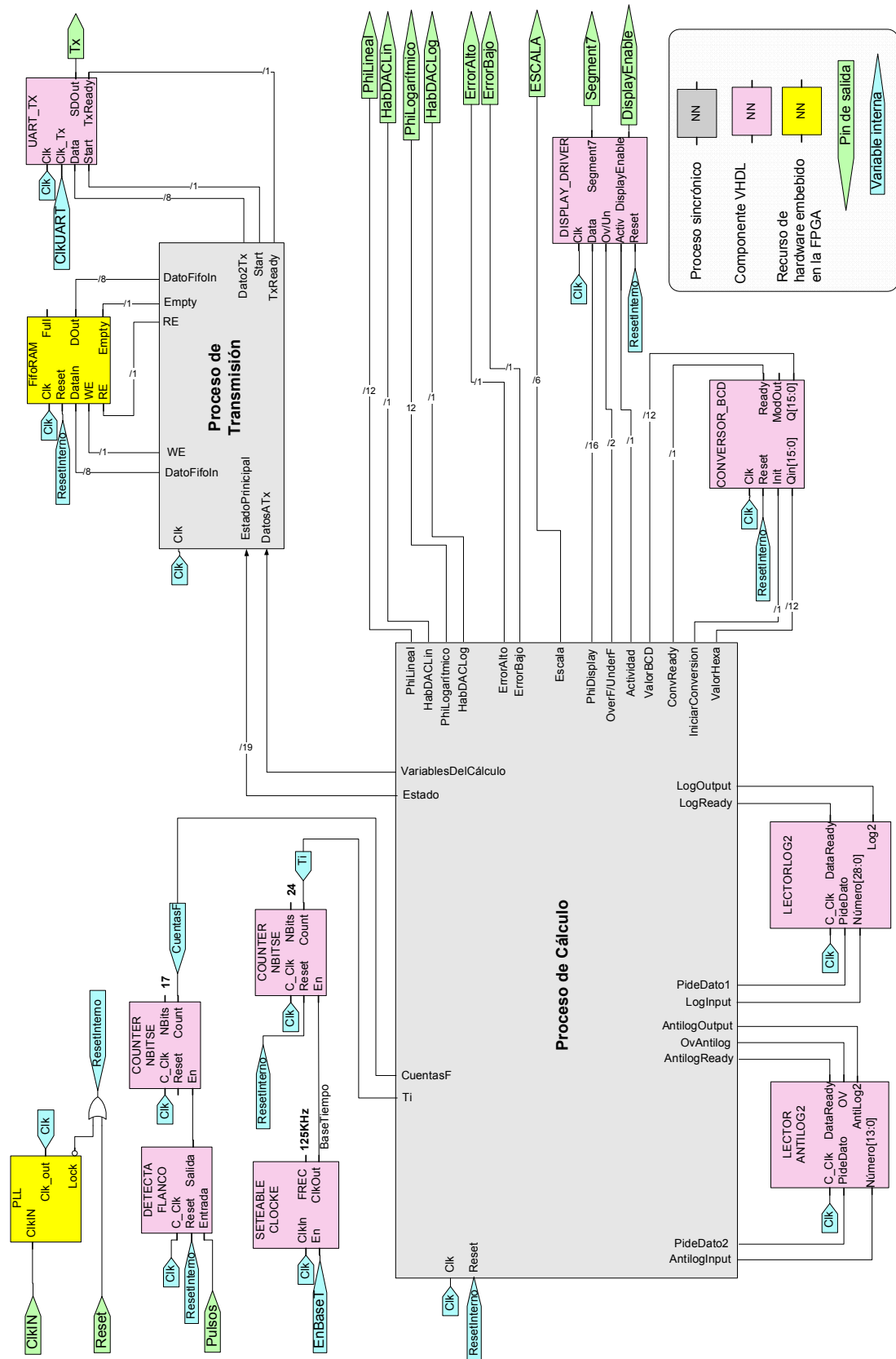


Figura 33. Diagrama en bloques principal del código VHDL.

El bloque ‘Proceso de cálculo’ es una máquina de estado de 19 estados, que tarda 47 ciclos de clock, equivalentes a 1175ns, en hacer los cálculos descritos en el Capítulo 2. También obtiene el valor BCD para indicar en el display utilizando el componente CONVERTOR_BCD.

Para hacer los cálculos se utilizan los componentes ‘LECTOR_LOG2’ y ‘LECTOR_ANTILOG2’. Dentro de estos componentes se resuelven los cálculos descritos en las fórmulas (62.) y (63.), y (65.). Dentro de cada uno hay un componente que es una tabla de 256 valores de 8 bits que contienen el logaritmo base 2 de 1.00_H a $1.FF_H$, en ‘LECTOR_LOG2’, y $2^{0.00_H}$ a $2^{0.FF_H}$ en ‘LECTOR_ANTILOG2’.

El componente ‘DISPLAY_DRIVER’ recibe el valor ‘PhiDisplay’, que cambia cada vez que se obtiene una nueva medición. El valor corresponde al valor de flujo lineal dividido por 10^{escala} . El punto decimal está fijo en centro, pudiéndose indicar desde 0.01 a 10.00. Se hace un barrido de los 4 displays, habilitando el transistor correspondiente durante 1ms cada uno. El valor representado se vuelve a leer antes de comenzar cada barrido. En caso de tener que indicar underflow o overflow, se indican de la siguiente manera:

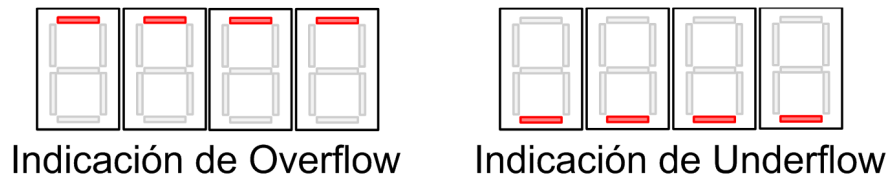


Figura 34. Indicaciones en el display.

El punto decimal del último display se prende durante 100ms luego de la detección de un pulso a la entrada, para indicar que se detecta actividad.

El proceso de transmisión consiste en dos máquinas de estado, una que llena la memoria RAM y otra que envía datos desde la RAM a la UART una vez que ésta tiene algún dato (‘EMPTY’ = 0). La RAM es usada como una FIFO organizada como 9x512 bits.

4.1.2 Diagrama funcional del Proceso de Cálculo

En el Anexo I muestran las operaciones que realiza el bloque ‘Proceso de Cálculo’. Está separado en bloques según la fórmula que implementan. Se indican los tamaños de las variables, indicando la cantidad de bits enteros y decimales. Los nombres de las variables indicadas son las utilizadas en el código VHDL.

Las variables tienen el tamaño necesario para representar los valores posibles de cada magnitud. De esta manera se ahorra espacio en el dispositivo ya que las variables son registradas, y cada bit extra implica la asignación de una celda como elemento de memoria. Donde es necesario se extiende el vector de bits con un bit ‘0’ como bit más significativo para poder hacer las sumas y restas correctamente, manteniendo las variables con el tamaño mínimo.

Se debe hacer notar que no hay ninguna indicación temporal en el diagrama. Dado que se instancia un único componente de ‘LECTOR_LOG2’ y un único

componente de 'LECTOR_ANTILOG2', los cálculos mostrados se van intercalando en el uso de estos dos componentes.

Se describe a continuación qué fórmula implementa cada bloque del Anexo I

- **Bloque 1:** implementa la fórmula (49.) :

$$\varphi_K = \text{antilogaritmo2} \left(\logaritmo2 \left(\frac{K_i}{\text{Periodo}} \right) + j - \logaritmo2 \left(\text{floor} \left(\frac{T}{\text{Periodo}} \right) \right) + \logaritmo2(F_{\text{Correccion}}) \right)$$

- **Bloque 2:** realiza el cálculo del promedio EWMA del flujo instantáneo, implementando la fórmula (50.) :

$$\varphi_N(i) = \varphi_N(i-1) + \frac{\varphi_{i,K} - \varphi_N(i-1)}{N}$$

El valor del parámetro del filtro es determinado en el test de hipótesis de la medición anterior, comenzado con $c = 0$, $N = 1$ para la primer medición.

- **Bloque 3:** realiza la determinación de la escala comparando con el valor $9.5 \cdot 10^{\text{Escala}}$, para que el cambio de escala se produzca antes de llegar al fondo de escala tanto en la señal analógica como en el display.
- **Bloque 4:** calcula el logaritmo de flujo promedio que se requiere para el cálculo de flujo logarítmico, lineal e indicación en display.
- **Bloque 5:** indica que el valor del logaritmo de la 10^{escala} que se utiliza para el cálculo de flujo lineal e indicación en display, se obtiene de una tabla en función del valor de la variable 'D'.
- **Bloque 6:** calcula el flujo logarítmico según la fórmula (58.):

$$\varphi_{\log} = \text{antilog2} \left(\logaritmo2(\logaritmo2(\varphi_{\text{Prom}})) + \logaritmo2 \left(\frac{2^{12}}{10 \cdot \logaritmo2(10)} \right) \right)$$

Al inicio del bloque resta 'n' porque el cálculo del logaritmo supone que a la entrada el número no tiene decimales. También se requiere más adelante como indica la "Nota 6" en el diagrama funcional.

Se chequea si el logaritmo es cero, lo que indicaría que el flujo promedio es menor a 1. En ese caso la salida de flujo logarítmico satura en el valor correspondiente a 1c/s.

- **Bloque 7:** realiza el cálculo del valor BCD indicado en el display.

$$\varphi_{\text{display}} = \frac{\varphi_{\text{Prom}}}{10^{\text{Escala}}} \cdot \frac{999}{A.00_H - 1} \quad 68.$$

A este valor se le truncan los bits decimales y se toman $4 \cdot N_{\text{DISPLAY}} = 16$ bits que son convertidos a BCD.

- **Bloque 8:** realiza el cálculo del flujo lineal que controla al DAC correspondiente, según la fórmula (54.):

$$\varphi_{lineal} = \text{Antilogaritmo2}(\text{logaritmo2}(\varphi_{Prom}) + \text{logaritmo2}(409.6) - \text{logaritmo2}(10^{Escala}))$$

Si el valor del flujo promedio dividido por la escala da menor a uno, indica "0.00" en el display.

- **Bloque 9:** realiza el cálculo del promedio EWMA con parámetro 2N del flujo instantáneo. implementa la fórmula (51.) :

$$\varphi_{2N}(i) = \varphi_{2N}(i-1) + \frac{\varphi_{i,K} - \varphi_{2N}(i-1)}{2 \cdot N}$$

El valor del parámetro del filtro es determinado en el test de hipótesis de la medición anterior.

- **Bloque 10:** realiza el test de hipótesis que compara la diferencia en valor absoluto del flujo instantáneo con el flujo promedio, con el valor del sigma del flujo instantáneo por NSIGMAS = 3. Tiene prioridad sobre el bloque 11, que no se ejecuta si se decide bajar el valor de 'c'. Queda establecido el nuevo valor de 'c' para la próxima medición.
 - **Bloque 11:** realiza el test de hipótesis que compara la diferencia en valor absoluto del flujo promedio N con el flujo promedio 2N, con el valor del sigma del flujo promedio N dividido 4. Si la diferencia es menor a esa porción del valor del sigma, sólo aumenta el valor de 'c' si en el bloque 10 no se decidió bajarlo.

4.1.3 Depuración del código VHDL.

Para depurar el código se utilizó la herramienta Modelsim AE, que está incluida dentro del entorno Libero, con la que se realizó la simulación a nivel comportamiento del código VHDL.

Para la simulación se requiere crear un 'testbench', esto es un archivo escrito en VHDL donde se instancia el componente a simular y se lo 'conecta' a las señales que simulan el comportamiento de las entradas. En esta modalidad se utilizan otras funcionalidades del VHDL como es la especificación de tiempo, un concepto que no tiene sentido en código escrito para ser implementado en hardware.

Se utilizó el mismo procedimiento de cálculo en Matlab utilizado en el Capítulo 2 para la elección de parámetros, para generar los vectores resultado de las variables más importantes, como flujo instantáneo, flujo promedio, valores de 'c' y 'K', entre otros. Los vectores resultado se guardaron expresados en notación hexadecimal, para facilitar la comparación en el entorno Modelsim. Como entrada al cálculo se generaron conjuntos de datos de tiempo entre pulso con distribución exponencial, con simulación de ancho de pulso real porque se simula la entrada real de pulsos, y ésta debe tener una duración distinta de cero para poder ser leída.

Los tiempos entre pulsos se simularon según el modelo no paralizante de tiempo muerto, en el que un nuevo pulso que aparece en un tiempo más corto que el ancho de pulso no extiende la longitud del pulso. Este modelo simplifica la generación de los vectores de simulación, ya que el ancho de pulso es constante. Para la depuración del código no interesa analizar el error por

tiempo muerto que introduce el ancho de pulso. Este error se analiza en detalle en el párrafo 5.1.1.

El testbench del impulsímetro requiere únicamente generar los estímulos de 2 señales, la del clock de 40MHz y la de la entrada de pulsos aleatorios. En la Figura 35 se un extracto del código VHDL que se utiliza como estímulo. El vector de tiempos es el que se obtiene del conjunto de datos de tiempos entre pulsos en Matlab. El resultado de la simulación se compara con el resultado obtenido y guardado en el archivo Excel indicado.

```

Reseti <= '0' after 10 ns;
Clki <= not Clki after 12.5 ns;

tb : PROCESS
BEGIN
    while Reseti = '0' loop
        --tiempos generados tomando en cuenta el pulso
        --real de 100nseg
        --Corresponde a caso '1' de PuestaenMarcha
        --Comparar resultados con reg_VectorSimuF1000000TOP100.xls
        wait for      0.15542      us;  Pulsosi <= '1';
        wait for      0.1          us;  Pulsosi <= '0';
        wait for      1.4264       us;  Pulsosi <= '1';
        wait for      0.1          us;  Pulsosi <= '0';
        wait for      4.1123       us;  Pulsosi <= '1';
        wait for      0.1          us;  Pulsosi <= '0';
        .....
    end loop;
end process;

```

Figura 35. Porción del código VHDL usado en una simulación.

4.1.4 Utilización de recursos de la FPGA

La herramienta de 'place&route', llamada 'Designer', realiza la implementación física del código en la FPGA elegida. Terminado el proceso de implementación, permite hacer un análisis de tiempo y análisis de potencia disipada.

El proyecto ocupa el 28% de los recursos de celdas lógicas de la FPGA, como se ve en la salida del reporte de salida del 'Designer' en la Figura 36.

Compile report: =====				
CORE	Used:	6863	Total:	24576 (27.93%)
IO (W/ clocks)	Used:	57	Total:	154 (37.01%)
Differential IO	Used:	0	Total:	35 (0.00%)
GLOBAL (Chip+Quadrant)	Used:	6	Total:	18 (33.33%)
PLL	Used:	1	Total:	1 (100.00%)
RAM/FIFO	Used:	1	Total:	32 (3.13%)
Low Static ICC	Used:	0	Total:	1 (0.00%)
FlashROM	Used:	0	Total:	1 (0.00%)
User JTAG	Used:	0	Total:	1 (0.00%)

Figura 36. Reporte de salida de la herramienta 'Designer'. Utilización de recursos de la FPGA utilizados.

Además se utilizó el PLL, el 3% de la memoria RAM disponible, y 57 pines de entrada/salida.

5. Capítulo 5

5.1 Verificación utilizando el simulador de pulsos aleatorios

Para la verificación del equipo se utilizó un simulador de detector de flujo neutrónico desarrollado en el sector Instrumentación y control de CNEA. El mismo está basado en una FPGA de Xilinx, e implementa un generador pulsos aleatorios con distribución Poisson. El parámetro de la distribución λ =flujo, es controlado desde una PC a través de un puerto USB. En la FPGA se generan números aleatorios de 32bits con distribución uniforme entre 0 y 1 en cada ciclo del clock de la FPGA ($f = 50\text{MHz}$), que junto con el valor enviado por la PC determinan con una resolución en tiempo de 20ns, la aparición o no de un pulso [6]. La tasa de cuentas generada por este equipo cubre todo el rango de medición del impulsímetro digital. Es posible simular flujo con una cierta tasa de variación variando el parámetro que envía la PC. La salida del simulador es una señal LVTTTL de 3.3V, con ancho de pulso variable desde 20ns a 1.3us, configurado desde la PC.

Para la verificación se generaron perfiles de flujo constante y con salto por 2, para analizar el retardo en la detección del salto.

Para el análisis de los resultados obtenidos se utilizó la salida optoaislada de datos serie del impulsímetro, mediante la cual se dispone de todos los valores de medición y de algunas variables internas. Para esta verificación se comparan los resultados con el perfil teórico únicamente.

El simulador envía los tiempos de interarribo con un timestamp por bloques, lo que permitiría rehacer el cálculo con Matlab para comparar los resultados, como se hizo en la fase de depuración del código VHDL, pero ahora con el valor real indicado por el equipo, obtenidos por la salida optoaislada del impulsímetro.

Para hacer la sincronización entre los datos de resultado y el perfil teórico es importante resetear primero al impulsímetro y luego iniciar la simulación. De este modo se sincroniza el inicio de la base de tiempo, ya que luego del 'reset' espera la detección de un pulso de entrada antes de iniciar la acumulación de tiempo. De este modo el primer valor de flujo es un valor válido y no resulta excesivamente bajo.

5.1.1 Error por tiempo muerto

En casi todos los sistemas de detección se requiere una cantidad mínima entre 2 eventos próximos para poder distinguirlos individualmente. Este tiempo mínimo necesario se suele llamar 'tiempo muerto'. Debido a la naturaleza aleatoria de los pulsos a medir, existe una probabilidad distinta de cero de que un pulso esté tan próximo al anterior que resulte dentro del tiempo muerto, detectándose menor cantidad de pulsos. Este error es tanto peor, cuanto mayor sea el flujo.

Existen dos modelos de comportamiento de tiempo muerto: modelo 'paralizable' y modelo 'no paralizable'. Son modelos ideales y los sistemas pueden describirse aproximadamente por uno de ellos.

En el caso de la cadena de medición de flujo neutrónico, se deben tomar en cuenta todos los elementos del sistema. En el caso del contador proporcional,

cuantos más neutrones se detecten el nivel de corriente sube hasta hacerse una continua con variaciones proporcionales al flujo, nivel en el cuál se puede medir en modo Campbell. Pero visto desde el impulsímetro, según sea la técnica de medición, se verá como un modelo no paralizante si se mide por integración de la señal de pulsos, y la medición llegará a un máximo. O como modelo paralizante, que es el caso del impulsímetro digital, en el que se ve como la desaparición de pulsos y por lo tanto el conteo baja.

La ecuación que calcula el flujo observado según el tiempo muerto para el caso del modelo paralizante que se aplica al impulsímetro digital, es la (69.) [1]. En la Figura 37 se muestran los flujos observados para ambos modelos.

$$m = n \cdot e^{-n \cdot \sigma} \quad 69.$$

donde m : flujo observado
 n : flujo real
 σ : tiempo muerto.

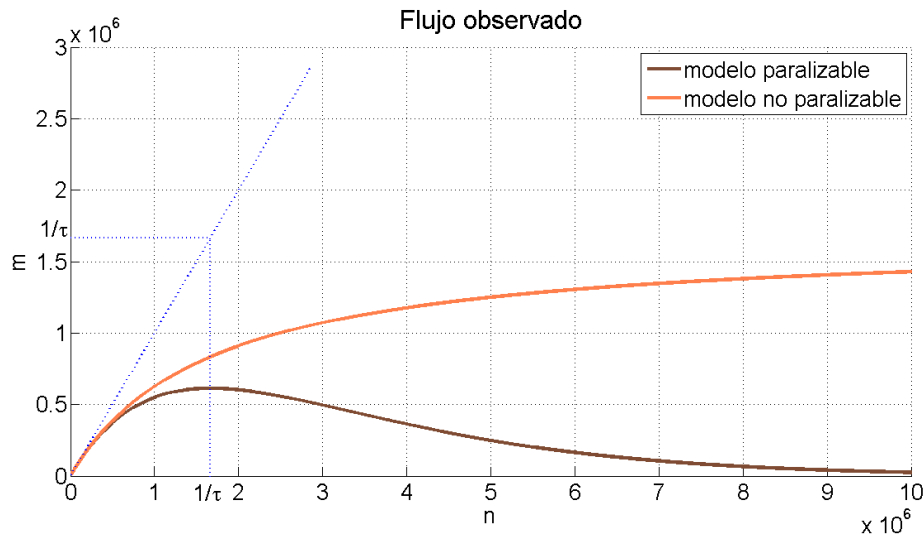


Figura 37. Flujo observado en cada modelo según el flujo para tiempo muerto = σ

Para la electrónica, el tiempo muerto es debido al ancho del pulso, que depende del detector y del amplificador conformador, y a la separación entre pulsos mínima que debe haber entre flanco de bajada y subida del siguiente. En el caso del impulsímetro debe haber 25ns de separación, que es el ciclo del clock interno de la FPGA, a la que se muestrea la señal de entrada.

En la Figura 38 se muestra el error que se observará en el proceso de verificación en que se simuló con ancho de pulso igual a 100ns. Para la cadena de arranque del RA1 la electrónica detecta pulsos separados 600ns. Y para la cadena Campbell que se desarrolla en el sector de Instrumentación y control de CNEA, el ancho de pulso está alrededor de 400ns.

En el proceso de verificación se tendrá un error alrededor del 11% para flujo igual a 10⁶ c/s.

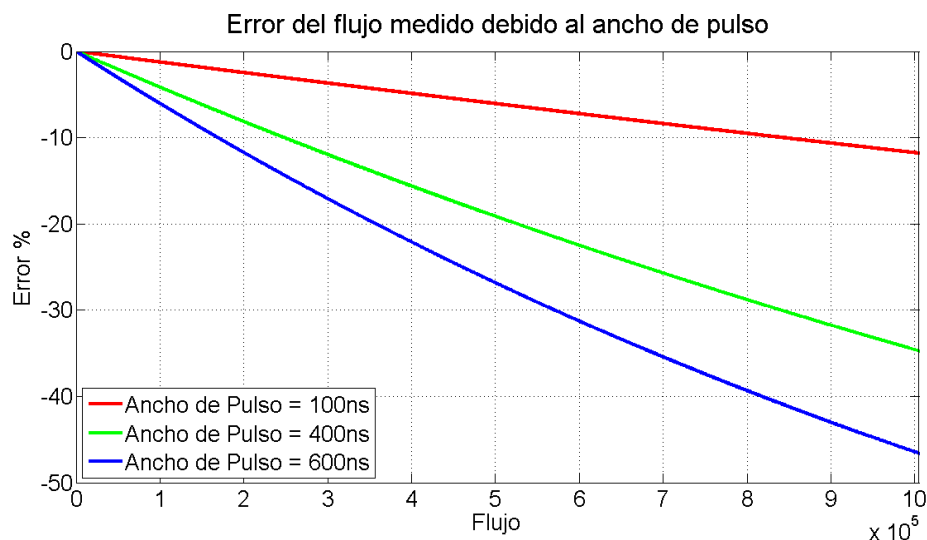


Figura 38. Error del flujo observado según el ancho de pulso.

5.1.2 Software utilizado en la adquisición de datos

Para la verificación del impulsímetro se utilizan 3 aplicaciones desarrolladas ad-hoc.

La primera es la que controla al simulador de detector. En la Figura 39 se muestra la ventana de la misma.

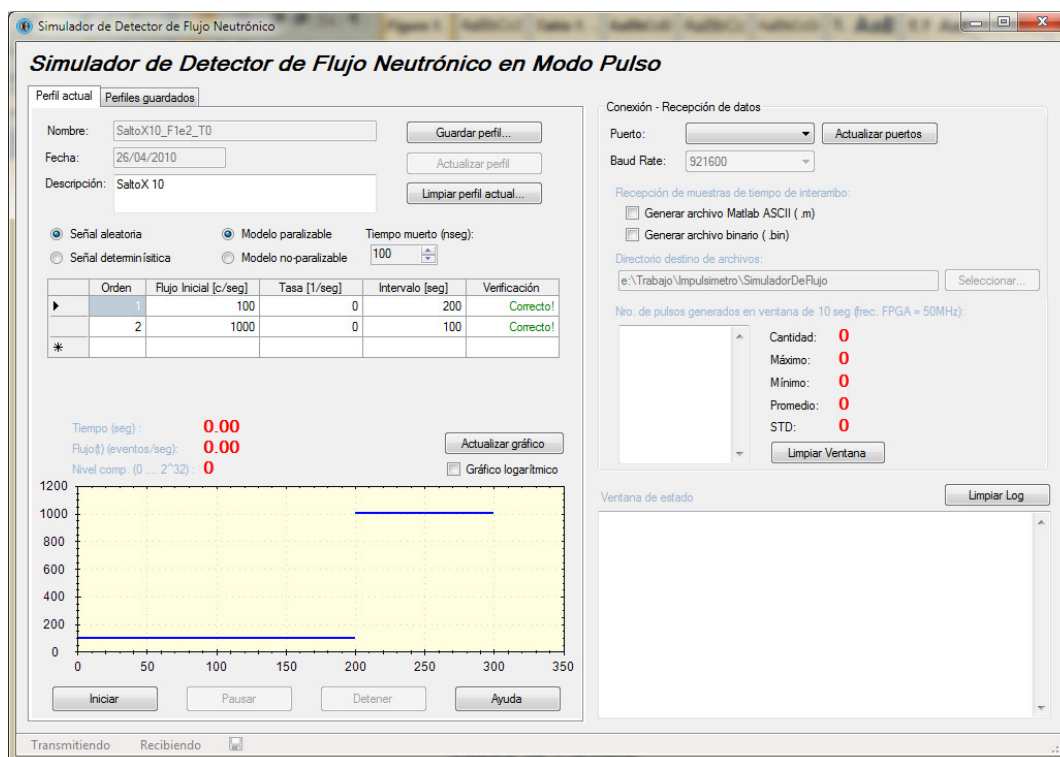


Figura 39. Ventana principal de la aplicación Simulador de Detector.

La aplicación tiene una sección para la configuración de perfiles de simulación que quedan guardados en un archivo XML. Se configuran por secciones el tiempo, el flujo y la tasa de variación en cada sección. Además se

puede elegir modelo 'paralizable' o 'no paralizable', y el ancho de pulso de la señal generada.

La aplicación utiliza un puerto COM virtual utilizando un puerto USB mediante un driver adecuado. La velocidad de transmisión de datos es 921600 bits/s, la cual no se podía lograr con un puerto COM normal. El integrado que permite la conversión de trama serie TTL a USB, está integrado al equipo, por lo que la conexión a la PC se hace con un cable USB A-B normal.

A través de esta conexión el equipo recibe el parámetro de simulación y a la vez transmite cada 10 segundos la medición del flujo promediado en ese tiempo. También se transmiten todos los tiempos de interarribo que sea posible transmitir, lo cual dependerá del flujo simulado. Cuanto más alto el flujo mayor cantidad de datos debe transmitir y se llena el canal de datos. Junto con el archivo con los datos de tiempo interarribo, se genera un archivo XML con la configuración del perfil con un nombre similar al elegido para guardar los tiempos.

La segunda aplicación necesaria es la que recibe los datos de las variables internas del impulsímetro. La recepción de datos se realiza también por un puerto COM virtual. En este caso se requiere un cable con el integrado que hace la conversión de trama serie TTL a USB, e instalar el driver correspondiente en la PC. En la Figura 40 se muestra la ventana de la aplicación.

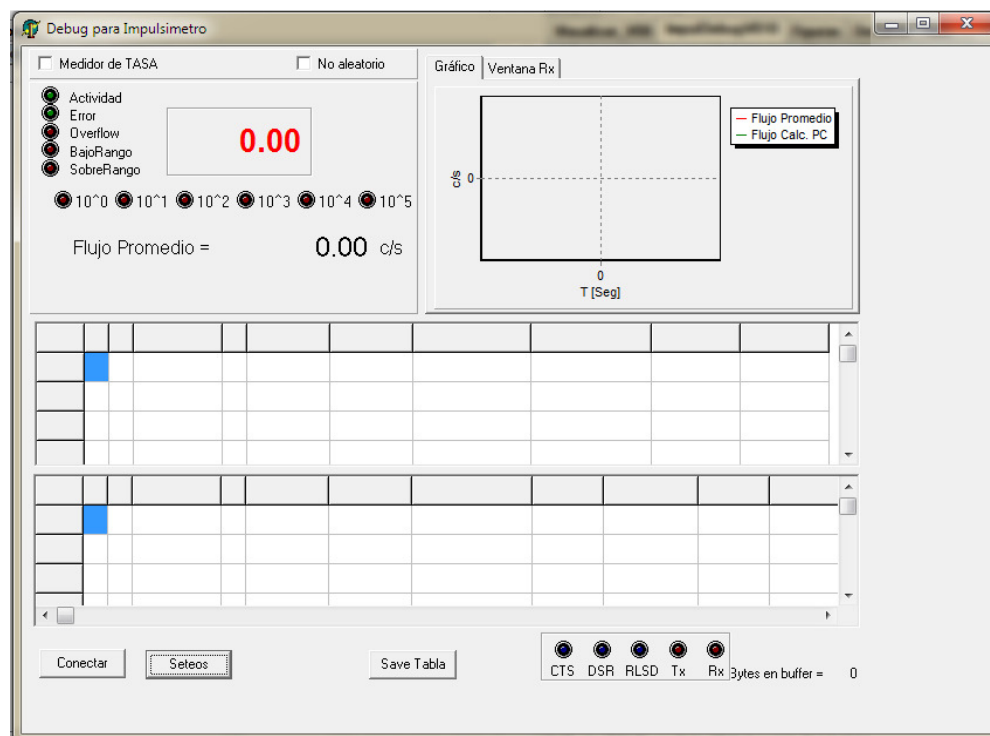


Figura 40. Ventana de la aplicación “Debug para Impulsímetro”.

Una vez abierto el puerto con velocidad de recepción de 115200bits/s, se muestran en las tablas En los ejes de la derecha se grafican la variable flujo promedio enviado por el equipo y un flujo calculado a partir de la variable tiempo acumulado y la cantidad de cuentas K. Esto se utilizó en la fase de

debug. Las tablas se pueden grabar en un archivo con formato texto separado por comas para hacer el análisis con la aplicación siguiente.

La tercera aplicación se utiliza para analizar los resultados de medición obtenidos. Está desarrollada en Matlab y la ventana que abre se ve en la Figura 41.

La aplicación permite elegir un directorio donde se encuentran los resultados de las mediciones. Sólo son visibles los archivos con un formato de nombre específico. Junto con este archivo se debe encontrar el archivo con el perfil que genera la primera aplicación. Todos esto se hizo para limitar la posibilidad de error y forzar a hacer lo toma de datos de forma controlada.

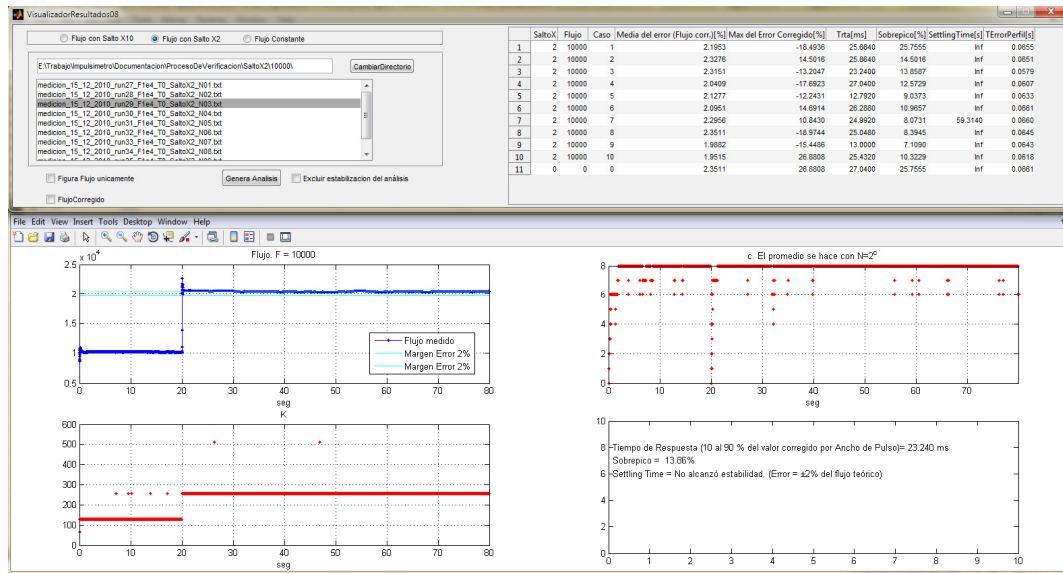


Figura 41. Aplicación 'VisualizadorResultados' desarrollada en Matlab.

En la tabla de la derecha se muestran los valores de los indicadores de performance para los archivos de mediciones que se encuentren en el directorio especificado. La tabla se guarda automáticamente en formato de texto separado por comas.

5.1.3 Procedimiento de adquisición de datos

Se requiere una fuente de alimentación partida de entre $\pm 24V$ a $\pm 20V$, y generar $+9V$ a partir de ésta. Para la conexión a PC de la placa del impulsímetro y del simulador de flujo, se requieren 1 cable USB A-B, 1 cable Convertidor USB a COM, basado en CP2101, y un cable coaxil BNC-cocodrilo

En la Figura 42 se muestra cómo se conecta la placa eurocard del impulsímetro al Simulador de Flujo y a la PC para la toma de datos.

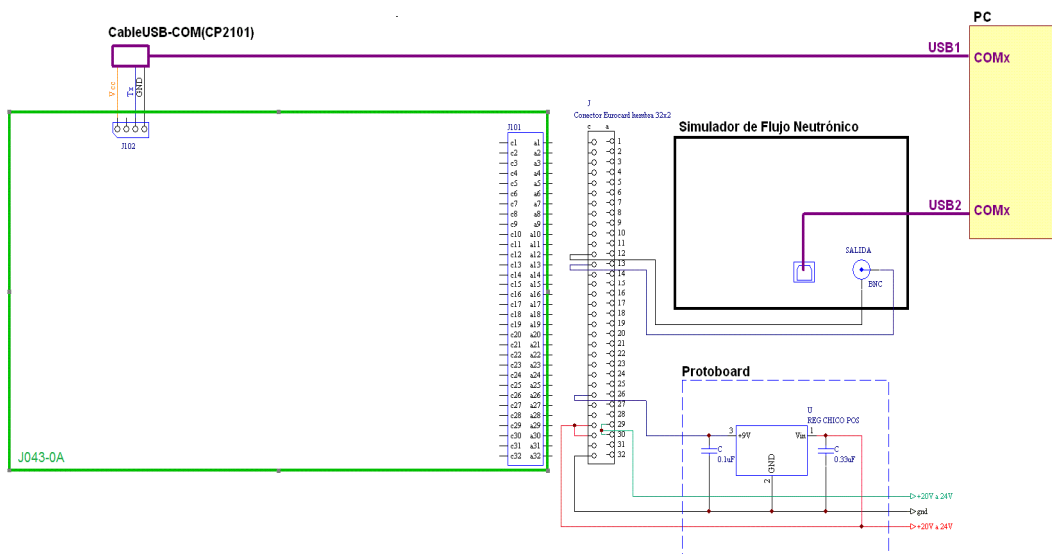


Figura 42. Conexionado para la adquisición de datos.

Para cada uno de los perfiles indicados en la Tabla 7, se generaron 10 mediciones. El procedimiento de toma de datos es la siguiente:

1. En el programa “Simulador de Detector de Flujo Neutrónico”, elegir el puerto COM donde se encuentre conectado el Simulador. Activar la opción de “Generar archivo binario (.bin)”. Luego cargar el perfil que corresponda de la lista.
2. En el programa “Monitor de Impulsímetro”, clicar el botón “Seteos” y elegir el puerto COM dónde se encuentre conectado el módulo Impulsímetro. La velocidad debe ser 115200.
3. En cada medición la secuencia a ejecutar es:
 - i. Pulsar el “Reset” en el módulo Impulsímetro Digital.
 - ii. Apretar el botón “Abrir Puerto” en el programa “Monitor de impulsímetro”.
 - iii. Apretar el botón “Iniciar” en el programa “Simulador de Detector de Flujo Neutrónico”.
 - iv. Al finalizar el perfil de simulación, apretar el botón “Cerrar Puerto” en el programa “Monitor de impulsímetro”.
 - v. Guardar los datos obtenidos, presionando “Guardar Tabla” en el programa “Monitor de Impulsímetro”. El nombre de archivo sugerido comenzará con el nombre del archivo que guardó el simulador, y se deberán reemplazar las ‘x’ por lo que corresponda. Por ejemplo:

medicion_19_11_2010_run2_Fxex_Tx_SaltoXxx_Nxx.txt

Para flujo = 100 c/s, Tasa= 0%, SaltoX = 1, Medición = 3, se reemplaza por:

medicion_19_11_2010_run2_F1e2_T0_SaltoX1_N03.txt

5.1.4 Tablas de Perfiles

Se simula un flujo constante para cada década del rango de medición. Además se simula la curva de flujo utilizada para la determinación de los parámetros fijos de medición en el Capítulo 2, y que se muestra en la Figura 4, con salto por 2. Los perfiles generados con el simulador de flujo se listan en la Tabla 7. Los otros parámetros del simulador necesarios se configuraron en:

- Señal aleatoria : *sí*
- Modelo Paralizable: *sí*
- Tiempo muerto (nseg) = 100

Se utiliza un ancho de pulso pequeño ya que el objetivo es analizar el comportamiento del equipo, y no simular el comportamiento real de un detector.

Flujo Inicial [c/s]	Flujo constante	Salto X2	
	TSimulación [s]	TSalto[s]	TSimulación [s]
1e0	2000	-	-
1e1	2000	2000	2000
1e2	400	200	200
1e3	90	60	60
1e4	90	20	60
1e5	60	20	60
1e6	60	20	60

Tabla 7. Listado de los perfiles simulados.

En todos los casos se utilizan los siguientes indicadores de performance para analizar los resultados, que son los mismos que los utilizados en el capítulo 2 para la elección de los parámetros:

- Valor medio del RMS del error del flujo promedio respecto al teórico, para flujo constante.
- Tiempo de retraso del flujo promedio en llegar al 90% del valor final en un salto de flujo por 2, medido respecto al tiempo teórico del perfil.
- Valor medio del tiempo de refresco de cada medición, para flujos constante.

5.2 Resultado de las simulaciones

Para las 10 mediciones realizadas se guardaron los datos que recibe la aplicación 'Debug Impulsímetro', y el archivo con los datos del perfil y los tiempos de interarribo que genera el simulador de detector.

Entre los datos que envía el impulsímetro digital se tiene el valor de la variable flujo promedio, que es de la que se obtienen los valores a mostrar en el display y a enviar a los DACs. El error de esta variable respecto al valor de flujo teórico esperado se puede comparar con los resultados obtenidos para el caso de simulación 19 elegido en el Capítulo 2.

Como se tiene el vector completo con el valor de la variable en función del tiempo, es posible calcular el valor de la raíz cuadrática media del error del flujo promedio. En la Tabla 8 se listan los valores medios del RMS del error, en función del flujo. Para el flujo medido, el valor medio es sobre 10 mediciones, y para el caso calculado, sobre 100 corridas. Los valores esperados se grafican en la Figura 17.

	RMS del error de flujo promedio medido		RMS del error de flujo promedio calculado en el Capítulo 2	
Flujo [c/s]	Valor medio [%]	Desvío [%]	Valor medio [%]	Desvío [%]
10^0	9.32	1.86	14.63	5.12
10^1	3.45	1.33	1.40	0.55
10^2	2.67	0.65	1.34	0.55
10^3	2.30	0.54	1.52	0.60
10^4	1.96	0.10	1.15	0.56
10^5	0.59	0.06	0.64	0.21
10^6	8.42	0.02	0.18	0.09

Tabla 8. Valores de RMS del error de flujo promedio medido, y calculado en Matlab.

El error más alto para flujo igual a 1 c/s, es debido a que el medidor de flujo satura en 1 c/s, entonces la media esperada es:

$$E(\varphi_{medido} = 1) = \int_0^1 1 \cdot g_{\varphi|K=32}(\varphi) d\varphi + \int_1^{\infty} \varphi \cdot g_{\varphi|K=32}(\varphi) d\varphi \quad 70.$$

Se calculó en Matlab utilizando la fdp de flujo corregido, ecuación (29.) y el resultado fue:

$$E(\varphi_{medido} = 1) = 1.072 \text{ c/s}$$

Es decir que se tendrá un error de más del 7% para flujo igual a 1 c/s.

Para flujo 10^6 c/s se tiene un valor importante de error por tiempo muerto, que no fue analizado en el capítulo 2.

Para los valores intermedios de flujo el valor medio del RMS del error del flujo promedio resulta entre el 1 y 2% más alto que el esperado. Esta desviación sugiere que hay alguna fuente no considerada de error, que se debe analizar. Es importante señalar que el análisis realizado en el capítulo 2 con el que se obtuvieron los valores esperados de los indicadores, facilita la detección de anomalías en los resultados de la implementación. Esto es particularmente importante cuando se trabaja con señales aleatorias, en que los errores reales pueden enmascarse por la dispersión propia de la medición.

Se analiza el tiempo de retraso para perfil de flujo con saltos por 2 para comparar con los valores obtenidos en el capítulo 2. En la Tabla 9 se muestran los valores medios sobre las 10 mediciones para cada flujo de los tiempos de retraso entre el tiempo del salto del perfil simulado y el instante en que el flujo promedio medido llega al 90% del valor final. Los valores esperados con los que se comparan están graficados en la Figura 18.

	Tiempo de retraso medido		Tiempo de retraso calculado en el Capítulo 2	
Flujo [c/s]	Valor medio [segundos]	Desvío [segundos]	Valor medio [segundos]	Desvío [segundos]
10^1	78.601	39.832	83.87	5.6E-01
10^2	7.415	3.935	8.81	5.3E-02
10^3	0.531	0.290	0.849	5.4E-03
10^4	0.064	0.003	0.063	6.4E-05
10^5	0.056	0.007	0.056	8.0E-05
10^6	0.051	0.006	0.066	4.6E-05

Tabla 9. Tiempos de retraso medidos comparados con los calculados en Matlab.

Los tiempos de retardo resultan menores a 1 segundo para flujos mayores a 10^3 c/s.

Se calculó el valor medio del tiempo de refresco de cada medición y se halló el valor medio de las 10 mediciones, para cada rango de flujo. Para flujo constante el tiempo de refresco se mantiene estable, y está determinado por error mínimo que se eligió en el Capítulo 2, ERROR_INFERIORF = 20%. Para flujos altos en cambio, el tiempo entre mediciones está determinado por el tiempo de acumulación requerido, elegido en TACUM = 10mS. En la Tabla 10 se muestran los valores medidos y los esperados, graficados en la Figura 16.

Se ve que los valores de tiempo de refresco medidos son similares a los esperados, siendo alrededor de 10ms para flujos altos.

	<i>Tiempo de refresco medido</i>		<i>Tiempo de refresco calculado en el Capítulo 2</i>	
<i>Flujo [c/s]</i>	<i>Valor medio [segundos]</i>	<i>Desvío [segundos]</i>	<i>Valor medio [segundos]</i>	<i>Desvío [segundos]</i>
10^0	31.471	8.4E-01	31.99	1.3E-01
10^1	3.142	2.7E-02	3.201	1.4E-02
10^2	0.313	1.2E-03	0.320	1.3E-03
10^3	0.031	1.4E-04	0.032	1.3E-04
10^4	0.013	1.5E-05	0.013	5.4E-05
10^5	0.014	1.0E-04	0.013	1.9E-04
10^6	0.018	2.4E-06	0.016	7.9E-06

Tabla 10. Tiempos de refresco medido comparados con los esperados.

En la Figura 43 se grafican las curvas de resultado para flujo constante igual a 10^3 c/s, y en la Figura 44 se grafican las curvas de resultado de la medición del perfil con salto por 2 para flujo inicial 10^3 c/s. En ambos gráficos se muestran los resultados para las 10 mediciones realizadas para cada perfil.

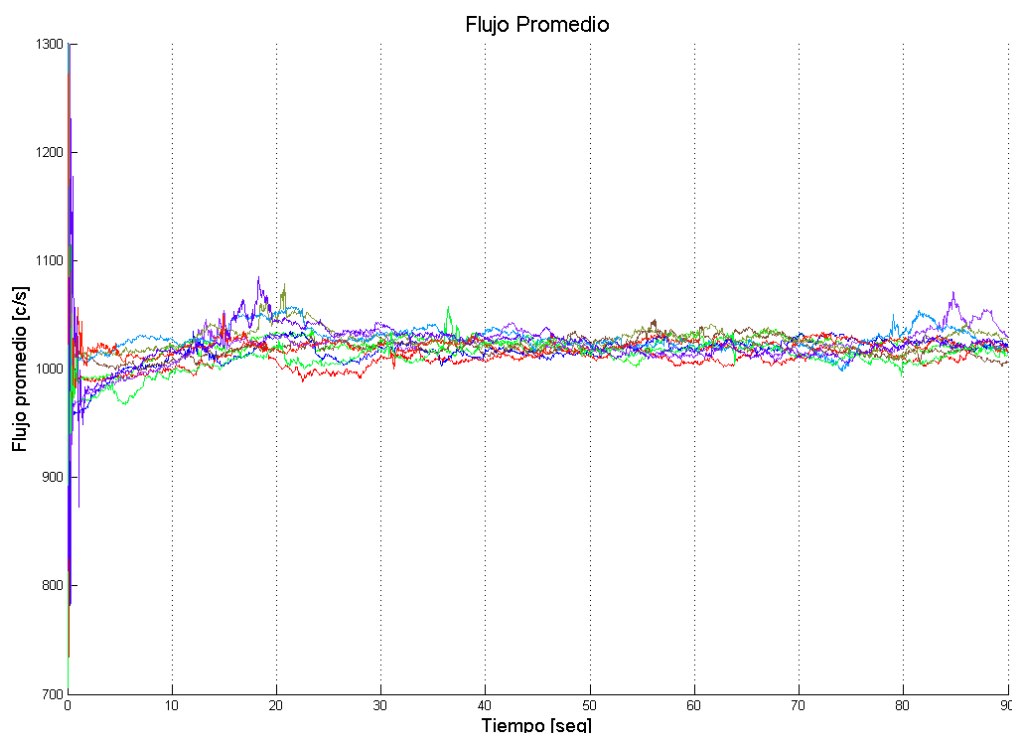


Figura 43. Flujos promedio medidos en las 10 mediciones con flujo igual a 10^3 c/s.

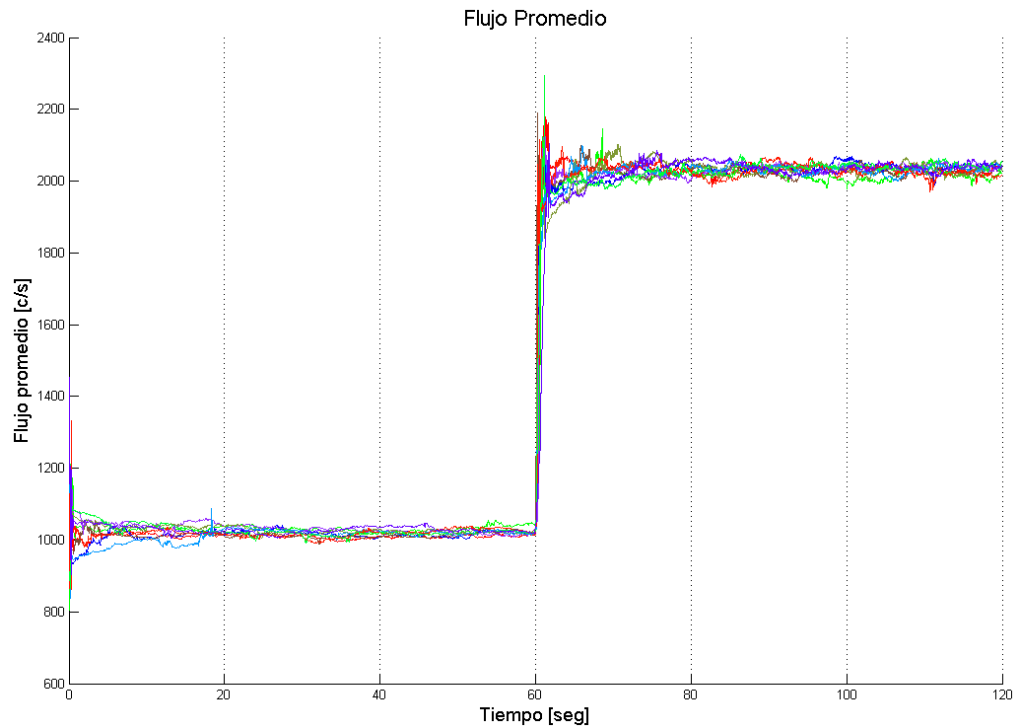


Figura 44. Flujos promedio medidos en las 10 mediciones de perfil de salto por 2 para flujo inicial igual a 10^3 c/s.

6. CONCLUSIONES

El equipo desarrollado permite la medición de flujo neutrónico en régimen de pulsos en el rango de 1 c/s a 10^6 c/s con tiempo de respuesta adaptado al flujo en forma automática, manteniendo un error estadístico controlado. El compromiso entre tiempo de respuesta y precisión es resuelto por el método de cálculo, con una velocidad de respuesta ante cambios de flujo que sería imposible de lograr si se dependiese de la intervención de un usuario.

Dado que los cambios de los parámetros de medición son determinados en función de valor de flujo obtenido en cada medición, los tiempos de respuesta se mantienen proporcionales al flujo.

En la Tabla 11 se muestran los valores de los indicadores valor medio del RMS del error, valor medio del retraso en la detección de un salto por 2 de flujo, y valor medio del tiempo de respuesta, en función del flujo. Estos resultados se obtuvieron utilizando un simulador de detector neutrónico que genera pulsos aleatorios con tiempo de interarribo con distribución exponencial.

Los tiempos de respuesta son menores a 50ms para flujo mayor a 10^3 c/s.

El error de la medición se mantiene alrededor del 2% en los rangos intermedios, siendo algo mayor para flujo igual a 10^1 c/s. El error para el rango de 10^6 c/s es debido a la pérdida de pulsos por tiempo muerto, y para el rango de 10^0 c/s, por la implementación del cálculo, que satura el flujo en 1c/s.

El retraso en la detección de un salto por 2 del flujo, es menor a 1 segundo para flujos mayores a 10^3 c/s. Se mide el retraso para salto por 2 que resulta más desfavorable ya que se enmascara en la dispersión propia de la medición.

<i>Flujo [c/s]</i>	<i>Valor medio del RMS del error de flujo promedio medido [%]</i>	<i>Valor medio del tiempo retraso medido en la detección de un salto por 2 de flujo [segundos]</i>	<i>Valor medio del tiempo de refresco medido [segundos]</i>
10^0	9.32	-	31.471
10^1	3.45	78.601	3.142
10^2	2.67	7.415	0.313
10^3	2.30	0.531	0.031
10^4	1.96	0.064	0.013
10^5	0.59	0.056	0.014
10^6	8.42	0.051	0.018

Tabla 11. Valores medidos. Valores medios sobre 10 mediciones.

Actualmente el valor analógico proporcional al flujo logarítmico es utilizado como valor de disparo en el arranque por la lógica de SCRAM, para la condición de flujo mayor que cierto nivel. Esto implica tener un módulo comparador para realizar decidir el.

La función de comparación podría incluirse dentro del mismo módulo impulsímetro, agregando los controles de nivel de comparación en el frente del mismo, o si fuesen constantes, incluirlos en el código. La variable a comparar sería el flujo promedio calculado, que es una variable en 21 bits que cubre todo el rango de medición. La comparación de variables binarias tiene mejor precisión y gran estabilidad frente a derivas térmicas y por envejecimiento, que sí afectan a los componentes que determinan la precisión en los circuitos analógicos. Esta forma de trabajo además elimina la necesidad de obtener el valor del flujo logarítmico.

Este reemplazo requiere un análisis de seguridad que va más allá del tema del presente trabajo, pero se deja constancia de que el impulsímetro digital configurado como medidor de flujo o de tasa de variación, puede generar las señales de disparo por alto y por bajo para el flujo lineal y para la tasa de variación.

7. Referencias bibliográficas

- [1] "Radiation Detection and Measurement", Glenn Knoll, third edition.
- [2] "Manual de Fórmulas y Tablas matemáticas". Schaum
- [3] Walpole, Myers. Cuarta Edición. "Probabilidad y Estadística".
- [4] Hunter J.S., (1986), "The exponentially weighted moving average", Journal of Quality Technology. 18, 155-162.
- [5] "Señales y Sistemas". Alan V Oppenheim, Alan S. Willsky.
- [6] Ferrucci, F.N.; Verrastro, C.A.; Rios, G.E.; Estryk, D.S.; , "FPGA-based random pulse generator for emulation of a neutron detector system in a nuclear reactor," *Programmable Logic (SPL)*, 2011 VII Southern Conference on , vol., no., pp.103-108, 13-15 April 2011. doi: 10.1109/SPL.2011.5782633

8. Anexos

- I. Diagrama de operaciones para cálculo de flujo.

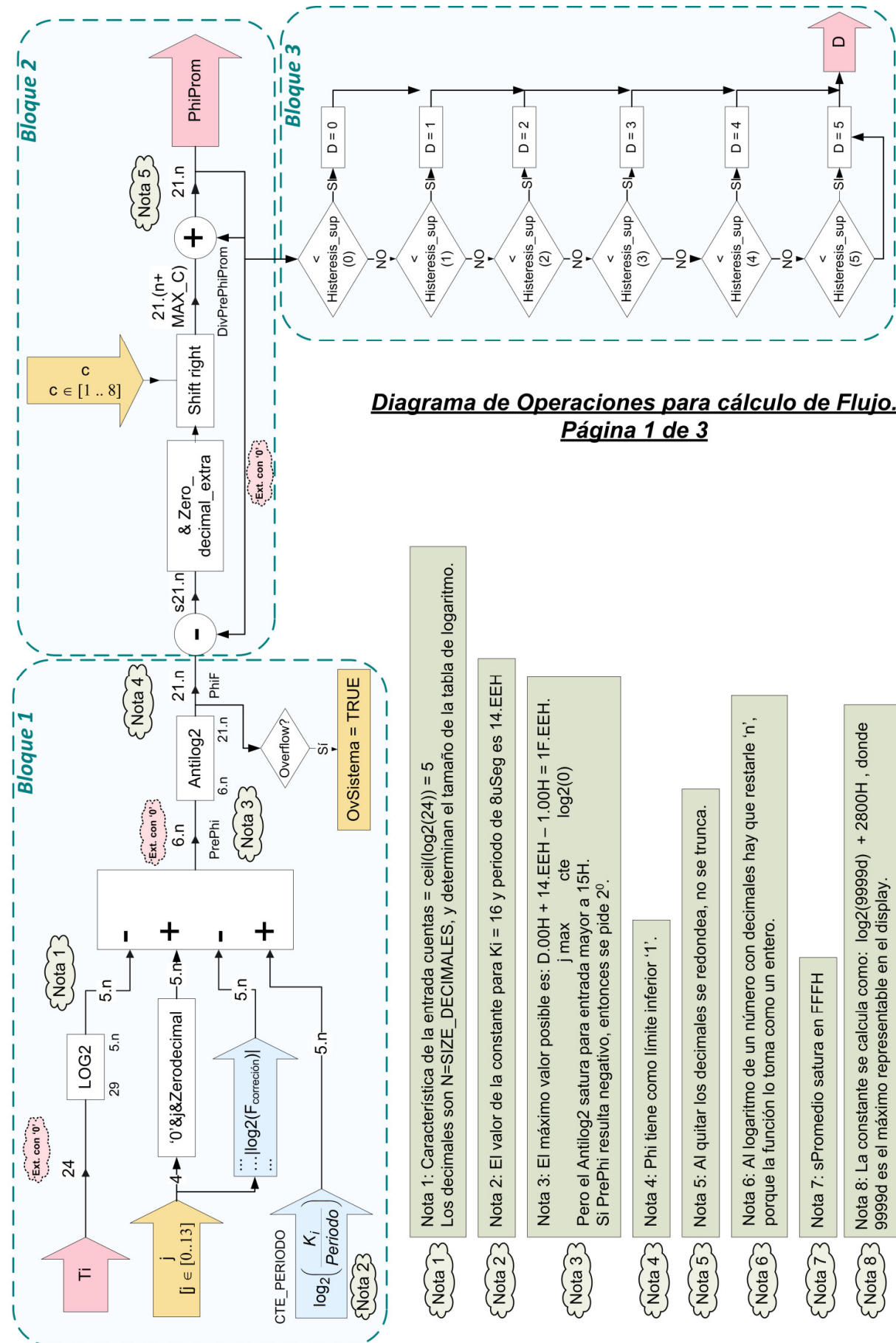


Diagrama de Operaciones para cálculo de Flujo. Página 2 de 3

