



Probits: Interfaz Gráfica para la Adquisición y Procesamiento de Señales Biomédicas con la Tarjeta Bitalino.

V. Ramírez Medina¹, G. Dorantes-Méndez¹

¹ Facultad de Ciencias, Universidad Autónoma de San Luis Potosí, México.

Resumen— Este artículo describe el diseño de Probits, una interfaz gráfica de usuario (GUI) para la adquisición de señales biomédicas con la tarjeta Bitalino®. Ésta es una tarjeta de adquisición inalámbrica de bajo costo, que cuenta con varios tipos de sensores, entre ellos de electrocardiografía, electromiografía y electroencefalografía. Bitalino® cuenta con su software de adquisición, Open Signals. Sin embargo, es limitado, por lo que la interfaz Probits añade las características de autoescala de las señales y colocación de marcadores para indicar eventos durante la adquisición. Además, incluye opciones de procesamiento como la de calcular la transformada de Fourier y la densidad espectral de potencia, utilizadas para análisis de diversos tipos de señales. Adicionalmente, incluye opciones para el análisis de variabilidad de la frecuencia cardíaca y de electromiografía. Probits está desarrollado en el lenguaje Python, utilizando Qt para el diseño de la GUI y PyQtgraph, que se enfoca en la visualización de señales e imágenes en tiempo real. Todas estas herramientas son de software libre y sencillas de utilizar, con la intención de que Probits sea de distribución libre y fácil adaptación a distintos tipos de investigación.

Palabras clave— Interfaz gráfica de usuario, adquisición de señales, procesamiento de señales.

I. INTRODUCCIÓN

El estudio de señales biomédicas es un área que requiere el uso de equipo de adquisición y software de procesamiento. Existen equipos de adquisición de señales sofisticados como Biopac® o ADInstruments®, que tienen la capacidad de adquirir diferentes señales y poseen diferentes características de adquisición e incluso software de procesamiento. Sin embargo, tienen un costo elevado. Un sistema de adquisición alternativo de bajo costo es Bitalino®, que cuenta con sensores de: electrocardiografía (ECG), electromiografía (EMG), electroencefalografía (EEG), acelerometría, fotoplestismografía (PPG), temperatura, respiración y actividad electrodermal. Bitalino® cuenta con un procesador ATMEGA328P, y funciona de manera inalámbrica gracias a su módulo Bluetooth. Además, cuenta con un software de adquisición gratuito, llamado Open Signals. Este software se encarga de la visualización de señales durante una adquisición; sin embargo, carece de las características de autoescala de la señal y colocación de marcadores para indicar eventos propios del protocolo de adquisición. Uno de los objetivos de este proyecto es diseñar una interfaz gráfica de usuario (GUI), de distribución libre, que realice adquisiciones incluyendo estas características.

Frecuentemente, el procesamiento de señales biomédicas se hace fuera de línea, utilizando software de programación como Matlab®. Aunque Bitalino® ofrece software diseñado para este tipo de procesamiento, tiene un costo. Por esto, se

buscó diseñar una interfaz que cuente, además de la opción de realizar adquisiciones, con opciones de procesamiento, como el análisis de variabilidad de la frecuencia cardíaca (VFC), que se realiza con series de tiempo de intervalos RR, tomados a partir de señales de ECG. Otra opción es el análisis de la fatiga durante contracciones en señales de EMG. Estos tipos de análisis se realizan obteniendo índices en el dominio del tiempo y de la frecuencia, por lo que también debe incluir un método de transformación al dominio de la frecuencia, como lo es la transformada de Fourier y la densidad espectral de potencia (DEP). Estos métodos pueden ser utilizados para el análisis de otros tipos de señales, no solo de los mencionados.

Para esto, se diseñó una GUI llamada Probits. Esta es una herramienta de software libre, ya que está diseñada con Python, que es un lenguaje sencillo de comprender, lo que facilita su adaptación a distintos tipos de investigación. Además, utiliza Qt, una herramienta para el diseño de cualquier tipo de GUI's, y PyQtgraph, librería que se ha utilizado para la visualización de señales en tiempo real [1].

II. METODOLOGÍA

El desarrollo de Probits estuvo basado en la API de Python creada para la tarjeta Bitalino® [2]. Además, para el diseño de las ventanas se utilizó Qt Designer. Probits contiene tareas fundamentales como: la adquisición de señales en tiempo real, la opción de abrir y visualizar archivos de datos, y varias opciones de procesamiento disponibles para las señales.

1) *Adquisición de señales en tiempo real*: Para lograr una visualización de los datos en tiempo real, se creó un ciclo llamado **update**, dentro del cual se lleva a cabo la lectura y graficación de los datos de la tarjeta Bitalino®, utilizando una estructura FIFO, encargada de desplazar la señal. La lectura de los datos se realiza en paquetes, cuyo tamaño se definió equivalente a 1 segundo de muestras. Por ejemplo, un paquete de 1000 muestras corresponde a una frecuencia de muestreo (Fm) de 1000 Hz. Esta configuración es utilizada para todas las Fm mayores a 10 Hz disponibles en Bitalino®.

Posteriormente, los paquetes son almacenados de forma acumulada en un vector, y después, para la graficación, se realiza un desplazamiento y actualización de los datos por sub-paquetes, cuyo tamaño corresponde a 1/10 de segundo. Por ejemplo, a una Fm de 1000 Hz, la señal es desplazada por sub-paquetes de 100 muestras, y la gráfica es actualizada en cada desplazamiento. Además, con el fin de crear un vector de tiempo que corresponda al eje x de la graficación, se incluye dentro del ciclo una lectura del tiempo desde la computadora, con el comando `pyqtgraph.time()`, y se genera

un vector a partir de él, que es desplazado con la misma estructura FIFO. Otra característica del ciclo es la de colocar marcadores con el objetivo de indicar eventos durante la adquisición; para esto, el usuario debe teclear F9. Estos marcadores y los paquetes de datos son almacenados en un vector y al terminar la adquisición son almacenados en un archivo de texto.

Para iniciar el ciclo **update** en la GUI, el usuario debe dar clic en el botón 'Empezar', dentro de la opción 'Nueva Adquisición'. Este botón utiliza **signals** y **slots**, una herramienta de Qt creada para implementar la comunicación entre objetos de manera sencilla [3]. En este caso el botón envía la señal a la tarjeta Bitalino® para el envío de datos, y a la ventana para la visualización de los canales elegidos.

Para detener el ciclo, se da clic en el botón 'Detener', que detiene la comunicación con Bitalino®, y almacena los datos y los marcadores en su archivo de texto. La Fig. 1 muestra un diagrama con las características de este ciclo.

2) *Opción de abrir y visualizar un archivo de datos*: Si el usuario desea visualizar una señal almacenada en un archivo txt, Probits tiene la opción de desplegarlo. Esta visualización incluye las características de zoom en los dos ejes y autoescala, que son propias de la librería PyQtGraph. Una vez abierta una señal desde un archivo, esta puede ser procesada y el resultado es desplegado en otra ventana. El diseño de esta GUI permite desplegar múltiples ventanas de procesamiento a partir de un archivo de datos.

3) *Procesamiento de señales*: Probits cuenta con las opciones de procesamiento general de la transformada de Fourier y la DEP, y con opciones específicas para el análisis de la VFC y de señales de EMG. Todas estas son descritas a continuación.

Transformada de Fourier, usada para analizar los componentes en frecuencia de varios tipos de señales fisiológicas. Se utiliza el comando de Python **numpy.fft.fft()**

Densidad espectral de potencia (DEP), calculada a partir del método de Welch, por medio del comando **scipy.welch()**. Este método consiste en dividir la señal en varios segmentos traslapados entre sí, de los cuales se calcula su periodograma, definido por (1):

$$DEP(f) = \frac{\Delta t}{N} \left| \sum_{n=0}^{N-1} x(n) e^{-j2\pi f n} \right|^2 \quad (1)$$

$$-1/2\Delta t < f \leq 1/2\Delta t$$

donde x es la señal, Δt el periodo de muestreo y N la longitud de la señal. Estos periodogramas son multiplicados por una ventana Hamming, y promediados, lo que genera una DEP suavizada [4].

Variabilidad de la frecuencia cardiaca (VFC), las fluctuaciones de frecuencia cardiaca (FC) se evalúan utilizando la serie de tiempo de intervalos RR, correspondientes a la diferencia en tiempo entre las posiciones de la onda R, o a través de la serie de tiempo de intervalos NN, la cual contempla solo latidos ocasionados por despolarizaciones originadas en el nodo sinoauricular [5].



Fig. 1. Diagrama del ciclo llevado a cabo por Probits durante una adquisición.

- Los índices en el dominio del tiempo de VFC son:
 - ✓ Media y desviación estándar de la señal de intervalos RR y NN.
 - ✓ Media y desviación estándar de la FC, calculada como el inverso de la serie de tiempo de intervalos RR y multiplicada por 60.
 - ✓ RMSSD, definido en (2).

$$RMSSD = \sqrt{\frac{1}{N-1} (\sum_{i=1}^{N-1} RR_{i+1} - RR_i)^2} \quad (2)$$
 donde N es la longitud de la señal.
 - ✓ NN50 y porcentaje de NN50, es el número de intervalos NN que excede 50 ms y su porcentaje con respecto a la serie de intervalos NN.
 - ✓ Gráfica de Poincaré, es una gráfica de la distribución de puntos de la señal de intervalos RR. Consiste en graficar la serie de datos (RR_i) , en el eje x , contra la serie adelantada un punto (RR_{i+1}) , en el eje y [6]. Posteriormente, esta gráfica es aproximada a una elipse, en la que son calculadas las desviaciones estándar de sus ejes transversal (SD1) y longitudinal (SD2) mediante (3) y (4)

$$SD1 = \frac{\sqrt{2}}{2} SD(x_n - x_{n+1}) \quad (3)$$

$$SD2 = \sqrt{2SD(x_n)^2 - \frac{1}{2}SD(x_n - x_{n+1})^2} \quad (4)$$

donde x representa los datos de la señal de intervalos RR y SD la desviación estándar de los datos.

- Para obtener índices en el dominio de la frecuencia, se calcula la DEP de la señal de intervalos RR y a partir de la DEP se calcula:

- ✓ La potencia total, definida como el área bajo la curva de la DEP en el intervalo de 0-0.5 Hz.
- ✓ El área de la DEP en intervalos específicos definiendo los índices: VLF (muy baja frecuencia, 0-0.04Hz), LF (baja frecuencia, 0.04-0.15 Hz), y HF (alta frecuencia, 0.15-0.4 Hz).

Análisis para EMG, se lleva a cabo para cuantificar y caracterizar cambios en las contracciones musculares [7]. Para esto, Probits incluye las opciones descritas a continuación.

- EMG integrado, integra una señal de EMG para observar cambios correspondientes a la contracción y relajación de un músculo. Antes de integrar la señal, se le resta su valor medio y se rectifica.

- Índices en el dominio del tiempo:

- ✓ Valor cuadrático medio: $RMS = \sqrt{\frac{1}{n} \sum_{i=1}^N x_i^2}$

- ✓ Integral: $I = \sum_{i=1}^N x_i$

- ✓ Integral cuadrada simple: $ICS = \sum_{i=1}^N x_i^2$

- ✓ Integral del valor absoluto: $IVA = \frac{1}{N} \sum_{i=1}^N |x_i|$

en los que N corresponde a la longitud de la señal.

- Índices en el dominio de la frecuencia:

- ✓ Potencia total, el área bajo la curva de la DEP, definida en (5).

$$PT = \sum_{i=1}^M DEP_i \quad (5)$$

- ✓ Frecuencia mediana, calcula el valor de frecuencia en el cual la DEP está dividida a la mitad. Este índice es muy utilizado para la evaluación de la fatiga muscular y es definido por (6).

$$\sum_{i=1}^{MDF} DEP_j = \sum_{j=MDF}^M DEP_j = \frac{1}{2} \sum_{j=1}^M DEP_j \quad (6)$$

donde M es la longitud de la DEP, y MDF es la frecuencia mediana.

III. RESULTADOS Y DISCUSIÓN

Probits fue desarrollado dentro del sistema operativo Ubuntu, versión 14.04 LTS. El equipo de cómputo utilizado para ejecutar esta GUI es una laptop con procesador Intel® Core™ i5. Se probó con tarjetas de Bitalino® y de su versión más reciente, Bitalino® (r)evolution.

A. Prueba de Adquisición

Probits realiza adquisiciones con todas las frecuencias y canales disponibles para la tarjeta, incluyendo a 1 Hz, aunque para esa opción no se implementó un desplazamiento. La Fig. 2 muestra la ventana que se despliega al seleccionar la opción de realizar una nueva adquisición. En esta ventana se puede

seleccionar cualquier combinación de canales a adquirir, y alguna de las Fm disponibles.

Una vez introducida la información, comienza la adquisición y se despliegan las señales deseadas, como se muestra en la Fig. 3, en la que se realizó una adquisición de 6 canales a una Fm de 1000 Hz, con señales de ECG, respiración, PPG, EMG y de 2 señales de acelerometría. Esta adquisición se hizo por una hora, durante la cual no hubo pérdida de comunicación. Al final de este periodo, se observó un retraso en la visualización de 2 segundos. Además, se colocaron marcadores, y al abrir el archivo donde fueron almacenados se comprobó que concordaron con los tiempos de colocación. Adicionalmente, el tamaño del archivo donde se almacenó la señal coincidió con el tiempo de adquisición.

Durante la adquisición, puede modificarse la escala de las señales en el eje y por un rango definido, o puede usarse la opción de autoescala, por medio de un menú desplegable de PyQtGraph, que se activa al dar clic derecho en una gráfica.



Fig. 2. Interfaz Probits, donde se selecciona 'Nueva Adquisición' en la barra de herramientas y se despliega una ventana que pide al usuario información sobre los canales y Fm deseados.

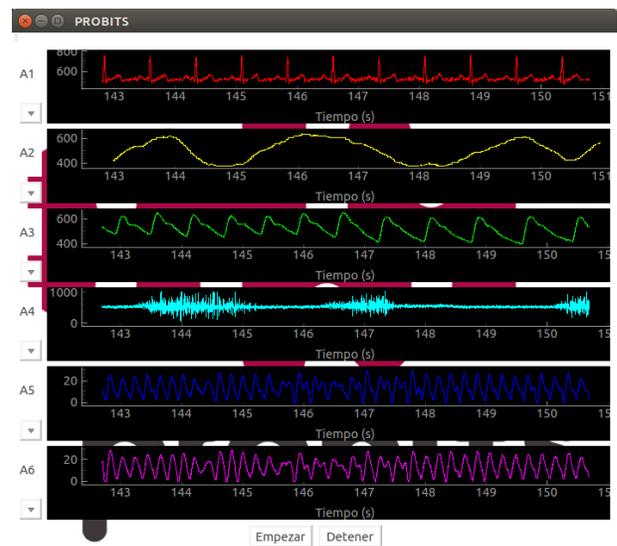


Fig. 3. Interfaz Probits durante una adquisición de las siguientes señales: A1) ECG, A2) respiración, A3) PPG, A4) EMG, A5) y A6) acelerometría.

B. Pruebas de Procesamiento

Por otro lado, en la Fig. 4 se observa un archivo de una señal de intervalos RR abierto en la GUI, en el que se elige como opción de procesamiento la DEP. En la parte superior derecha de la Fig. 4 se puede observar la ventana desplegada que pide al usuario el tiempo inicial y final, la Fm de la señal, los canales a procesar, y la opción de desplegar índices en frecuencia para ECG o para EMG. Estos datos fueron introducidos como se observa en la Fig. 4, y los resultados se despliegan en otra ventana, mostrada en la parte inferior de la misma figura, donde se puede observar la DEP con índices de VFC en el dominio de la frecuencia de las series de tiempo de intervalos RR.

En la Fig. 5 se observa una señal de EMG en la que se muestran 4 contracciones del músculo flexor de la muñeca. Para esta señal se elige la opción de EMG integrado, y su procesamiento se puede observar en la parte inferior de la figura.

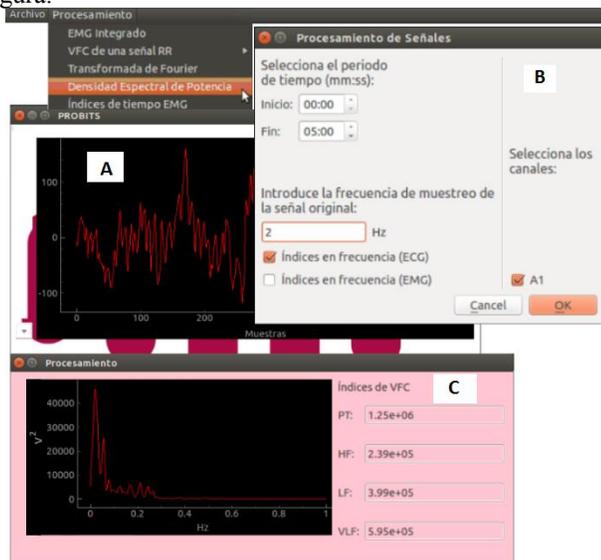


Fig. 4. Señal RR desplegada en Probits (A), para la que se elige la DEP como opción de procesamiento. Se despliega una ventana que pide datos del procesamiento al usuario (B). Una vez introducidos, la DEP de la señal es desplegada en otra ventana (C).

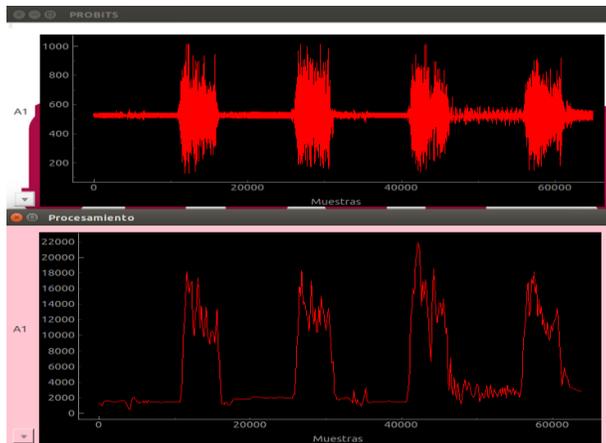


Fig. 5. Señal de EMG en la que se realizó una serie de contracciones (parte superior), y señal de EMG integrada (parte inferior).

Las demás opciones de procesamiento de Probits pueden ser utilizadas de manera similar a los ejemplos de las Fig. 4 y 5. Las señales procesadas tienen la opción de almacenarse como datos en formato CSV o como imágenes de tipo PNG. Esta opción es parte del diseño de PyQtGraph. Existen otras opciones de esta librería que no se exploraron en el diseño de Probits, y que dan pie a la implementación de más opciones de procesamiento para el análisis de diferentes tipos de señales biomédicas. Un ejemplo de ello es que PyQtGraph cuenta con la opción de calcular la transformada de Fourier de una señal a partir de su menú desplegable. Esto permite, por ejemplo, ver el espectro en frecuencia de una señal en tiempo real, durante una adquisición.

IV. CONCLUSIÓN

La GUI Probits, de distribución libre, aporta las características de autoescala y eje del tiempo para la visualización de señales biomédicas. Además, incluye la opción de colocar marcadores durante la adquisición, y varias opciones de procesamiento para el análisis de señales. Los resultados obtenidos en estos rubros son satisfactorios, por lo que es un software que ofrece ventajas sobre el software existente para Bitalino®, Open Signals. Las herramientas de VFC e índices de EMG de esta GUI pueden ser utilizados para protocolos donde se hacen variaciones de señales cardiovasculares y para la evaluación de la señal de EMG. Además, Probits tiene el potencial para incluir características más avanzadas, como procesamiento en tiempo real y más opciones de procesamiento, como las incluidas en la librería de Python BioSPPy. Sin embargo, implementar más características en tiempo real implica una mayor carga de procesamiento, por lo que se requiere cambiar Probits a una arquitectura de multiprocesamiento.

BIBLIOGRAFÍA

- [1] S. Sepúlveda, P. Reyes y A. Weinstein. "Visualizing Physiological Signals in Real Time," in *Proc. of the 14th Python in Science Conf: SCIPY 2015*, Austin, TX, pp. 190-194.
- [2] H. da Silva, A. Lourenço, A. Fred y R. Martins, "BIT: Biosignal Igniter Toolkit," *Computer Methods and Programs in Biomedicine*, vol. 115, no. 1, pp.20-32, 2014.
- [3] M. Summerfield. *Rapid GUI Programming with Python and Qt: the definitive guide to PyQt programming*, Prentice Hall, 2008.
- [4] O. M. Solomon, Jr., "PSD computations using Welch's method", Sandia National Laboratories., ABQ, NM, SAND91-1533, 1991.
- [5] Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology, "Heart rate variability: standards of measurement, physiological interpretation and clinical use", *European Heart Journal* vol. 17, no. 3, pp. 354-381, 1996.
- [6] M. B. Tayel y E. Al Saba, "Poincaré Plot for Heart Rate Variability", *International Journal of Medical Health, Biomedical, Bioengineering and Pharmaceutical Engineering.*, vol. 9, no. 9, pp. 708-711, 2015.
- [7] A. Phinyomark, C. Limsakul y P. Phukpattaranont "A novel feature extraction for robust EMG pattern recognition," *Journal of Computing*, vol.1, no. 1, pp. 71-80, 2009.