MATERIA DE SISTEMAS MICROPROCESADOS LABORATORIO PUERTOS ENTRADA-SALIDA

Wilmer D. Farinango-Tallana

15 de diciembre de 2020

1. Introducción

Se busca implementar los criterios de filtros FIR y algoritmos de suavizado de la señal para determinar cuál de ellos es el adecuado. Todo este proceso es mediante la programación dentro del entorno de Arduino y comparar el rendimiento por medio de la métrica relación señal ruido (SNR). Las señales son proporcionadas por el docente, en un bloc de notas.

Se escoge dos señales la primera señal es ECG (electrocardiograma), y la segunda señal será cualquier señal dada por el docente. La primera señales debe pasar por los dos algoritmos (convolución y suavizado), la segunda señal solo se codifica el suavizado y se determina cual es el mejor.

2. Diseño del Sistema

2.1. Diagrama de Flujo

Figura 1: Diagrama de flujo del sistema de suaviazado y convolucion

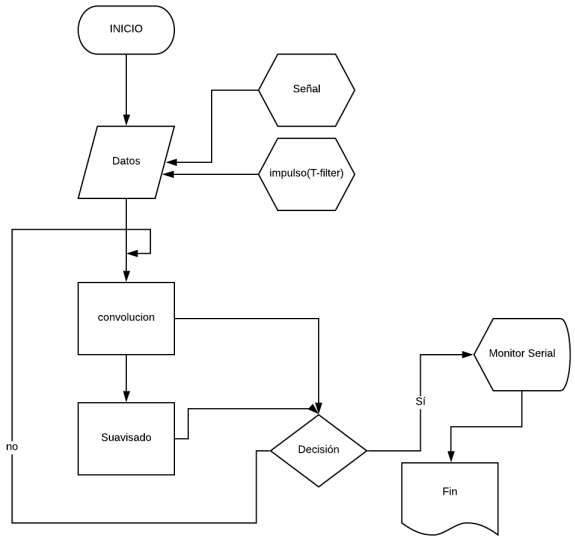
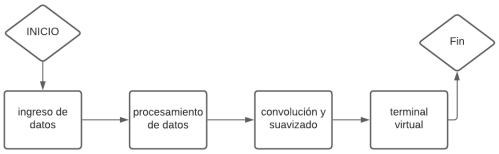


Figura 2: Diagrama de bloques del sistema



3. Desarrollo

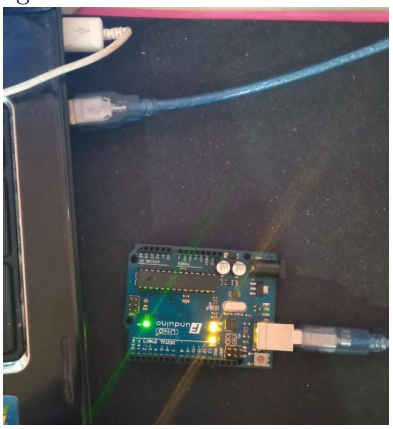
3.1. Simulación

Se observa el código a implementar para realizar el suavizado y el filtro.

```
#define tam_sig 88
#define tam_imp 23

// iportar se ales de una pesta a
extern double short_InputSignal_1kHz_15kHz[tam_sig];
extern double Impulse_response[tam_imp];

double output[tam_sig+tam_imp]; // vector de salida
double output_signal[tam_sig];

void convolution(double *sig_in, double *sig_out, double *imp, int sig_tam, int imp_tam);
void moving_average(double *sig_in, double *output_signal, int sig_tam, int filter);
void plot_signal(void);

void setup() {
Serial.begin(9600);
}

void loop() {
  convolution((double *)&short_InputSignal_1kHz_15kHz[0], (double *)&output[0], (double *)&
                , (int) tam_sig, (int) tam_imp);
  moving_average((double *)&short_InputSignal_1kHz_15kHz[0], (double *)&output_signal[0],(
            plot_signal();
            delay(100);
}
 ////////////// FILTROS FIR ////////////////////
void convolution(double *sig_in, double *sig_out, double *imp, int sig_tam, int imp_tam){
  int i,j;
  //ceros en el vector de salida
  for(i=0;i<(sig_tam+imp_tam);i++){
    sig_out[i]=0;
    }
    for(i=0;i<sig_tam;i++){
      for(j=0;j<imp_tam;j++){
        sig_out[i+j]=sig_out[i+j]+sig_in[i]*imp[j];

        }
      }
  }
//////////////// SUAVIZADO DE LA SE AL ////////////////////////
void moving_average(double *sig_in, double *output_signal, int sig_tam, int filter ){
  int i,j;
  for(i=floor(filter/2);i<sig_tam-floor(filter/2)-1;i++){
    output_signal[i]=0;
    for(j=-floor(filter/2);j<floor(filter/2);j++){
      output_signal[i]=output_signal[i]+sig_in[i+j];
      }
      output_signal[i]=output_signal[i]/filter;
    }
  }
void plot_signal(void){
  int i;
  for(i=0;i<tam_sig;i++){
    Serial.print(short_InputSignal_1kHz_15kHz[i]+10);
    Serial.print(",");
```

```
    Serial.print(output_signal[i]+5);
    Serial.print(",");
    Serial.println(output[i]/10);
    delay(5);
    }
  }

double short_InputSignal_1kHz_15kHz[88]={
402,386,386,382,379,397,396,416,431,442,470,481,514,
530,535,553,546,556,552,541,545,524,
520,505,487,488,466,467,459,449,457,
443,453,450,444,456,443,454,452,450,
464,454,466,466,464,477,468,480,480,
477,489,479,491,487,483,493,484,498,
499,498,516,513,544,578,627,700,751,
816,854,871,887,863,846,802,749,705,
638,593,542,495,470,431,422,408,396,
404,395,413};
//filtro pasabajos de 1kHz
 double  Impulse_response[23] = {
  -6,  2,  19,  11,  -18,  -7,
  37,  8,  -92,  -93,  60,  163,
  60,  -93,  -92,  8,  37,  -7,
  -18,  11,  19,  2,  -6
};
```

Figura 3: Conexion real en arduino



4. Análisis de Resultados

- Graficas del ECG.

Figura 4: Señal OCG con ruido

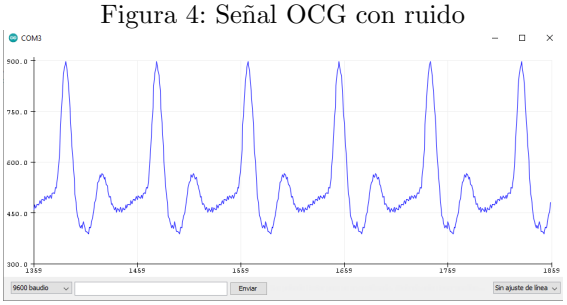


Figura 5: Señal OCG con Convolucion

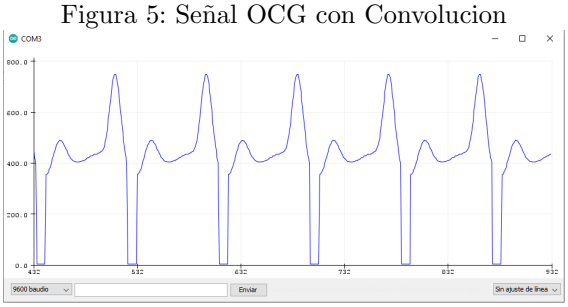


Figura 6: Señal OCG Suavizado conTfilter

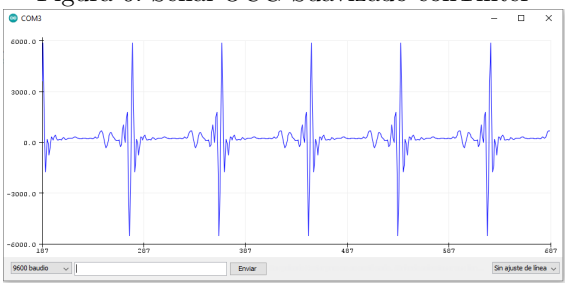


Figura 7: Filtro Pasabandas

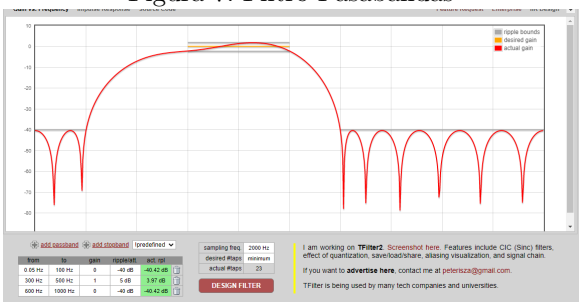
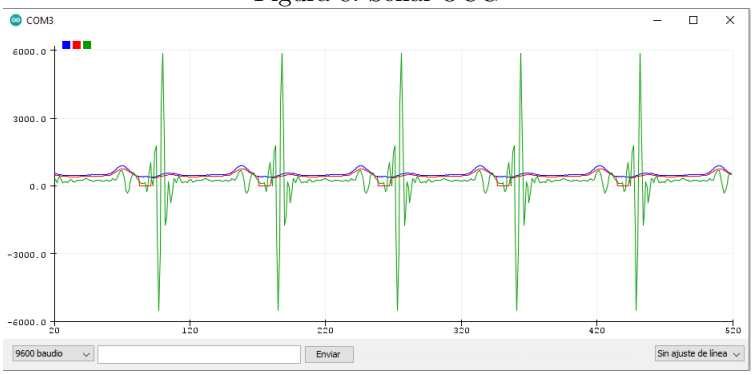


Figura 8: Señal OCG



- Señal Elegida por el estudiante, flexsensoraco.

Figura 9: Señal Flex Ruido

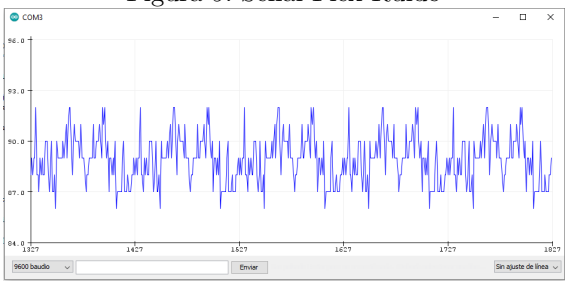


Figura 11: Señal Flex suavizado

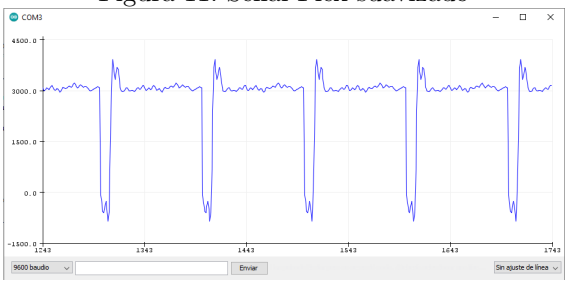


Figura 10: Señal Flex, convolucion.

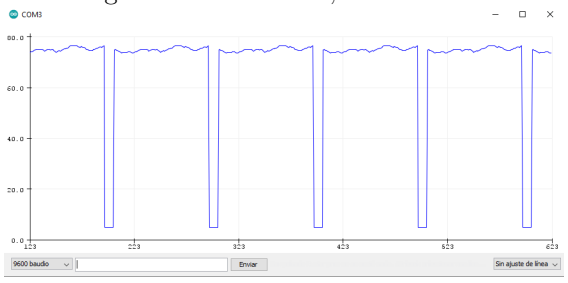


Figura 12: Filtro paso-bajo

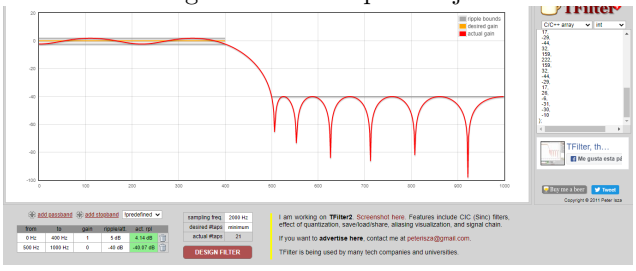


Figura 13: Señal Flex

5. Conclusiones

- Para el filtrado de la señal ECG, se comparó la respuesta en frecuencia de las funciones de transferencia de cada familia de los filtros y se observó que la función que nos brinda una ganancia lo mas plana posible.
- Para obtener la función de transferencia filtro pasa-banda a partir de los datos se debe obtener la siguiente una ecuacion cuadratica pasa-banda.
- ventaja de los filtros FIR es el hecho de que pueden producir fases lineales . Entonces, si una aplicación requiere fases lineales, la decisión es simple, se debe utilizar un filtro FIR

6. Recomendaciones

- Recordar que el analisis del filtro pasa banda detallado para las ECG son parametros la establecidos con frecuencia de corte.
- Analizar las entradas de los metodos y asu ves la dimesion de los arrays, la memoria diponible para evitar la saturacion.
- La estructura de programacion del filtro es necesario entender las salidas y las ubcaciones que recorre y guarda en cada vector.