

Medición Angular

Santiago Burgos Martínez¹, Edwin Javier Barbosa Molina², Daniel Esteban González Zuluaga³

Facultad de Ingeniería, Pontificia Universidad Javeriana.

Bogotá, Colombia.

santiagoburgos@javeriana.edu.co¹, edwin_barbosa@javeriana.edu.co², gonzalez_de@javeriana.edu.co³

Resumen—Este documento muestra la concepción, diseño, implementación y operación de un sistema capaz de medir posición angular por medio de un *Synchro/Resolver*. Además, se presenta lo fundamental de la teoría para entender el funcionamiento del dispositivo junto con los resultados obtenidos.

Abstract—This paper presents from scratch the design process, the workings and the way to do a system capable of measuring the angular position using a *Synchro/Resolver*. Furthermore it is presented the fundamental theory to understand the device along with the results obtained.

Palabras Clave: Ángulo, Medición, Posición, Scott-t, Puente de Wien, Filtro pasa bajos.

I. INTRODUCCIÓN

La medición angular es de alta importancia en la industria actual, debido a que por medio de esta se pueden controlar la posición y ubicación de diferentes dispositivos como los brazos robóticos y servomotores. Teniendo en cuenta todas las posibles aplicaciones de la medición angular, se desea crear un dispositivo el cual por medio de un *Synchro/Resolver* sea capaz de medir y determinar la posición angular.

Para llevar a cabo este proyecto, se utiliza una metodología de diseño tipo CDIO (Concebir, Diseñar, Implementar, Operar), con la cual se busca llevar a cabo y de la mejor forma los objetivos propuestos.

II. OBJETIVOS

A. Objetivos Generales.

- Diseñar, construir e implementar un dispositivo de medición angular (*Synchro/Resolver*).

B. Objetivos Específicos.

- Diseñar, construir e implementar un oscilador “puente de wien”, que sea capaz de oscilar a una frecuencia de 400 Hz, con máxima excursión en su salida.
- Diseñar, construir e implementar un circuito *Scott-t*, un Multiplicador y un Filtro pasa bajos.
 - El circuito Scott-t, es para poder realizar la transformación de 3 señales provenientes del *Synchro* a dos señales, las cuales son

aproximaciones de una onda seno y coseno respectivamente[3].

- Implementar un circuito multiplicador, el cual realice la operación entre las señales seno y coseno provenientes del Scott-t.
- Realizar un Filtro pasa bajas a la señal proveniente del circuito multiplicador para obtener un valor DC.
- Detección del ángulo.
 - Acondicionar la señal proveniente del filtro pasa bajas para que esta pueda ser leída y analizada por un Arduino.
 - Realizar el código el cual determine la posición angular del rotor del *Synchro* haciendo uso de la señal descrita previamente, este código se realizará en el programa Arduino Software IDE[1].

III. PROCEDIMIENTO

A. Problema y Requerimientos.

Se plantea y se organiza un proyecto para la realización de un dispositivo que sea capaz de medir la posición angular del rotor de un *Synchro*, esté debe con fuentes de polarización de ± 5 V y tener la capacidad de operar en temperaturas de hasta 40°C.

B. Diseño.

Synchro: Es un dispositivo el cual consta de 4 bobinas, donde tres de ellas están conectadas en delta y la cuarta es aquella que gira junto al rotor del *Synchro* y es conectada a una señal sinusoidal de 400 Hz. Al girar esta cuarta bobina se produce un cambio de corriente (inducción magnética) en las otras tres bobinas, lo que a su vez produce a la salida 3 señales con diferente amplitud e igual frecuencia.[2]

Para la señal de 400 Hz se realiza un oscilador con la topología de puente de *Wien* con ganancia controlada. Figura 1.

Las señales y las respectivas ecuaciones que el *Synchro* entrega son:

$$S1 = V_{3-1} = K * \sin(\theta) * \sin(w * t) \quad (1)$$

$$S2 = V_{2-3} = K * \sin(\theta + 120) * \sin(w * t) \quad (2)$$

$$S3 = V_{1-2} = K * \sin(\theta - 120) * \sin(w * t) \quad (3)$$

donde de $K = 0.46$ Vp (oscilador).

Oscilador: Un oscilador es un circuito que al ser polarizado y que por causa del ruido blanco (presente en todas las frecuencias) en sus componentes pasivos produce una señal periódica armónica (Sinusoidal o cuadrática). La frecuencia de oscilación es controlada a través de los componentes pasivos que lo componen como lo son las resistencias, los capacitores y las bobinas. El integrado escogido para esta etapa es el LM318. [8]

A continuación, se observa el esquemático del oscilador puente de Wien con ganancia controlada. [10]

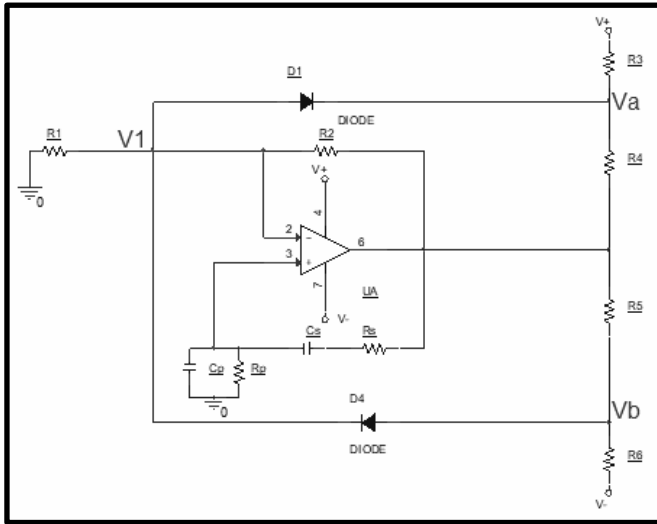


Figura 1. Topología escogida para realizar el oscilador de puente de Wien. [10]

Los valores de los componentes se hallan por medio de las siguientes fórmulas:

$$f_{osc} = \frac{1}{2\pi\sqrt{R_p R_s C_p C_s}} \quad (4)$$

$$\frac{R_2}{R_1} = 2 \quad (5)$$

resolviendo se tiene :

$$R_s = R_p = 3.9K$$

$$C_s = C_p = 100nF$$

Con estos componentes la $f_{osc} = 408.09$ Hz.

Para el cálculo de los componentes de la ganancia controlada se tienen las siguientes fórmulas:

$$Va = \frac{5 \cdot 2R_4}{2 \cdot R_4 + 2 \cdot R_6} \quad (6)$$

$$Vb = \frac{5 \cdot 2R_4}{2 \cdot R_4 + 2 \cdot R_6} \quad (7)$$

Donde los valores escogidos son:

$$R3=1K$$

$$R4=3.9K$$

Resolver: Es un dispositivo que convierte las 3 señales provenientes del Synchro (vistas anteriormente) en dos, una sinusoidal y otra cosenoidal aproximadamente. [3]

$$V_x = K * \sin(\theta) * \sin(w * t) \quad (8)$$

$$V_y = K * \sin(\theta) * \sin(w * t) \quad (9)$$

Transformador de Scott-t: Este circuito tiene como finalidad funcionar o hacer el papel del Resolver.

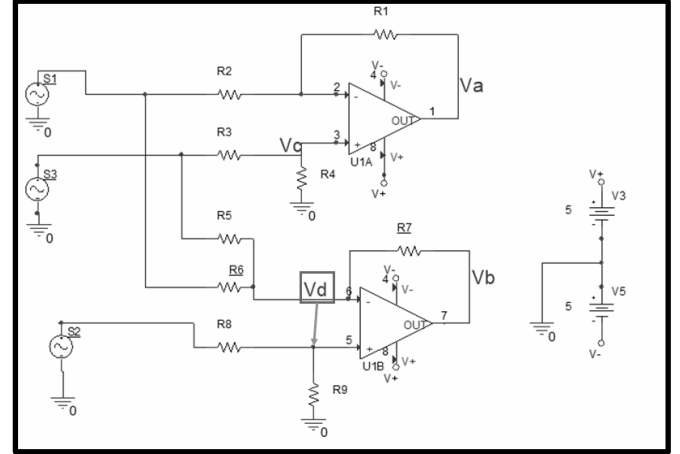


Figura 2. Esquemático del transformador Scott-T. [3]

Donde las ecuaciones de salida del dispositivo son:

$$Va = R_1 \left(-\frac{S_1}{R_2} + \frac{S_3 \cdot R_4}{R_3 + R_4} \cdot \frac{R_2 + R_1}{R_2} \right) \quad (10)$$

$$Vb = \frac{S_2 R_9}{R_9 + R_8} \cdot \frac{R_5 R_6 + R_7 R_6 + R_7 R_5}{R_5 R_6} - \frac{(S_3 R_5 + S_1 R_7) \cdot R_7}{R_5 \cdot R_6} \quad (11)$$

Las ecuaciones que se busca obtener son:

$$Va = S3 - S1 = \sin(\theta) \quad (12)$$

$$Vb = \frac{2}{\sqrt{3}} \left[S_2 - \frac{S_1 + S_3}{2} \right] = \cos(\theta) \quad (13)$$

Para lograr las ecuaciones anteriores se escogen los siguientes valores para las variables/componentes:

$$R_2 = R_3 = R_5 = R_6 = 10K \Omega$$

$$R_1 = R_4 = R_7 = R_9 = 10K \Omega$$

$$R_8 = 5.1K \Omega$$

Con estos datos se tiene que Va y Vb reales son :

$$Va = S3 - S1 \quad (14)$$

$$Vb = 2S2 - S1 - S3 \quad (15)$$

Ahora, reemplazando S3, S2 Y S1 por las ecuaciones (1) (2) (3) del Synchro se tiene :

$$Va = -\frac{3}{2} K * \sin(wt) \left[\frac{\sqrt{3}}{3} \cos(\theta) + \sin(\theta) \right] \quad (16)$$

$$Vb = \frac{3}{2} K * \sin(wt) \left[\sqrt{3} \cos(\theta) - \sin(\theta) \right] \quad (17)$$

Multiplicador: Este circuito, como su nombre lo dice, se encarga de multiplicar las dos señales de salida del Scott-t (señales idealmente iguales a las del Resolver) ecuaciones (8), (9) del apartado de Resolver. Para la realización del circuito multiplicador se usa el integrado

AD633ARZ [5] y un circuito elevador de voltaje que se encuentra en el datasheet [7] (el elevador se realiza debido a que para polarizar el integrado AD633ARZ se necesita como mínimo $\pm 8V$).

La ecuación característica del multiplicador es:

$$Vm = \frac{X_1 * Y_1}{10}, X_2 = 0, Y_2 = 0, Z = 0 \quad (18)$$

donde X_1 y Y_1 son las señales procedentes del *Scott-t*.

Idealmente se espera una salida del tipo:

$$Vm = \frac{K^2}{4*10} * \sin(2\theta) - \frac{K^2}{4*10} * \sin(2\theta) * \sin(2wt) \quad (19)$$

Parte DC Parte AC

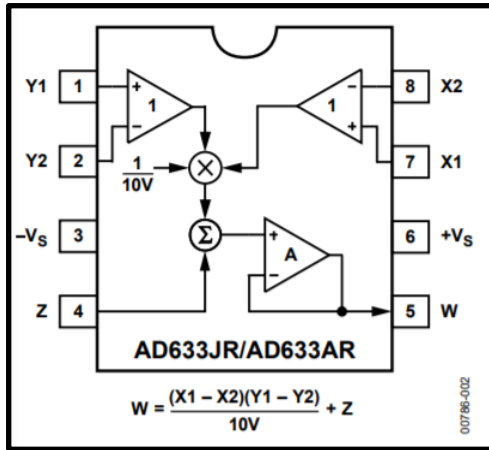


Figura 3. Esquemático interno y fórmula del multiplicador.[5]

Elevador de voltaje: para conseguir $\pm 8V$ o más se opta por usar el integrado LT1054, el cual en su datasheet proporciona una topología de elevador de voltaje para obtener $\pm 12V$ de $+5V$ [7]. El esquemático del circuito se observa en la Figura 4.

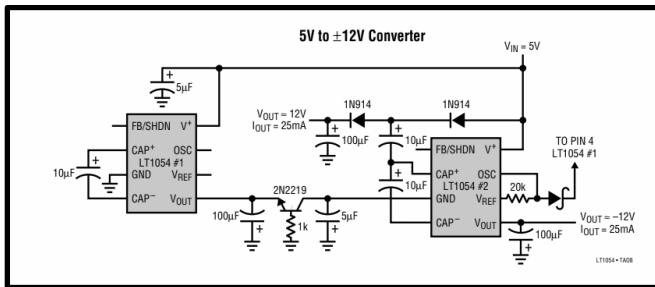


Figura 4. Esquemático elevador de voltaje.[7]

Filtro pasa bajas: Un filtro pasa bajas es un dispositivo que sirve para distintas aplicaciones. En esta ocasión se utiliza para poder eliminar todas las componentes AC de la señal y de esta forma obtener solo el valor DC. Para realizar esto, se busca construir un filtro que corte frecuencias superiores a 5 Hz logrando así quitar la parte AC de la señal (ecuación (19)) y adicional a esto se logra quitar ruido que se produce en altas frecuencias.

La topología escogida es *MFB - Chebyshev* [9]:

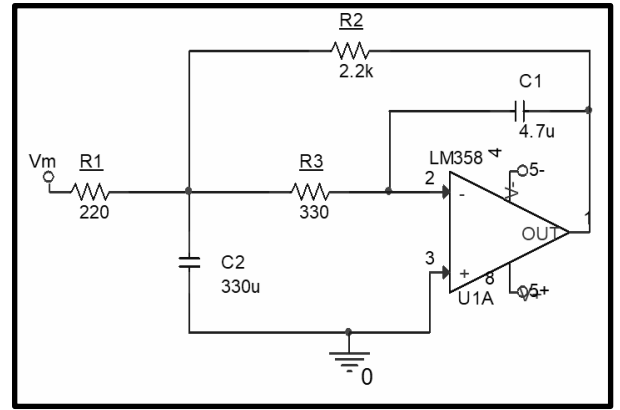


Figura 5. Filtro pasa bajas.

Las ecuaciones del filtro son:

$$\omega^2 = \frac{1}{C_1 C_2 R_2 R_3} \quad (20)$$

$$f = 4.74 \text{ Hz}$$

$$Av = \frac{R_2}{R_1} = 10V/V \quad (21)$$

La señal a la salida del filtro será de la forma:

$$Vm = \frac{K^2}{4} * \sin(2\theta) \quad (22)$$

La señal obtenida anteriormente (ecuación (22)) es la que se quiere analizar para obtener la posición angular del rotor del Synchro, para esto se opta por analizar esta señal por medio de un Arduino. Pero por características internas del Arduino, éste no puede recibir voltajes negativos, lo cual hace necesario pasar la señal por un circuito de acondicionamiento.

Acondicionamiento: Este circuito se realiza para adecuar la señal proveniente del filtro, obteniendo solo valores positivos y que estén en el rango de 0 V a 5 V los cuales el Arduino puede recibir. El esquemático consta de un divisor de voltaje conectado a un *buffer* que posteriormente se conecta a un sumador, y por último pasa por una etapa de ganancia con la cual se busca llegar a la máxima excursión de la entrada analógica del Arduino. Este acondicionamiento se realiza con el integrado LM358[6].

La función de transferencia para esta etapa es:

$$Vo = (V_{offset} + V_{filtro}) * Av \quad (23)$$

$$V_{filtro} = K \rightarrow 0.425V > K > -0.450V$$

$$Av = 3.3V/V$$

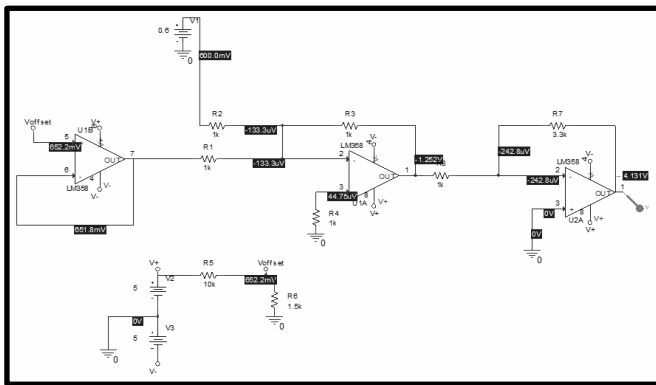


Figura 6. Esquemático circuito acondicionamiento de señal. “Orcad”

C. Implementación

PCB: Para la construcción y realización del PCB (PCB FR4 de 1.6mm de grosor, doble cara) se tiene en cuenta las recomendaciones de los fabricantes de cada componente que se va a usar, es decir, los fabricantes de los integrados en los datasheets proveen una guía indicando la forma y los componentes recomendados para su polarización y funcionamiento.

Por otro lado, la organización de los componentes del proyecto en el PCB sigue la premisa de colocar los circuitos relacionados con la polarización y alimentación de los componentes en la capa inferior y colocar los componentes relacionados con señal en la capa superior para así de esta forma garantizar que no habrá interferencia en las señales tratadas en el proyecto (**Ver Anexo 1, Esquemático del proyecto *Synchro/Resolver***).

Los componentes que se utilizan son de montaje superficial con tamaños SOIC 8 para los integrados, tamaño 1206 Y 0805 para las resistencias y capacitores, SOD123 para los diodos, SOT-23 para los transistores y tamaño DIP 8 para los elevadores de voltaje (Integrados LT1054).

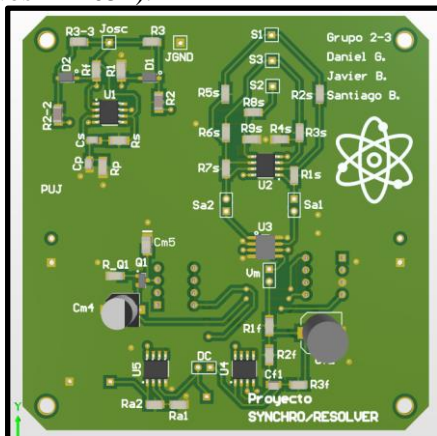


Figura 7. Layout del PCB con los circuitos descritos previamente.

ARDUINO MEGA2560: El código de Arduino debe poder identificar en qué cuadrante del círculo unitario se encuentra el rotor del Synchro, ya que de otra forma solo se estaría identificando alrededor de los 45° a -45° rango en el que funciona la detección del ángulo. El código se realiza en el software de programación Arduino Software IDE [1] en el cual la programación está basada en lenguaje C.

IV. RESULTADOS

A continuación, se muestran los resultados obtenidos en las distintas etapas del proyecto:

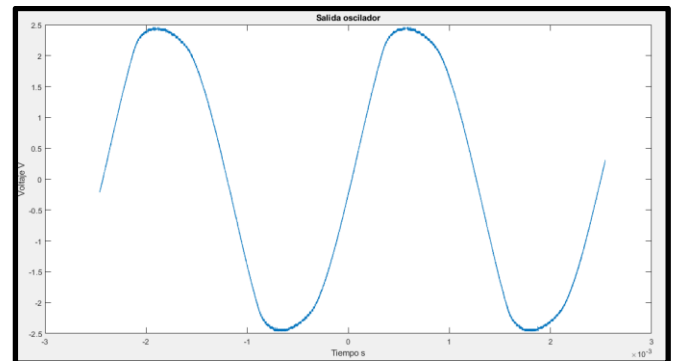


Figura 8. Onda seno / salida del oscilador.

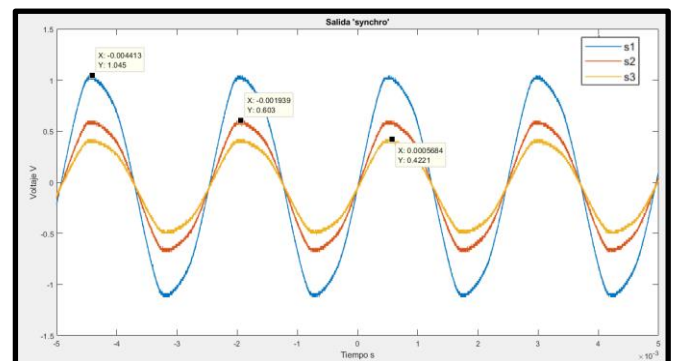


Figura 9. Salidas de la señal del Synchron.

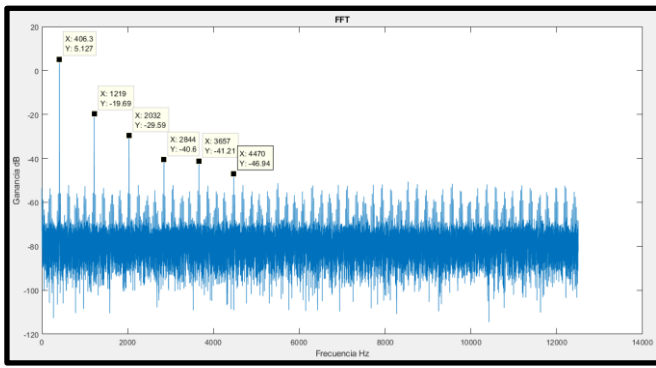


Figura 10. FFT obtenida de la implementación del oscilador.

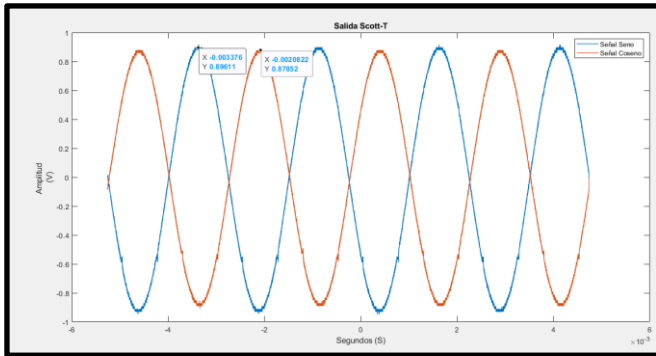


Figura 11. Señales seno y coseno en la salida del Scott-t.

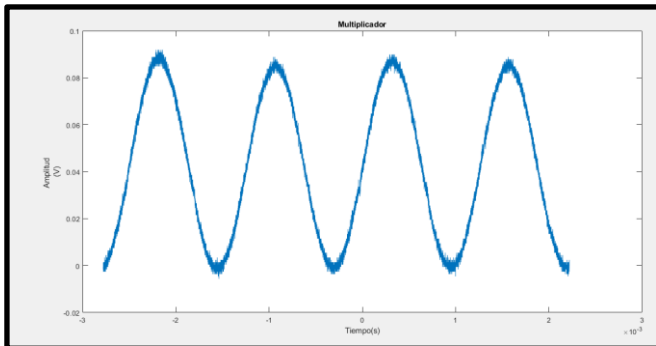


Figura 12. Señal a la salida del multiplicador

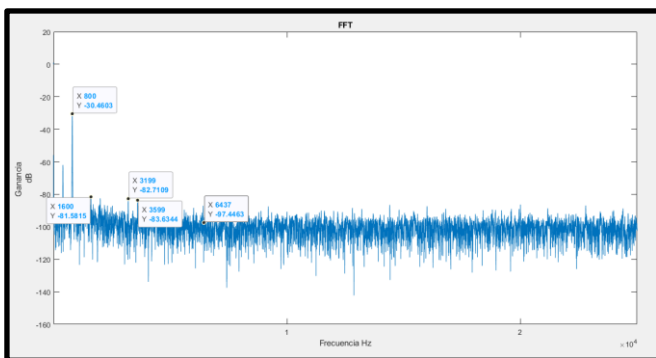


Figura 13. análisis de Fourier en la salida del multiplicador.

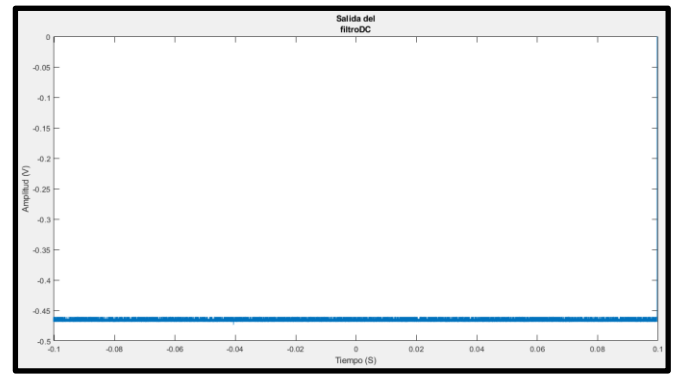


Figura 14. Señal DC a la salida del filtro pasa bajos.

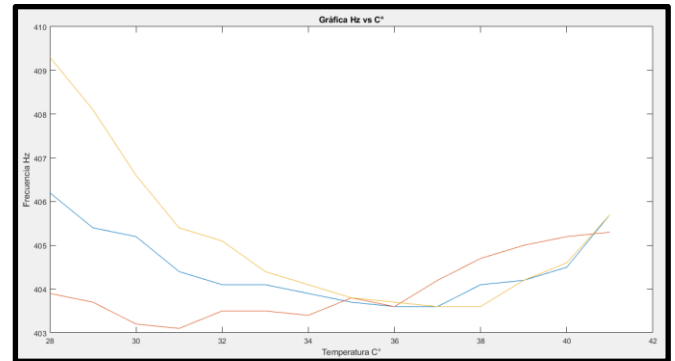


Figura 15. Dependencia de frecuencia oscilador según la temperatura.

Al realizar las pruebas de conexión y funcionamiento de la etapa del multiplicador se encontraron problemas con el elevador de voltaje, ya que no estaba cumpliendo su función como se esperaba (Eleva el voltaje de 5 V a 12 V). Se realizaron las pruebas correspondientes a los integrado LT1054 para comprobar su correcto funcionamiento y se encontró que uno estaba dañado, esto al comprobar el diodo que se encuentra entre los terminales 8 y 2, el cual en el caso del integrado dañado no se detectó el diodo. Debido a que no es factible cambiarlo teniendo en cuenta el cronograma del proyecto y a que el componente no tenía más inventario localmente.

Para dar solución al problema descrito previamente se decide modificar la polarización del circuito de $\pm 5V$ a $\pm 8V$. Este cambio afecta la amplitud de la señal seno del oscilador, mas no representa mayores cambios a este y al resto del proyecto.

Los datos del oscilador fueron:

$$V_{osc} = 3.93 V_p$$

$$F_{osc} = 397 \text{ Hz}$$

Posteriormente se conecta el oscilador al *Synchro* y este último se conecta a la etapa del *Scott-T* con los siguientes resultados:

$$V_{cos} = \pm X V_p$$

$$F = 393 \text{ Hz}$$

$$V_{sen} = \pm X V_p$$

$$F = 395 \text{ Hz}$$

A continuación, se realiza la medición del multiplicador con resultados:

$$V_m = -130 \text{ mV a } +140 \text{ mV}$$

$$F_{osc} \approx 792 \text{ Hz.}$$

Seguido se realiza la medición del filtro pasa bajas el cual tiene un voltaje de salida aproximado de:

$$V_{Filtro} \approx -1.25 \text{ a } +1.35 \text{ V}$$

Por último, se conecta la etapa de acondicionamiento la cual varía:

$$0 < V < 4.3 \text{ V}$$

Esta última señal se ingresa al Arduino Mega 2560, en donde a la señal se le quita la ganancia y dependencia de K, para así obtener una señal que varía de 0 a 5 V. Ahora, por medio de la librería matemática de Arduino, a la señal se le aplica el seno inverso y se divide sobre dos para obtener el ángulo, adicionalmente se codifica haciendo uso de contadores para así saber en qué segmento de 45° se está realizando la medida. Este resultado finalmente se muestra en una pantalla LCD de 16 x 2. Con un error aproximado de %10 (Dato Real Vs Dato observado) en la medición. Adicionalmente se observa que en el momento de cambiar de segmento se presenta una discontinuidad entre los datos reales y los datos que se observan en la LCD, es decir, el sistema se detiene y pierde la medida en algunas ocasiones.

V. CONCLUSIONES Y RECOMENDACIONES

- Se observa un funcionamiento lento y discontinuo en la correcta detección del ángulo cuando el rotor se aproxima a los 45° y sus múltiplos, además, de la falta de información acerca del cuadrante en el que se encuentra esté dentro del círculo unitario. Lo mencionado previamente se debe a la limitación que presenta la función seno inverso (Limitada entre -90° a 90° y que para este proyecto se observó una limitación de entre: -45° a 45°).

- Se logró obtener un funcionamiento estable del proyecto hasta la etapa del filtro pasa bajas, las etapas subsecuentes presentaron contratiempos y dificultades en cuanto a su correcto funcionamiento, la forma en que se implementó y a las restricciones impuestas por los componentes usados.

- Se recomienda buscar una forma más eficiente de detectar el ángulo (de forma analógica o digital) debido a las dificultades presentadas con la que se implementó en este proyecto, ya que a pesar de que se logra obtener el ángulo esté posea un error de aproximadamente 10% con respecto al teórico

esperado. Por último, cuando se cambia de segmento se tiene una precisión del $\pm 5\%$, precisión que no es útil en aplicaciones de alta exactitud.

VI. REFERENCIAS.

[1]Arduino(Nd).Arduino Mega. Retrieved from https://www.robotshop.com/media/files/pdf/arduino_mega2560datasheet.pdf

[2]Salz, K. (1998). How does synchros and resolvers work. (February), 1–4.

[3]Kessler, J. (1994). *Synchro/resolver conversion handbook*. (631). Retrieved from <http://www.ddc-web.com/documents/synhdbk.pdf>

[4]Synchro and Resolver Engineering Handbook. Retrieved from: <https://www.moog.com/literature/MCG/synchrobook.pdf>

[5]Datasheet del integrado AD633 <https://www.analog.com/media/en/technical-documentation/data-sheets/AD633.pdf>

[6]Datasheet del integrado LM358M <http://www.ti.com/lit/ds/symlink/lm2904-n.pdf>

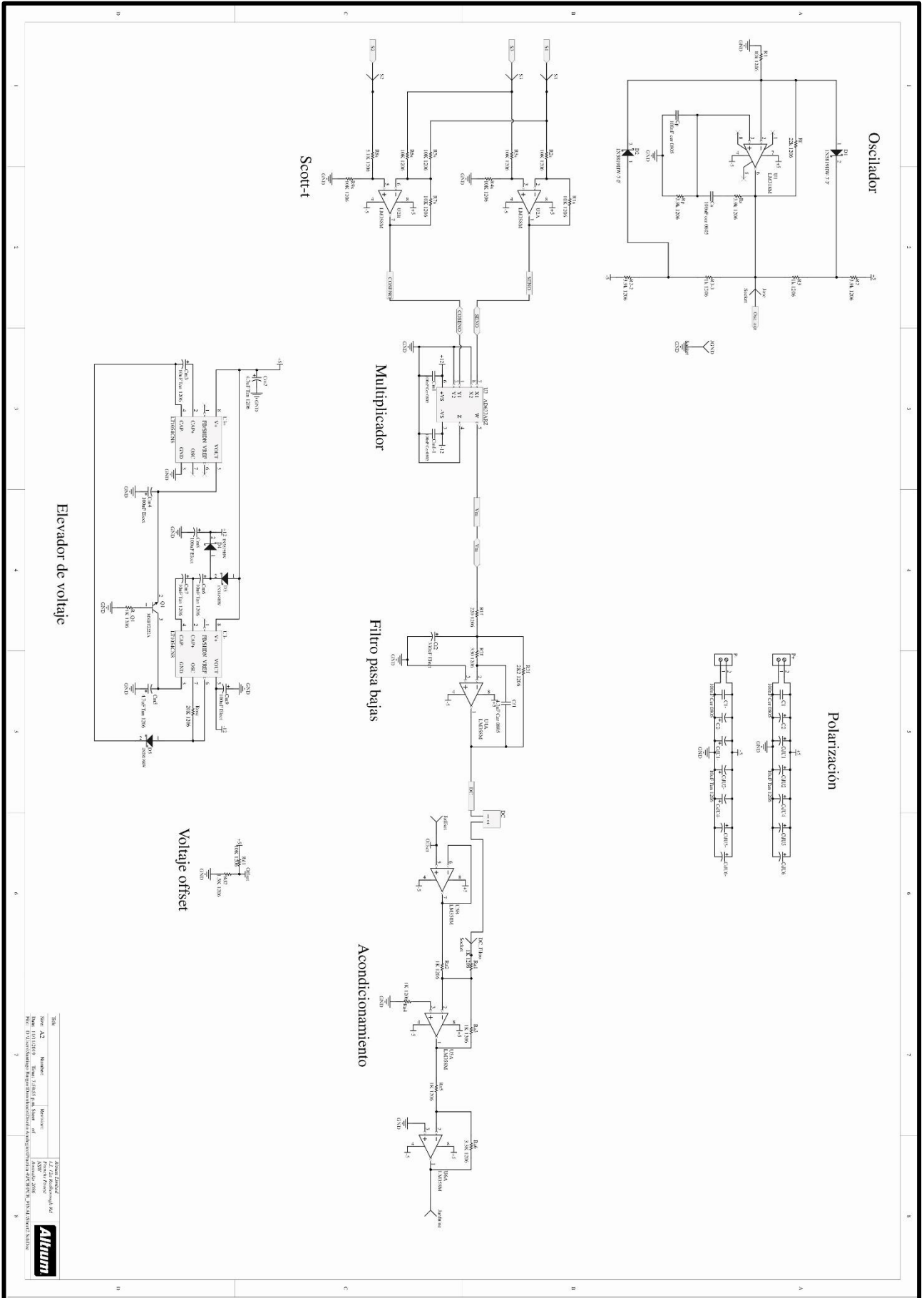
[7]Datasheet del integrado LT1054 <https://www.sigmaelectronics.net/manuals/LT1054.pdf>

[8]National Semiconductor (1998). Operational Amplifiers LM318. from. <https://www.sigmaelectronics.net/manuals/LM318.pdf>

[9]Nota de aplicación. *Active Low-Pass Filter Design*, de Texas Instruments. <http://www.ti.com/lit/an/sloa049b/sloa049b.pdf>.

[10]Sedra, A. S., & Smith, K. C. (2011). *Microelectronic circuits*. Oxford University Press.

Anexo 1. Esquemático del proyecto Synchro/Resolver.



Anexo 2. Código Arduino

```
#include <LiquidCrystal_I2C.h>
#include <math.h>

LiquidCrystal_I2C lcd(0x27,16,2);
int d=0;
float analogPin = A0;
float val = 0;
float a = 0;
float b = 0;
float c = 0;
float f=0;
float g=0;
float e = 0;
int cont = 0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  val = analogRead(analogPin); //Se lee el voltaje de salida del circuito de acondicionamiento.
  a = (val/205); // El valor leído por el Arduino se pasa a voltios.
  b = (a/1.52)-1.51; // Se quita de la ecuación la ganancia y el DC puesto en el acondicionamiento.
  b = b/1.255; // Se halló la constante K práctica para quitarla de la ecuación. Es ligeramente distinta a la teórica.
  if(b>1){
    b=1;
  }
  if(b<-1){
    b=-1;
  } // Estos 2 if acotan el resultado de b para que al arcoseno no le entre un resultado mayor a  $\pm 1$ .
  c = (asin(b)/2); // Se hace el último despeje de la ecuación para hallar el ángulo.
  c = (c*180)/M_PI; // Se pasa a grados el ángulo.
  if( -44.39<c && c<0 && cont==0){ // Se observa si el ángulo está en el primer cuadrante con el contador (0°-45°).
    lcd.init();
    lcd.backlight();
    lcd.clear();
    lcd.setCursor(2,0);
    lcd.print("El angulo es"); // Mensaje a despegar
    lcd.setCursor(5,1);
    lcd.print(-c+e); // Mensaje a despegar
    if(c < -42){
      cont = 1; // Se realiza el cambio de cuadrante.
    }
  }

  if( -44.39<c && c<0 && cont==1){ // Se observa si el ángulo está en el segundo cuadrante con el contador (45°-90°).
    lcd.init();
    lcd.backlight();
```



```

lcd.clear();
lcd.setCursor(2,0);
lcd.print("El angulo es"); // Mensaje a despegar
lcd.setCursor(5,1);
lcd.print(c+90+e); // Mensaje a despegar
    if(-0.1<c && c<0.1){
        cont = 2; // Se realiza el cambio de cuadrante.
    }
}
if( 0<c && c<44.39 && cont==2){ // Se observa si el ángulo está en el tercer cuadrante con el contador (90°-135°).
    lcd.init();
    lcd.backlight();
    lcd.clear();
    lcd.setCursor(2,0);
    lcd.print("El angulo es"); // Mensaje a despegar
    lcd.setCursor(5,1);
    lcd.print(90+c+e); // Mensaje a despegar
    if(30<c && c<45){
        cont=3; // Se realiza el cambio de cuadrante.
    }
}

if( 0<c && c<44.39 && cont==3){ // Se observa si el ángulo está en el tercer cuadrante con el contador (90°-135°).
    f = 45-c;
    g = 135+f+e;
    lcd.init();
    lcd.backlight();
    lcd.clear();
    lcd.setCursor(2,0);
    lcd.print("El angulo es"); // Mensaje a despegar
    lcd.setCursor(5,1);
    lcd.print(g); // Mensaje a despegar
    if(-0.1<c && c<0.1 && e==0){
        cont = 0; // Se realiza el cambio de cuadrante.
        e=180; // Se utiliza esta variable para repetir el proceso de identificación del ángulo, pero esta vez de (180°-360°).
    }
    if((g)>358.8) // Es el final del programa donde se reinicia el ángulo a 0° cuando llegue a 360°.
    {
        e=0;
        cont=0;
    }
}
}

```