

# Proyecto Final: Diseño e implementación de un servomotor para proyectos robóticos

Jose Fabio Navarro, Moisés Salas

josefabio1127@estudiantec.cr; moisosr17@estudiantec.cr

Área Académica de Ingeniería Mecatrónica

Instituto Tecnológico de Costa Rica

**Resumen**—En este escrito se describe el diseño e implementación de un servomotor, el cual fue diseñado con el objetivo de ser utilizado en aplicaciones de robótica a mediana y gran escala, motivo por el cual, se presenta un diseño con dimensiones optimizadas. En cuanto a la electrónica, dicho servomotor fue implementado con un controlador de velocidad ESC, y un potenciómetro de precisión para asegurar el control de posición. Asimismo, implementa el uso de un relé para el control de cambio de giro, un motor sin escobillas como fuente principal de movimiento y un microcontrolador para coordinar y controlar las funciones del mismo. Por último, en cuanto a la mecánica del dispositivo, este utiliza, entre otros componentes, un tren de engranes que permite obtener una velocidad y torque similar al de dispositivos análogos en el mercado.

Como resultado final, se obtuvo un dispositivo de 369 g de masa, y 502 cm<sup>3</sup> de volumen, que alcanza en condiciones nominales una velocidad de 46 rpm, y un torque de 1.07 Nm, en un rango de giro desde -360° hasta 360°.

**Palabras clave**—Servomotor, motor brushless, sensor de posición, sistema de reducción de velocidad.

## I. INTRODUCCIÓN

Este documento corresponde al informe del proyecto final del curso Sensores y Actuadores del segundo semestre del año 2022. En este proyecto se lleva a cabo el diseño e implementación de un Servomotor para proyectos robóticos.

Primero, un servomotor es un actuador rotativo, que de manera precisa permite obtener un control que no se puede obtener con un motor convencional. Dicho control, se obtiene en términos de variables como la posición angular, la velocidad e incluso la aceleración.

Por lo anterior, se podría decir que un servomotor, no es en sí un motor, sino un dispositivo de control de lazo cerrado, el cual tiene dentro de sus componentes, un motor que puede ser de corriente directa o altera, como fuente de movimiento rotacional.

Generalmente, los servomotores basan su control principalmente en las mediciones de velocidad y posición angular, esto debido a que se suelen usar motores de corriente alterna, sin embargo, para efectos de este proyecto, debido a que el motor dado es un motor brushless DC, no se va a implementar una medición de velocidad, ya que este tipo de motores generalmente se utilizan con un controlador de velocidad, que, mediante pulsos de PWM permiten el control de este parámetro. Por lo anterior, se utilizó únicamente como variable de control, la posición angular.

Ahora bien, para medir la posición angular, existen diversas opciones de sensores, en diversas categorías, como sensores magnéticos resistivos y ópticos, sin embargo, entre los que más destacan en estas categorías debido a su funcionalidad y fácil implementación son codificadores (encoders), y los potenciómetros.

Los codificadores, son sensores ópticos que permiten la medición de posición angular, mediante la detección de patrones en discos concéntricos al eje rotativo, sin embargo, estos dispositivos que suelen ser costosos para proyectos de muy bajo presupuesto. Por otra parte, los potenciómetros rotacionales permiten medir la posición en función de la variación de la resistencia, y debido a su bajo costo, son de más fácil acceso.

En cuanto al diseño del servomotor, a pesar de los componentes antes mencionados, en un servomotor, la sección mecánica es vital, ya que es la que asegura la velocidad y el par de torsión deseados a la salida, además de brindar soporte a todos los componentes involucrados en el sistema.

Para efectos de la reducción y transmisión de potencia del servomotor, se consideraron diversas opciones como el uso de bandas, o diversas configuraciones de engranes, principalmente en tren, en tornillo sin fin, y en configuración planetaria, sin embargo, debido a que se buscaba una reducción de aproximadamente 1:160, las bandas y los engranes planetarios se descartaron debido a que son configuraciones que abarcan una cantidad considerable de espacio, y este parámetro es uno de los que se desea optimizar.

Finalmente, se optó por una configuración de engrane con tornillo sin fin, unida a un tren de engranes rectos, esto principalmente por 2 factores. Primero, el tornillo sin fin permite una gran reducción en un volumen bastante pequeño, además de que utiliza engranes helicoidales que eliminan el ruido y reducen la fricción. Luego, la configuración de engrane con tornillo sin fin encontrada en el mercado, permitía un acople ideal al eje del motor DC que se estaba utilizando como fuente de movimiento rotacional, ya que implementaba el uso de un tornillo prisionero, con lo cual se solucionaba otro problema al mismo tiempo.

Luego de la configuración de tornillo sin fin, que proporcionaba una reducción de 1:25, se utilizó un tren de engranes rectos de 2 etapas, la primera de 1:2.2, y luego la etapa de salida de 1:2.8. Con esto finalmente se obtuvo una relación de salida 1:154, bastante cercana a la relación esperada. De igual manera, para el sostén de los ejes y demás componentes

rotatorios, se utilizaron rodamientos, los cuales de hecho, se utilizaron para dimensionar los ejes que se iban a utilizar, y con esto finalmente dimensionar los engranes.

En cuanto a los materiales, los engranes fueron confeccionados en ABS mediante impresión 3D, mientras que los ejes fueron maquinados en acero. De igual manera, se confeccionó una caja para almacenar y dar sostén a todos los componentes en acrílico, mediante corte láser, con algunas adiciones de piezas en impresión 3D.

Respecto a la parte electrónica y de control del diseño, se utilizó un controlador ESC-30A para manipular el giro del motor. Con dicho dispositivo, es posible modificar la velocidad al cambiar el ciclo de trabajo de las señales pwm que envía el controlador al motor. Además, fue necesario el uso de un microcontrolador, el cual se encarga de la parte lógica del diseño, mediante un código de programación. El microcontrolador empleado fue el Arduino UNO, el cual está basado en el microchip ATmega328P, y cuenta con un conjunto de pines tanto digitales como analógicos (ya sea de entrada o salida) que permiten el control de otros dispositivos necesarios para el funcionamiento del circuito. Esto se logra al cargar un código de programación en dicho dispositivo, mediante comunicación serial, lo cual se hace con ayuda de un cable USB. La implementación del código se lleva a cabo en el programa Arduino IDE, el cual ofrece una versión simplificada del lenguaje de programación C.

Por otra parte, dado que se busca implementar un servomotor, es necesario que este tenga la capacidad de realizar giros en ambos sentidos, por lo cual se tuvo que emplear un módulo de relé de 2 canales, el cual permite la conmutación de las señales enviadas por el ESC al motor, según se requiera, logrando los cambios de sentido de giro. Cabe mencionar que los relés son manipulados por señales de control enviadas mediante el Arduino, y además, son capaces de soportar corrientes de hasta 10A, las cuales podrían llegar a alcanzarse en el circuito, por lo que también son necesarios para aislar el microcontrolador de la parte de potencia.

Respecto al sensor de giro y posición, se utilizó un potenciómetro de precisión, esto debido a que es un tipo de sensor resistivo muy sencillo de implementar, de bajo costo, y cumple perfectamente la capacidad de realizar mediciones de posición, e incluso velocidad, en donde la resistencia proporcional al desplazamiento angular, esto para el caso de los potenciómetros rotativos. Este potenciómetro, se acopla a uno de los ejes del sistema de reducción. La señal de realimentación se basa en la lectura del valor analógico de tensión del pin variable del potenciómetro, tensión que cambia con el giro del potenciómetro, por lo que, se aprovecha esta característica para utilizar el potenciómetro como un sensor de giro y posición del motor, el cual al girar, provocará que la perilla del potenciómetro también lo haga.

## II. DESCRIPCIÓN DE LA SOLUCIÓN

### II-A. Sistema de reducción

Para el sistema de reducción, se planteó como objetivo alcanzar una relación de 1:160 debido a las especificaciones de velocidad nominal del motor utilizado como fuente

de movimiento rotacional. Como primera opción se habían planteado 4 etapas, cada una de con una relación de 1:3.5 aproximadamente, sin embargo, dichas relaciones provocaban una un volumen mucho mayor al deseado.

Debido a lo anterior, se consideró la alternativa de utilizar un engrane con tornillo con una reducción de 1:25, por lo que ahora solo resta una reducción de 6.4, para obtener la reducción esperada. Debido a la principal limitación, que es el espacio, se decidió realizar esta reducción en 2 etapas, una de 2.2 y otra de 2.8, con lo cual se obtuvo una reducción final de 1:154. Estas 2 últimas reducciones se dimensionaron en función del acomodo espacial de los demás componentes, considerando que el potenciómetro se encuentra conectado a la etapa previa a la salida.

### II-B. Control del motor BLDC

En cuanto al uso del microcontrolador para la manipulación del motor mediante el controlador ESC-30A. Para esto se muestra en la Figura 1, un diagrama de conexión del Arduino al ESC.

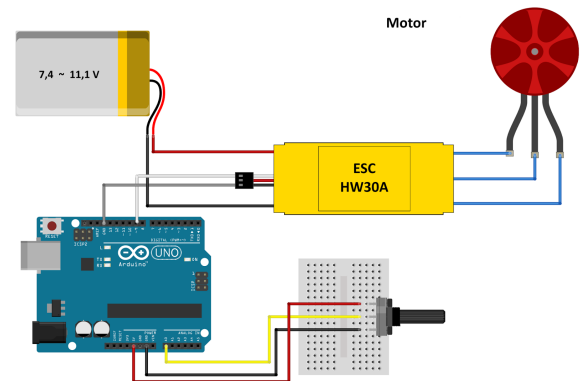


Figura 1. Diagrama de conexión del microcontrolador al ESC para el control del motor [1].

Como se puede observar, únicamente se utiliza una señal pwm de salida desde uno de los pines del Arduino, la cual se cablea hasta una de las entradas del ESC; además, se conecta la tierra del Arduino a la tierra del ESC, para que el ESC tome la señal pwm con la referencia correcta. Esta señal pwm es la encargada de controlar la velocidad del giro del motor, lo cual se logra al aumentar o disminuir el ciclo de trabajo, mediante la programación del microcontrolador. Las salidas del ESC se conectan a los devanados del motor BLDC, no obstante, más adelante se hablará sobre el uso de los relés, en donde se mostrará como 2 de las salidas se deben conectar al módulo de relé para lograr los cambios de sentido de giro del motor.

Además, la fuente de alimentación se conecta al ESC como se muestra en la Figura 1; para este proyecto, se utiliza una tensión de 12V DC, valor recomendado en el manual del motor BLDC empleado. Cabe mencionar que el ESC realiza internamente la conmutación de las señales requeridas para lograr el giro del motor, a partir de la señal pwm recibida desde el microcontrolador. Además, según el voltaje de alimentación, así será la velocidad máxima que pueda alcanzar

el motor si se configura la totalidad del ciclo de trabajo en la señal pwm, por cada 1V de alimentación, se consigue una velocidad de 1000rpm. La lógica del giro del motor se basa en energizar 2 de las bobinas del motor mediante la señal pwm, y siguiendo una secuencia específica. En total se requieren 6 distintas combinaciones de energización de los devanados, para conseguir que el motor gire; dicha secuencia se muestra en la Figura 2.

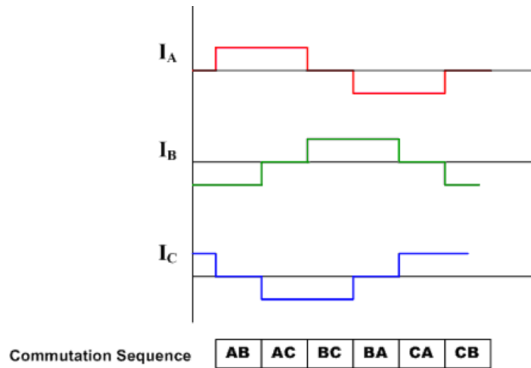


Figura 2. Secuencia de energización de los devanados del motor BLDC [3].

### II-C. Cambio de sentido de giro del motor

Respecto al control del cambio de giro del motor, tal como se mencionó anteriormente, se utiliza un módulo de relés de 2 canales aislado ópticamente de 5v. Para esto se emplea una conexión similar a la mostrada en la Figura 3.

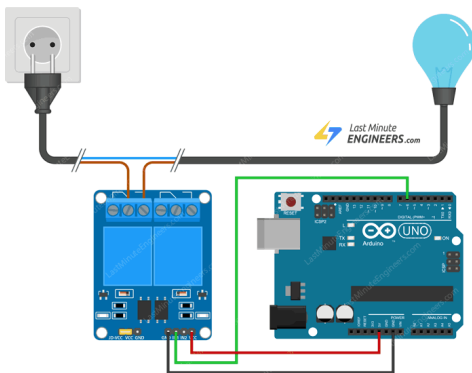


Figura 3. Diagrama de conexión del microcontrolador al módulo de relé de 2 canales [2].

Como se puede observar, primero se alimenta el módulo de relé con 5V desde el Arduino, y se interconectan las tierras; además, se envía una señal digital de control desde el Arduino al módulo de relé, con la cual es posible conmutar uno de los relés. La lógica consiste en que el relé permitirá el paso de la señal que se encuentre conectada al pin común, hacia el pin normalmente cerrado (NC), cuando la señal de control recibida se encuentra en bajo, y por el contrario, al recibir una señal en alto, energiza la bobina interna para que el contacto interno se mueva, haciendo un switch, y por lo tanto, ahora permitirá el paso de la señal del pin común hacia el pin normalmente abierto (NO), tal como se observa en la imagen de la Figura 4.

Para cambiar el giro del motor, simplemente se requiere intercambiar la conexión entre 2 de los cables de salida del ESC hacia el motor, por lo que ahora, lo que se hace es conectar 2 de dichas salidas, cada una a un pin común de uno de los relés, respectivamente, y se cablea la salida NO de un relé, hacia un mismo nodo junto con la salida Nc del otro relé, y dicho nodo será el que se conecte directamente a uno de los devanados del motor BLDC, lo mismo se hace con la salida restante de cada relé, y se conecta al otro devanado del motor. Para conseguir que la señal de 2 de los devanados se intercambie, simplemente se debe enviar una señal en bajo o alto desde el Arduino hacia el módulo de relé, no obstante, dado que ahora se utilizan ambos relés, es necesario conectar 2 señales de control distintas. Entonces, para conseguir un sentido de giro, ambas señales de control se ponen en LOW, y para el giro opuesto, ambas señales se configuran en HIGH.

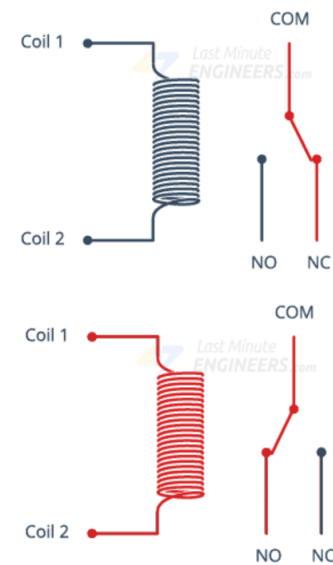


Figura 4. Conmutación de los relés internos del módulo de relé de 2 canales [2].

### II-D. Sensor de giro y posición

Respecto al potenciómetro empleado como sensor, se empleó un potenciómetro de precisión de 1k $\Omega$ , modelo 3590S-2-102L, mostrado en la Figura 5. La conexión del potenciómetro con el microcontrolador para utilizar la lectura del valor de tensión como señal de realimentación, se muestra en el diagrama de la Figura 1. Para esto, se alimenta el potenciómetro con 3.3V desde el Arduino (ya que la salida de 5V DC del Arduino está siendo utilizada para los relés), se interconectan las tierras; y se conecta el pin variable del potenciómetro a una entrada analógica del Arduino, con lo cual es posible leer valores desde 0 a 3.3V.

### II-E. Programación

Considerando todas las conexiones explicadas anteriormente, ahora se hablará sobre el código de programación realizado para el microcontrolador, para lograr el funcionamiento del



Figura 5. Potenciómetro de precisión de 1kΩ, modelo 3590S-2-102L.

servomotor. En prime lugar, se configuran los pines necesarios para controlar los relés, el ESC y leer el valor del potenciómetro; y se definen las variables necesarias. Seguidamente, en el setup se configuran los pines de salida y entrada. Esta sección del código se muestra en la Figura 6.

```
#include <Servo.h>
//Se crea un objeto tipo Servo para controlar el ESC
Servo ESC;
int pwm=0;
int sensorPin = A0;
int potValue=0;
int RelayPin = 6;
int RelayPin2 = 7;
int pwmIn=0;
int angulo=0;
int llego=0;
int cont=0;
int pregunta=0;

void setup() {
    Serial.begin(250000);
    pinMode(A0, INPUT);
    pinMode(RelayPin, OUTPUT);
    pinMode(RelayPin2, OUTPUT);
    ESC.attach(9, 1000, 2000);
    // (pin 9, ancho de pulso mínimo, ancho de pulso máximo en microsegundos)
}
```

Figura 6. Definición inicial de pines y variables.

Luego inicia la función loop, la cual se ejecuta infinitamente hasta terminar el programa. Dentro del loop, primero se revisa la condición de pregunta, la cual define si se solicitan los valores de velocidad y ángulo o no. Si ya se preguntó previamente, cada vez que se entre al loop, se ignoran dichos comandos para preguntar, hasta que se llegue al valor de ángulo deseado y se quiera ir a una posición nueva. Después, el ángulo introducido, se mapea desde los valores de -360-360, a los valores 136-522, esto para compararlos con los valores leídos del potenciómetro. Esta sección del código se muestra en la Figura 7.

Luego, se efectúa la lectura de la tensión del potenciómetro mediante un pin analógico del Arduino (este valor se guarda en la variable potValue), y se procede a comparar dicho valor, el cual indica el valor de ángulo actual, con el valor de referencia dado por el ángulo deseado (variable "ángulo"). Primero se verifica si el ángulo actual se encuentra dentro de un rango de  $\pm 6^\circ$ , en cuyo caso no es necesario que el motor gire, pues inicialmente ya se encuentra muy cerca del valor deseado. De lo contrario, luego se verifica si el valor actual es mayor al de referencia, y seguidamente se verifica si el valor actual ha entrado anteriormente o no, dentro del rango deseado. En caso de haber entrado, el giro del motor se detiene, ya que independientemente de que haya entrado en dicho rango o no,

```
void loop(){
    //Petición de valores de velocidad y ángulo
    Serial.println(pregunta);
    if(pregunta==0){
        llego=0;
        cont=0;
        pwm=0;
        pwm = map(pwm, 0, 1023, 0, 180);
        ESC.write(pwm);
        Serial.println("\n");
        Serial.println("Ingrese la velocidad deseada (0-12000rpm): ");
        while (true){
            pwmIn=Serial.parseInt();
            if(pwmIn==0){break;}
        }
        Serial.println("La velocidad es: ");
        Serial.println(pwmIn);
        Serial.println(" rpm");
        pwmIn=pwmIn*0.0025;
        Serial.println("Ingrese el ángulo deseado (-360-360): ");
        Serial.println(Serial.available());
        while (Serial.available()==1){
            angulo=Serial.parseInt();
            angulo=angulo*1.09;
            Serial.println("El Ángulo es: ");
            Serial.println(angulo);
            Serial.println("\n");
            //El ángulo deseado se mapea a valores entre 136 y 522 para ser comparados con la lectura del potenciómetro.
            //angulo=map(angulo, -360, 360, 136, 522);
            angulo=map(angulo, -360, 360, 153, 539);
            pregunta=1;
        }
    }
}
```

Figura 7. Petición de valores de velocidad y ángulo.

al entrar a la condición en que el valor actual es mayor, se configurarán los 2 pines de salida que controlan el módulo de relés en LOW, haciendo que el sentido de giro sea antihorario. Por lo que, de esta forma, se asegura que el motor no cambie el sentido estando en movimiento, ya que, según la lógica planteada, si el motor entra dentro del rango deseado, pero por la inercia sigue moviéndose y se sale de dicho rango, el motor debe cambiar de sentido de giro para volver a entrar en el rango deseado. Esta sección del código se muestra en la Figura 8.

```
//Lectura del potenciómetro
double potValue=analogRead(sensorPin);
Serial.println("Valor actual del potenciómetro: ");
Serial.println(potValue);
Serial.println("Tensión de referencia: ");
Serial.println(angulo);
//Si el ángulo solicitado es distinto a la posición actual
if(potValue<(angulo-3) or potValue>(angulo+3)){
    Serial.println("El valor actual es diferente: ");
    //Si el ángulo solicitado es menor al ángulo actual
    if(potValue>angulo){
        //En caso de entrar al rango del valor deseado se detiene el motor
        if(llego==1){
            //Serial.println("Se reinició el conteo");
            llego=0;
            pwm=0;
            pwm = map(pwm, 0, 1023, 0, 180);
            ESC.write(pwm);
            delay(800);
        }
        //Se configura el sentido de giro respectivo enviando señales de LOW a los relés
        Serial.println("Girando en sentido antihorario: ");
        digitalWrite(RelayPin, LOW);
        digitalWrite(RelayPin2, LOW);
        delay(100);
    }
}
```

Figura 8. Lectura del potenciómetro y condicionales de valor inicial, y valor actual mayor.

Luego, se verifica el caso contrario, en donde el valor actual es menor al de referencia, en donde ocurre lo mismo que el caso anterior, pero ahora, se configuran los 2 pines de salida para el módulo de relés en HIGH, provocando que el sentido de giro sea horario. Para este caso, el código es muy similar al mostrado en la la Figura 8.

Después, se verifica si se entró dentro del rango deseado, en este caso, se da la señal para detener el motor, y se procede a aumentar el valor de un contador (variable "cont"), lo cual se utiliza para lo siguiente: si el valor actual permanece durante un tiempo considerable dentro del rango deseado, el contador alcanza a contar hasta cierto valor, en donde ya se ha supuesto que el ángulo actual se alcanzó y el motor se encuentra detenido, por lo que se indica que se logró el ángulo deseado, y ahora la variable "pregunta" se reinicia, por lo que en el siguiente ciclo del loop, volverá a preguntar por un ángulo y velocidad nuevos. En caso de que el contador no llegue a

contar la suficiente cantidad, esto significa que se entró en el rango, pero la inercia provocó que se saliera del rango deseado, por lo que en el siguiente ciclo del loop, se invertirá el giro del motor con las condiciones mencionadas anteriormente, para corregir el ángulo actual y llegar al valor requerido. Dicha sección del código se muestra en la Figura 9.

Cabe mencionar que por la inercia del motor, según la velocidad configurada, puede ocurrir que en varias ocasiones se requiera cambiar el giro del motor para alcanzar el valor deseado, pero esto se realiza con el objetivo de mejorar la precisión. Finalmente, se encuentra la sección que da la señal de giro al motor, esta se encuentra al final ya que en este punto, se revisaron las condiciones necesarias para determinar si el motor debe girar o no, y hacia cual sentido. Esta sección del código se muestra en la Figura 9.

```
//Si se ingresa al rango del ángulo deseado
if(potValue>(angulo-3) and potValue<(angulo+3)){
    //Se detiene el motor
    pwm=0;
    pwm = map(pwm, 0, 1023, 0, 180);
    ESC.write(pwm);
    llego=1;
    //Aumenta un contador para mejorar la precisión o corregir el ángulo.
    cont=cont+1;
    Serial.println(cont);
    //Si ha contado la suficiente cantidad con el angulo actual dentro del rango deseado:
    if(cont>50){
        Serial.println("Se obtuvo el ángulo deseado: ");
        pregunta=0;
    }
}
potValue=analogRead(sensorPin);
//Comienza el giro del motor con la velocidad deseada
if(potValue>(angulo-3) or potValue<(angulo+3)){
    Serial.println("Motor girando: ");
    pwm=(pwm);
    pwm = map(pwm, 0, 1023, 0, 180);
    //Se envia la señal de pwm al ESC
    ESC.write(pwm);
}
```

Figura 9. Condicional de entrada al rango deseado.

### III. RESULTADOS EXPERIMENTALES

Primeramente, se realizaron 3 mediciones de la masa del servomotor, con una balanza digital. Estas mediciones se muestran en el Cuadro I.

Cuadro I  
MEDICIONES EXPERIMENTALES DE LA MASA DEL SERVOMOTOR.

Medición	Masa ( $\pm 0.01$ )g
1	369.41
2	369.41
3	369.42

También se efectuaron 3 mediciones de la masa del brazo acoplado al servomotor, con la balanza digital, esto cuando el motor está en funcionamiento; estos datos se muestran en el Cuadro II. A partir de dichos datos, se efectúa una conversión a peso, el cual se toma como la fuerza ejercida por el servomotor, y mediante el brazo de palanca, el cual es la longitud del brazo, se calcula el torque aplicado por el servomotor.

Cuadro II  
MEDICIONES EXPERIMENTALES DE LA MASA DEL BRAZO, CON EL SERVOMOTOR EN FUNCIONAMIENTO.

Medición	Masa ( $\pm 0.01$ )g
1	377.07
2	384.12
3	379.48

Además, se midieron 3 valores de la velocidad del motor, con ayuda de un tacómetro digital, mostrados en el Cuadro III.

Cuadro III  
MEDICIONES EXPERIMENTALES DE LA VELOCIDAD DEL SERVOMOTOR.

Medición	Velocidad ( $\pm 0.1$ ) rpm
1	48.7
2	45.4
3	43.6

Además, se midió con una regla, 3 valores del backlash del servomotor, datos mostrados en el Cuadro IV.

Cuadro IV  
MEDICIONES EXPERIMENTALES DEL BACKSLASH DEL SERVOMOTOR.

Medición	Backslash ( $\pm 0.5$ ) mm
1	6.0
2	6.5
3	7.0

Por último, se realizaron 2 mediciones para 4 distintos ángulos alcanzados por el servomotor, con ayuda de un transportador; estos valores se muestran en el Cuadro V, a partir de los cuales se determinó la precisión del servomotor.

Cuadro V  
MEDICIONES EXPERIMENTALES DE LOS ÁNGULOS DEL SERVOMOTOR.

V.Teórico ( $^{\circ}$ )	V.Experimental 1 ( $\pm 1$ ) $^{\circ}$	V.Experimental 2 ( $\pm 1$ ) $^{\circ}$
45	48	49
90	91	92
-45	-48	-47
-90	-89	-91

Resumidamente, se muestra en el Cuadro VI, los valores experimentales promedio de los distintos datos medidos, además de aquellos calculados indirectamente.

Cuadro VI  
VALORES EXPERIMENTALES PROMEDIO DE DISTINTAS MAGNITUDES FÍSICAS DEL SERVOMOTOR.

Medición	Valor Promedio
Masa	369.41 ( $\pm 0.01$ )g
Volumen	502.8 ( $\pm 0.1$ )cm <sup>3</sup>
Masa Ejercida	380.22 ( $\pm 0.01$ )g
Longitud Brazo	28.6 ( $\pm 0.1$ )cm
Torque	1.07 ( $\pm 0.01$ )N·m
Velocidad	45.9 ( $\pm 0.1$ )·rpm
Backslash	6.5( $\pm 0.5$ )mm
Precisión	96.3 %

### IV. ANÁLISIS DEL DISEÑO

Primeramente, se debe mencionar que el diseño del servomotor implementado, es completamente funcional, ya que es posible ingresar valores de velocidad y ángulo desde la terminal serial de la computadora, y esta se comunica correctamente con el microcontrolador, para indicarle al motor que comience a girar según la velocidad establecida.

Además, el sistema de reducción desarrollado, no presenta problemas para efectuar la reducción de velocidad del motor, ya que el eje de salida gira libremente, a una velocidad mucho menor, tal como lo indica el valor medido experimentalmente

de 45.9 rpm, mostrado en el Cuadro VI. Esto también implica, que la reducción aplicada aumentó correctamente el torque aplicado por el motor; esto se puede observar en el Cuadro VI, donde se observa que el valor promedio del torque aplicado, es de 1.07 N·m. Dicho torque fue calculado a partir del valor de masa del brazo acoplado al eje de salida del servomotor, mientras el motor se encuentra en funcionamiento, cuyo valor se muestra también en el Cuadro VI, considerando que el brazo de palanca, es decir, la distancia desde el eje de salida hasta la punta del brazo, es de 28.6cm.

También, es posible observar las características físicas del servomotor diseñado en el Cuadro VI, donde se tiene que la masa experimental promedio es de 369.41 g, y el volumen es de 502.76 cm<sup>3</sup>, valores apropiados para la aplicación requerida, ya que se alcanzó un diseño funcional con un tamaño bastante pequeño, y con poco peso.

En cuanto a la medición del backlash del servomotor, esta fue de 6.5mm, según lo muestra el Cuadro VI. Dicha medida se tomó con el juego presente en el brazo, el cual se debe a razones como el propio backlash del engrane de salida, además de que el eje de salida, presentaba un pequeño juego respecto al engrane.

Otro aspecto por destacar, es la presencia de la fricción estática, ya que el motor únicamente empieza a girar a partir de cierto valor de velocidad indicado, el cual es aproximadamente de 3500rpm, no obstante, esta fricción es intrínseca del propio motor BLDC empleado, ya que con ese mismo valor, el motor es capaz de comenzar a girar, incluso acoplado al sistema de reducción.

Respecto a las mediciones efectuadas para los ángulos solicitados, se obtuvieron mediciones con un 96.3 % de precisión, lo cual indica un acierto bastante bueno en cuanto a ángulos específicos. Esto demuestra que el dispositivo tiene una buena funcionalidad, ya que la precisión es una de las características más importantes en un servomotor.

## V. CONCLUSIONES

- El uso de un potenciómetro de precisión en la etapa previa a la salida permitió un control de posición con una precisión bastante alta.
- La reducción mediante el tren de engranes y el tornillo sin fin permitió obtener una velocidad de salida, y un par de torsión óptimos para el tamaño del dispositivo.
- Debido al pequeño tamaño de los engranes utilizados, se obtuvo un dispositivo con un backlash bastante pequeño, lo cual ayuda a un funcionamiento más preciso del dispositivo.
- La masa y volumen obtenidos demuestran que el dispositivo diseñado y manufacturado fue optimizado en sus dimensiones físicas para poder ser implementado en proyectos de robótica a mediana y gran escala.

## VI. RECOMENDACIONES

- Unos de los principales factores en la construcción del dispositivo es la precisión en la manufactura de elementos como ejes y engranes, por lo que se recomienda enfocarse principalmente en estos componentes.

- En caso de utilizar potenciómetros rotacionales, es importante cuidar que no se rebasen las revoluciones máximas que el potenciómetro puede soportar, por lo cual se recomienda implementar tanto en el control, como en la parte física del dispositivo, un sistema preventivo que evite que este elemento pueda dañarse, ya que en él se basa el funcionamiento del dispositivo.
- A la hora de diseñar, se recomienda tomar en cuenta el montaje y principalmente, el desmontaje de los componentes, ya que esto permite realizar cambios y reparaciones fácilmente en el dispositivo.

## REFERENCIAS

- [1] Sobhy, A. (2015). *How to control a brushless motor through a ESC with Arduino* Recuperado de: <https://sobhy.me/134/>
- [2] LastMinuteENGINEERS.(2022). *Interface Two Channel Relay Module with Arduino*. Recuperado de: <https://lastminuteengineers.com/two-channel-relay-module-arduino-tutorial/>
- [3] Renesas. (2022) *BLDC Motor Control Algorithms* Recuperado de: <https://www.renesas.com/us/en/application/key-technology/motor-control-robotics/bldc-motor-control-algorithms#document>