

Laboratorio 2 - Electrónica digital III (Caracterización de motor DC)

JUAN DAVID RIOS RIVERA¹, (1116922203), MARIO ALEJANDRO TABARES OREJUELA², (1113788443)

¹Estudiante Ingeniería Electrónica: (e-mail: juan.rios30@udea.edu.co)

²Estudiante Ingeniería Electrónica: (e-mail: alejndro.tabares@udea.edu.co)

Departamento de Ingeniería Electrónica - Universidad de Antioquia

• **ABSTRACT** En el presente informe se detalla el proceso llevado a cabo para el diseño e implementación de un banco de pruebas para un motor DC, obteniendo el sensado de variables dadas las condiciones a las que se somete el motor. El sistema de control y lectura es implementado en un microcontrolador arduino basado en el microprocesador AVR ATmega328P, usando el IDE de arduino.

• **INDEX TERMS** Amplificador, ATmega328P, transistor

I. INTRODUCCIÓN

EL microcontrolador Arduino basado en el microprocesador ATmega328P posee una resolución de 8 bits, un conversor AD, además de puertos análogos y digitales que permite la lectura y escritura en ellos, ya sea para leer valores de sensores o controlar actuadores. En este caso se usa para enviar una señal PWM por un puerto de salida digital (2) que controla la velocidad de un motor con ayuda de un transistor de potencia. A medida que esto se realiza se toman datos en diferentes punto de tiempo para analizar el comportamiento del motor bajo diferentes condiciones o velocidades. Las variables a tener en cuenta en las lecturas se leen a través de puertos análogos y digitales, como lo son temperatura, corriente, velocidad y empuje. Lo anterior con el objetivo de caracterizar el motor y tener valores conocidos para realizar su control sobre el mismo, una vez este funcione como motor de un dron de bajo costo planteado como proyecto final del curso de Electrónica Digital III.

II. MARCO TEÓRICO

Se muestran algunos de los teoremas y técnicas utilizados para la elaboración de esta práctica, como lo son los montajes usados en la parte analógica del sensado y acondicionamiento de las variables, además de los sensores usados para obtener los datos de temperatura, velocidad, corriente y empuje.

A. AMPLIFICADOR NO INVERSOR

Este tipo de configuración se caracteriza por tener conectado la señal de entrada (voltaje) a la terminal no inversora, esto nos indica que la ganancia será positiva (al contrario del inversor). También tiene como característica una realimentación negativa (conocida normalmente como R_f). En

esta configuración la entrada está en fase con la salida, es decir, no hay desfase en señal alterna, a diferencia de la configuración inversora. Otra característica es que la ganancia siempre será mayor a 1, su configuración se muestra en la figura 1. Y cumple con la ecuación de salida amplificada [1] dada por

$$V_{out} = V_{in} \left(1 + \frac{R_f}{R_i} \right)$$

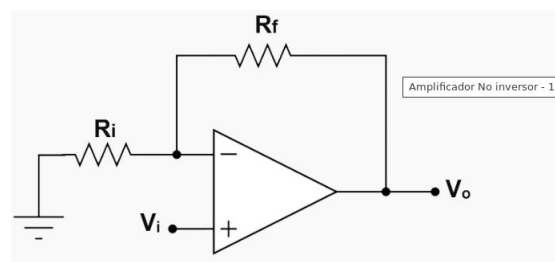


FIGURE 1: Amplificador no Inversor

B. LM35

El LM35 mostrado en la figura 2 es un sensor de temperatura analógico con buenas prestaciones, económico, sencillo de usar, dimensiones reducidas. Permite medir temperatura en un rango desde -55°C hasta 150°C . Su salida es de tipo analógica y lineal con una pendiente de $10\text{mV}/^{\circ}\text{C}$. El sensor es calibrado de fábrica a una precisión de 0.5°C . Es un sensor muy popular por su fácil uso y variadas aplicaciones. No necesita de ningún circuito adicional para ser usado. Se alimenta directamente con una fuente de 5V y entrega una salida analógica entre 0V a 1.5V. [2].



FIGURE 2: Medidor de temperatura LM35

C. CELDA DE CARGA

Las celdas de carga mostrada en la figura 3 son transductores que soportan cargas de compresión, tensión y flexión y las convierten en una magnitud eléctrica proporcional a la carga. Existen celdas de carga de tipo hidráulico, neumático y con galgas extensiométricas, siendo estas últimas las más utilizadas en la industria por su precisión, durabilidad y precio. La aplicación principal de las celdas de carga es la medición de peso en la industria de alimentos, medicamentos, transporte entre otras. Esta celda además viene con su conversor AD para evitar que la resolución del microcontrolador no pueda leer valores tan bajos de voltaje que entrega, este conversor es de 24 bits [3].

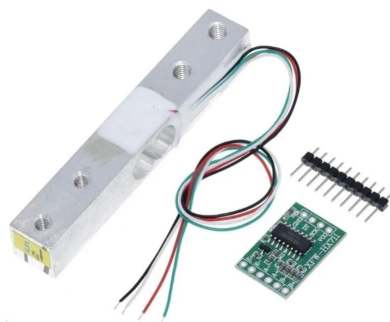


FIGURE 3: Celda de 5 kg

D. TIP 122

El TIP122 mostrado en la figura 4 es un transistor con diodo darlington TH (through hole) NPN de propósito general con encapsulado TO-220. Este transistor se utiliza para la conmutación de baja velocidad y amplificación. El funcionamiento y utilización de los transistores de potencia es idéntico al de los transistores normales, teniendo como características especiales las altas tensiones e intensidades que tienen que soportar y, por tanto, las altas potencias a disipar.

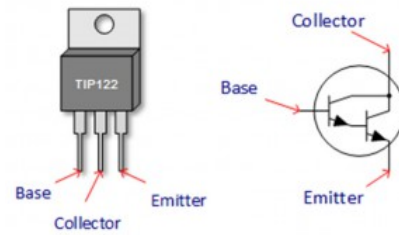


FIGURE 4: TIP 122

E. INFRARROJO E18-D80NK

El sensor de proximidad fotoeléctrico E18-D80NK [4] mostrado en la figura 5 permite detectar objetos sin necesidad de contacto en un rango de hasta 80cm. De fácil integración con sistemas digitales como Arduino, Pícs o PLCs. Solo es necesario conectar la salida del sensor a una entrada digital del microcontrolador. La salida de este sensor es de tipo NPN normalmente abierto, por lo que al detectar un objeto la salida será GND y en estado de reposo VCC. Puede ser alimentado con 5V y trabajar directamente con Arduino. La sensibilidad del sensor es regulable mediante un potenciómetro dentro del mismo sensor.



FIGURE 5: Infrarrojo de proximidad E18-D80NK

III. PROCEDIMIENTO EXPERIMENTAL Y RESULTADOS

Para la practica se desarrolla la caracterización de un motor por medio de lectura 4 variables que nos indican el estado del motor y sus propiedades. Cada una de estas es realizada por medio de circuitos independientes que luego son controlados por el microcontrolador y mostradas en pantalla, a través de la consola serial del Arduino IDE. Se explica la medición de cada una de las variables a continuación, para las cuales se usaron funciones que luego se llaman en otra parte del

código, cada una de estas variables fue testeada en forma individual para asegurarse del correcto funcionamiento y que la medida otorgada fuese real.

A. CONTROL DE VELOCIDAD

Para controlar la velocidad del motor se usa una señal PWM que sale por el pin 5 del arduino, este pin función a una frecuencia de 1KHz. Debido a que el arduino solo puede brindarnos un voltaje de 5v y una corriente de 10mA se hace imposible usar esta salida directamente a un motor sin adecuar el microcontrolador, así que para ello se usa el TIP122, al cual se conecta a la base la señal pulsada y al colector va el motor conectado a VCC, el emisor del transistor va a GND, y en esta configuración se logra que el transistor actúe como switch permitiendo el paso de un voltaje dado la dureza del pulso PWM, de esta manera a mayor dureza del pulso mayor voltaje va fluye de colector a emisor, aumentando el voltaje que va hacia el motor.

B. MEDIDOR DE CORRIENTE

Para la medida de corriente en el motor, se realizó un circuito en serie con una resistencia de bajo valor, teóricamente 10hm, medida 0.5ohm reales, para poder medir el voltaje que cae sobre la resistencia y calcular la corriente con las leyes de Kirchhoff. Se realiza el circuito con una amplificación previa de voltaje, esto debido a que la resolución del arduino de 8 bits no lee valores muy cercanos a 0V, es decir valores de milivoltios, el amplificador se usa con ganancia de 2, para que cuando obtengamos la medida de voltaje inferior a un voltio, esta pueda ser leída por el puerto análogo del arduino.

C. MEDIDOR DE VELOCIDAD

Para la medida de la velocidad del motor, se elaboró un circuito compuesto por un sensor infrarrojo que detecta un objeto al ser puesto frente al sensor, el sensor es calibrado para que cuando las hélices pasen por el infrarrojo se detecte un obstáculo y genere una interrupción que se lee por uno de los puertos digitales del arduino usando la función para establecer interrupciones `attachInterrupt()` [5], cuando una interrupción se presenta se llama una función que aumenta un contador y así se leen las revoluciones por un periodo de tiempo, que luego se convierten a revoluciones por minuto, multiplicando por el valor de un segundo en micros ($1e6$) y dividiendo por el tiempo en micros en el que se calcularon las interrupciones, luego se multiplican por 30 en lugar de 60 segundos en un minuto, para pasar a rpm, debido a que se tienen 2 hélices, es decir el doble de interrupciones.

D. MEDIDOR DE EMPUJE

Para la medida de empuje, se usa una celda de carga con capacidad de 5 kg en el montaje observado en la figura 6. Esta es adaptada en la estructura de tal forma que cuando

se enciende el motor, la fuerza realizada por las hélices es percibida por el sensor y amplificada por el driver conectado a este. Luego se envía la señal amplificada al microcontrolador por medio del pin 8. Vale resaltar que el driver de usado fue el HX711 el cual posee un conversor ADC con resolución de 24 bits para facilitar la lectura de valores muy bajos en la celda de carga. Así se evitan estos cálculos en el microcontrolador, y se obtiene una buena resolución que en este caso es de gramos de empuje, para la lectura de estos datos se usan la librería HX711 que viene adaptada a este conversor, el cual usa una reloj que se conecta al pin 7 en como salida del microcontrolador.

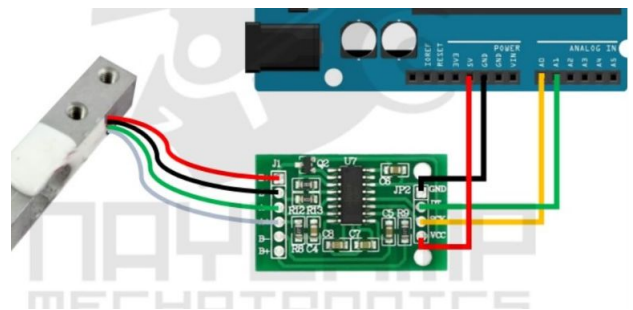


FIGURE 6: Frecuencia menor a 1Hz

E. MEDIDOR DE TEMPERATURA

Para la medida de temperatura, se elaboró un circuito con un sensor LM35 que hace contacto con la estructura del motor y percibe la variable. Posteriormente el dato es enviado al puerto A0 del arduino donde es procesado y transformado para su visualización en pantalla. El conversor AD del microcontrolador posee la resolución para leer los valores de temperatura del LM35 que brinda $10mV/^{\circ}C$, dado esto como son valores de más de 20 grados, la lectura se hace sin problema. La conversión es sencilla, solo se multiplica por el valor máximo de lectura que sería 5v con los cuales se alimenta el sensor, se divide además por la característica de transferencia mencionada de 10mV por grado y se divide por 1024 que son los posibles valores que entrega el puerto analógico.

F. FUNCIONES STP Y TRG

Para dar cumplimiento al desarrollo de la práctica se tiene en el banco de pruebas 2 diferentes modos. Por un lado está el modo de pasos de PWM donde cada uno de los pasos está controlado por una variable que se puede modificar desde la consola de comandos por el usuario el paso es un valor entre 0 y 100% del duty, y este se mantiene en un periodo de tiempo determinado que también es modificable por el usuario. Esta función se ejecuta mediante periodos de tiempo es decir con ayuda de la variable micros del entorno de arduino se toma inicialmente una medida de tiempo que luego con ayuda de un condicional se evalúa si ya ha superado el intervalo

de tiempo deseado, dando paso así al siguiente escalón o aumento de duty, en otras palabras aumentar la velocidad cada cierto tiempo. Las variables son censadas de acuerdo a un periodo de muestreo definido por el usuario, este tiene que ser obviamente menor al tiempo durante el cual el duty se encuentra estable, es decir el tiempo de periodo, puesto que en caso de que se desee tomar promedio de muestras no se desea que se solapen las muestras de una velocidad con otra.

Por otro lado en el triangular el duty se aumenta en pasos de uno durante la mitad del periodo de tiempo hasta 100% y en la otra mitad se reduce hasta 0. Esto se ve como si el duty estuviera aumentando de manera lineal, y las muestras se toman de acuerdo al periodo de muestreo definido anteriormente de manera similar al modo STP.

El promedio de las muestras se realiza llenando una matriz de filas y columnas fija donde las filas representan cada una de las variables sensada y las columnas las diferentes muestras a promediar. Con un índice se recorren las columnas por cada fila y se determina si ya se ha alcanzado el número de muestras que el usuario ha definido que desea promediar una vez alcanzado este índice las muestras se suman y se dividen por la cantidad de muestras. Esto se repite por cada fila, es decir cada variable y dicho valor se guarda en la tabla de datos que se mostrará al final de la prueba. Si el usuario no ha definido un promedio, entonces cada vez que se realiza una muestra se guarda en la tabla de resultados.

Para realizar la medida de las variables se realiza durante un periodo corto de tiempo definido por el usuario como periodo de muestreo. En este periodo de muestreo se espera para obtener las interrupciones producto de la velocidad. Y mientras esperan estas interrupciones, se toma en 3 diferentes instantes de tiempo cada una de las variables restantes, es decir, temperatura, corriente y empuje, esto se hace con el objetivo de que el conversor AD del microcontrolador se encuentre listo para obtener el valor que entra por el puerto analógico de cada una de las variables que lo requieran.

Vale resaltar que en el modo TRG la espera se hace durante el período de tiempo que el duty se encuentra estable, es decir, durante el periodo de tiempo en que se aumenta el duty, el cual es variable pues es el resultado de dividir el periodo de la prueba por 200, ya que esa es la cantidad de pasos que deben darse, estos periodos son más cortos que el periodo de muestreo, así que el periodo de muestreo simplemente se usa para saber en que punto se debe tomar la muestra a ingresar a la función muestreo. En cambio, en el modo STP el periodo de muestreo se usa para esperar durante todo este periodo a tomar las muestras. Esto debido a que en todo ese periodo el duty no ha cambiado.

IV. DISCUSIÓN DE RESULTADOS

A continuación, se muestra los resultados obtenidos en la practica luego de su caracterización. Estos datos fueron obtenidos por medio del software de Matlab con los datos recolectados en Arduino con la finalidad de lograr sacar un diseño de control del sistema ante los modos de operación de la practica.

A. DATOS STP

Para esta sección, se tomaran los datos del Arduino a partir del 70% de duty, esto debido a que el motor empieza a percibir movimiento luego de un valor superior a los 60%. En la figura 7, se muestra el incremento del duty cuando esta en modo STP, allí se pueden observar cambios de variable de duty del 10% y junto a esta, se visualiza la gráfica de empuje que recibe el sensor ante esos cambios.

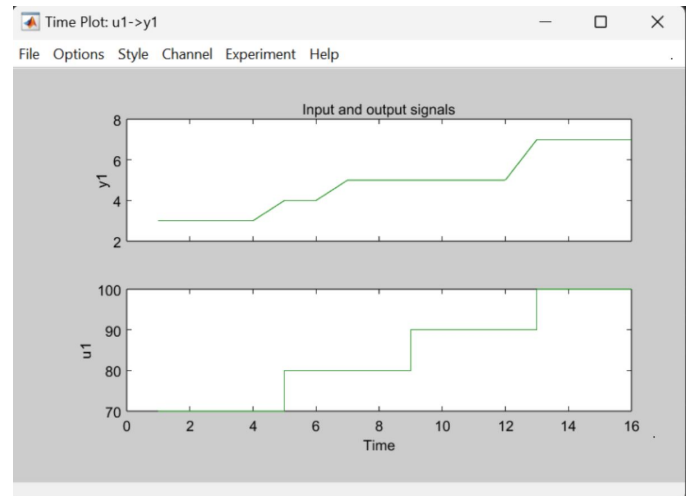


FIGURE 7: Gráfica STP y empuje

Se observaron mejoras en los valores de empuje del prototipo cuando se le hicieron modificaciones al sensor, colocándolo mas cerca y estabilizando mejor la base. En la gráfica se puede chequear que en los valores de duty en donde el motor tenia movimiento, el mayor empuje percibido por el sensor es cuando el duty es igual 100%, para esta ocasión el valor de empuje máximo es igual a 7g aproximadamente. Es importante tener en cuenta que los valores de empuje pueden verse influenciados por varios factores, como la carga del motor, la tensión de alimentación, la estabilidad del chasis, entre otros.

A continuación, la figura 8 muestra la ecuación que mejor se adapta al modelo, para esto se hizo uso de la herramienta de Matlab "System Identification Toolbox" [6], en donde se aprecia que la ecuación que mejor se ajustaba al sistema correspondía a un sistema de 2 Ceros y 4 Polos.

```
From input "u1" to output "y1":
-0.1073 s^2 + 0.01608 s + 0.004209
-----
s^4 + 0.834 s^3 + 1.234 s^2 + 0.6278 s + 0.06022
Name: tf2
Continuous-time identified transfer function.
```

FIGURE 8: Ecuación STP

En la figura 9 se obtuvo el diagrama de Polos y Ceros con la misma herramienta, allí se pudo observar que el sistema obtenido era estable, ya que sus polos se encuentran al lado izquierdo del plano imaginario.

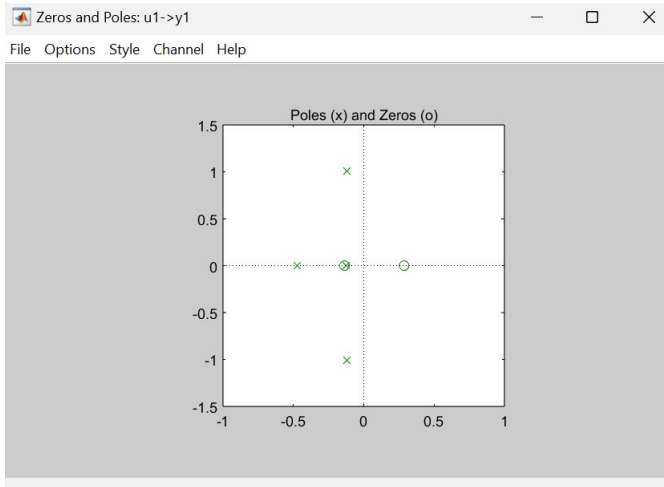


FIGURE 9: Diagrama de Polos y Ceros STP

También podemos observar de la gráfica anterior que hay dos polos en el sistema que no tiene parte imaginaria, mientras que los dos polos restantes son la parte compleja del otro. Por otra parte la figura 10, cuenta con dos gráficas en ella, la primera es la gráfica del modo STP, y solapada a esta obtenemos la gráfica de la función encontrada con la herramienta de Matlab, esta tiene una aproximación del 67%, con respecto a la gráfica original obtenida de los datos de la función STP. La segunda gráfica corresponde a la respuesta de la función ante una excitación escalón, de allí podemos corroborar lo del diagrama de polos y ceros, mirando que ante la perturbación en la entrada, el sistema se estabiliza.

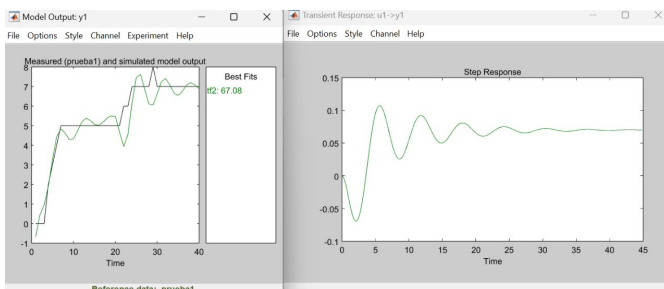


FIGURE 10: Respuesta al escalón

Para obtener mejores resultados del sistema de control, se realizó una toma de datos para cambios grandes de duty de al menos un 20% para observar su comportamiento. De igual manera que en el procedimiento anterior, la figura 11 muestra la gráfica del modo STP para un cambio de duty del 80% al 100%. Aquí también podemos observar el comportamiento del empuje cuando hay un cambio de nivel, su valor máximo lo alcanza cuando el duty es igual a 100% y tiene un valor de aproximadamente 8g.

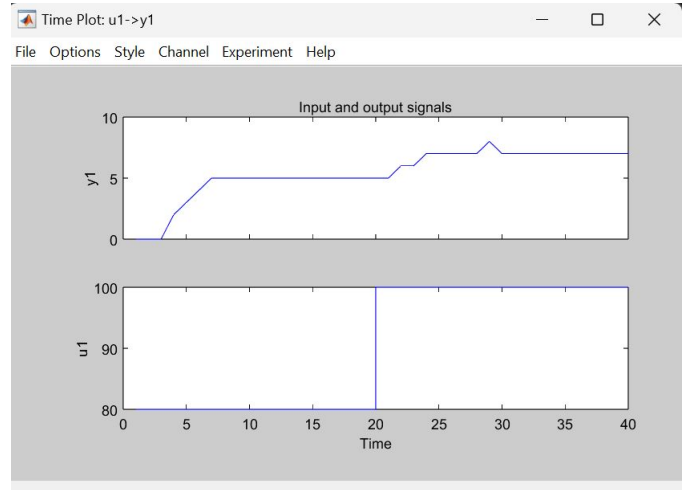


FIGURE 11: Grafica STP y empuje 20% duty

Para esta toma de datos, se tuvieron los mismos criterios a la hora de elegir la ecuación que mejor se ajuste al sistema con la herramienta de Matlab. Encontrando que el sistema que mejor se ajusta a los datos es una ecuación con dos ceros y dos polos como se muestra en la 12

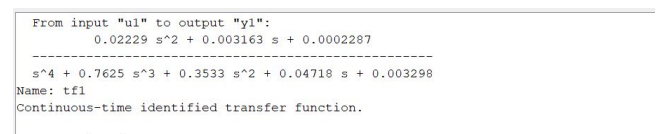


FIGURE 12: Ecuación STP duty con pasos de 20%

Se realiza esta retoma de datos para el modo STP, ya que los cambios de duty pueden ser determinados por el usuario en comparación con el TRG. En la figura 13 obtenemos nuevamente las dos gráficas, en la primera gráfica se aprecia el modo STP en donde observamos que la función solapada, es mas parecida con la original a comparación de la anterior, encontrando un grado de similitud del 89%. Para la segunda gráfica, encontramos la respuesta de la ecuación ante un escalón, observando mejorías en la estabilidad del sistema.

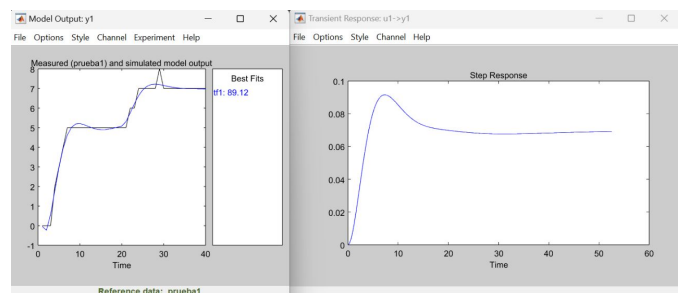


FIGURE 13: Respuesta al escalón modo STP con 20% duty

B. DATOS TRG

Por ultimo, en esta sección se muestran los datos obtenidos para el modo TRG, la figura 14 mostrada se tienen dos

gráficas en ella, la primera, muestra como es la variación del empuje a medida que el duty aumenta y disminuye. Mientras que la segunda grafica muestra los incrementos y decrementos de duty para este modo, los cuales se programaron para hacerse de 1% de aumento para que se observara esta señal triangular.

En este modo se corrobora que el sensor de empuje no empieza a mandar datos de la variable sino a partir del 65% del duty, obteniendo que la fuerza aumenta de manera lineal hasta alcanzar un valor máximo de 8g cuando el duty llega a su valor limite. Posterior a eso el empuje decrece de manera no tan lineal como al comienzo cuando el duty disminuye, debido a que el motor se encuentra en movimiento y lleva inercia en los cambios de velocidad.

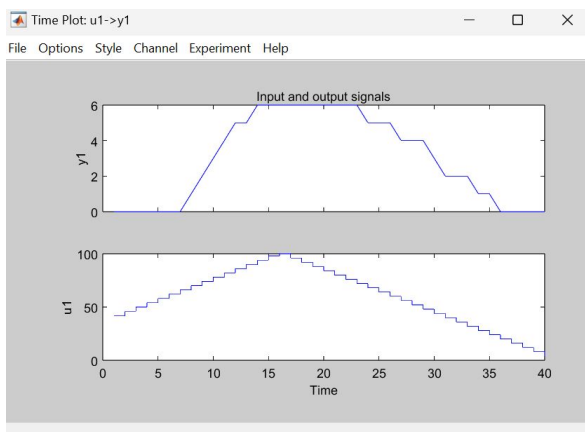


FIGURE 14: Grafica TRG y empuje

La figura 15 muestra la ecuación que mejor se ajusta a el modo TRG, de igual manera que en la tomas anteriores, la función que mejor se adapta al sistema es una función con dos ceros y 4 polos.

```
From input "u1" to output "y1":
-0.0003046 s^2 - 0.0003733 s + 0.0004334
-----
s^4 + 0.2872 s^3 + 0.2825 s^2 + 0.02282 s + 0.009462
Name: tf15
Continuous-time identified transfer function.
```

FIGURE 15: Ecuación modo TRG

Se observó que tanto para el modo STP en los dos casos y en el modo TRG, las funciones de transferencia arrojadas para el controlador PID por medio de Matlab, se ajustaban mejor a la de 4 polos, ya que con 2 polos la ecuación resultante no era tan exacta y a diferencia con la de 6 polos no hacia mejoras notables con la respuesta al sistema. La figura 16, muestra el diagrama de polos y ceros para el modo TRG, en esta podemos chequear que el sistema es estable por lo mencionado anteriormente y que además todos sus polos tienen parte imaginaria a diferencia con el modo STP.

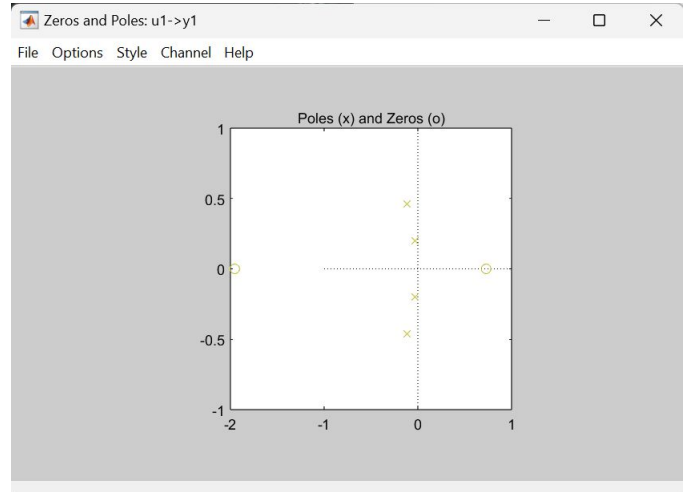


FIGURE 16: Diagrama de polos y ceros modo TRG

Ya por ultimo se obtiene la gráfica a la respuesta al escalón del modo TRG. Como se muestra en la figura 17 la gráfica cuenta con dos imágenes también, la primera con el modo TRG y solapada a esta, la función de transferencia encontrada por la herramienta. El valor arrojado de proximidad para estas dos gráficas es de un 89%, parecido al porcentaje encontrado al STP con cambios grandes de duty, lo que indica que en el modo STP, mientras mas pequeños sean los cambios de duty mas difícil va ser encontrar una función que tenga un mayor grado de proximidad y por último en la segunda imagen obtenemos la respuesta al sistema con la excitación escalón, confirmando los polos a la izquierda del sistema para un sistema estable.

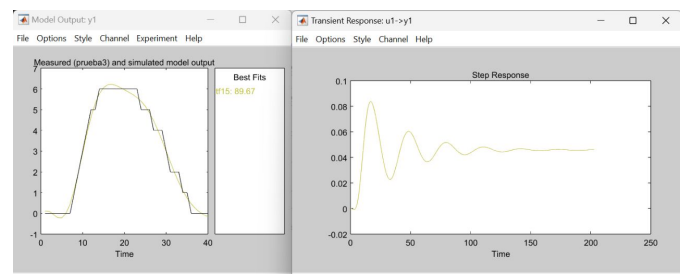


FIGURE 17: Modelo y Respuesta al escalón

V. CONCLUSIONES

- Se encontró que para el caso de los microcontroladores Atmega328P, los puertos análogos, hacen uso en conjunto del mismo convertor análogo digital. Lo que presentó un problema a la hora de utilizar varios sensores que tuvieran entrada análoga para convertirla en digital, ya que se tuvo que tener en cuenta un retardo en la toma de datos, para hacer funcionar la conversión de manera correcta para cada uno de los sensores debido a que si se realizaban en tiempos muy cortos entre un pin de entrada y otro, los datos resultaban erróneos.

- Los niveles de empuje estuvieron por debajo de los esperados para el diseño del prototipo, ya que el valor máximo de la variable es de 8g cuando el motor esta en funcionamiento. El diseño completo estaría construido con 4 motores con iguales características, lo cual hace que el empuje total esperado por el dron sea de 40g, lo cual es muy bajo para lo que se esta buscando, por tanto se plantea la revisión del estado del arte de motores más potentes para la construcción del dron, que en este caso no fueron conseguidos debido a que su costo. Por otra parte el uso de diferentes helices con las que se cuenta no hicieron una diferencia significativa, aunque se usaron con largos similares, es una opción conseguir hélices más grandes cuya área pueda proveer un mayor empuje teóricamente, lo cual se traduce también en mayor costo monetario para este experimento.
- Se recomienda para este tipo de caracterización, sensores de buena calidad que puedan facilitar los procesos matemáticos del micro controlador. Esto debido a que cuando se realizaron las pruebas de los sensores por separado, arrojaba valores diferentes a cuando se uso el sistema en conjunto. La poca falta de memoria del microcontrolador limita la capacidad de procesamiento, y por tanto al aumentar la cantidad de variables sensadas en un corto tiempo se disminuye la veracidad de los datos.
-
- Se observa un problema con el sensor LM35 de temperatura el cual es una alta sensibilidad al ruido provocado por el motor. Se hicieron pruebas para determinar la fuente de este comportamiento como el cambio de fuente de alimentación del sensor por una propia, en caso de que fuese un problema de alimentación por parte del arduino y este no pudiera proveer la corriente necesaria para el sensor, pero el diagnostico no fue acertado, debido a que medida de que el motor aumenta su velocidad los valores de temperatura se alejan cada vez más del valor real de medición, llegando a 50 grados tan solo midiendo el valor de la temperatura ambiente. Al parecer es un problema de ruido proveniente del motor hacia el microcontrolador.
-

ado de <https://controlautomaticoeducacion.com/arduino/interrupciones\protect\discretionary{\char\hyphenchar\font}{}{}arduino/>
 [6] System Identification Toolbox - MathWorks. Recuperado de <https://la.mathworks.com/products/sysid.html>

...

REFERENCES

- [1] Amplificador No inversor - Amplificadores operacionales - Solucion Ingenieril. Recuperado de http://solucioningenieril.com/amplificadores_operacionales/amplificador_no_inversor
- [2] LM35 - El sensor de temperatura más popular - HeTPro-Tutoriales. (2017). Recuperado de <https://hetpro-store.com/TUTORIALES/lm35/#:~:text=El%20LM35%20es%20un%20circuito,se%20puede%20medir%20la%20temperatura.>
- [3] Tutorial transmisor de celda de carga HX711, Balanza Digital. Recuperado de https://naylampmechatronics.com/blog/25_tutorial-trasmisor-de-celda-de-carga-hx711-balanza-digital.html
- [4] Sensor de proximidad fotoeléctrico Infrarrojo E18-D80NK. (2023). Recuperado de <https://naylampmechatronics.com/sensores-proximidad/236-sensor-de-proximidad-fotoelectrico-infrarrojo-e18-d80nk.html>
- [5] C, S. (2019). Interrupciones Arduino - [Que són y como usarlas]. Recuper-