**Universidad Nacional de Trujillo**

**Facultad de Ingeniería**

**ESCUELA DE INGENIERÍA MECATRÓNICA**

****

**ABSTRACCIÓN DE UN MOTOR DC EN UN MODELO DE MACHINE LEARNING PARA SINTONIZAR UN CONTROLADOR PID DE VELOCIDAD APLICANDO ALGORITMOS GENÉTICOS**

**TESIS**

**para optar el título profesional de**

**INGENIERO MECATRÓNICO**

**Autor:**

Bach. Pereyra Gonzáles, Irvin Jair

**Asesor:**

Mg. Ing. León Lescano, Edward Javier

**Trujillo, Perú**

**2018**

# **DEDICATORIA**

# **AGRADECIMIENTO**

# **RESUMEN**

# **ABSTRACT**

# **INTRODUCCIÓN**

## **Realidad problemática**

El control Proporcional-Integral-Derivativo (PID) es el algoritmo de control más comúnmente utilizado en la industria actual(Kocur, Kozak and Dvorscak 2014) y ha sido el método predominante desde el inicio de los sistemas de automatización (Vilanova and Visioli 2012). El impacto ha sido tal que más del 90% de todos los circuitos de control son PID. Y se utiliza para una amplia gama de problemas: control de procesos, accionamientos de motores, memorias magnéticas y ópticas, automotriz, control de vuelo, instrumentación, etcétera. Por lo que es bastante razonable predecir que el control PID continuará usándose en el futuro. La realimentación ha tenido una influencia revolucionaria en prácticamente todas las áreas donde se ha utilizado y continuará haciéndolo (Åström and Hägglund 2001).

La razón principal de esta popularidad es la facilidad de ajuste y puesta en servicio de controladores PID SISO. Considerando además que la mayoría de sistemas utilizados son SISO. El PID proporciona un algoritmo óptimo para perturbaciones no medidas y dinámicas desconocidas, el caso común en la manufactura. A pesar de su popularidad, el foco principal de la investigación en esta área se concentra principalmente en el desarrollo de metodologías de ajuste de los tres parámetros de control PID. El ajuste mencionado siempre es un tema de gran interés práctico y, a pesar del pequeño número de parámetros ajustables, sigue siendo tema de investigación (Vilanova and Visioli 2012).

Aún la mayoría de documentos sobre control tratan el método de Ziegler-Nichols como punto de referencia para sintonizar controladores PID. Esta es una situación muy insatisfactoria porque las reglas de Ziegler-Nichols son conocidas por dar resultados muy pobres en muchos casos (Vilanova and Visioli 2012). Es muy fácil demostrar que cualquier controlador con un ajuste razonable superará a un PID con el ajuste por Ziegler-Nichols. Incluso muchas estrategias propuestas pueden eliminarse fácilmente si se comparan con un PID bien ajustado (Åström and Hägglund 2001).

En las últimas dos décadas, los procesos modernos a gran escala han experimentado grandes desafíos y se han vuelto cada vez más complicados en todos los sectores industriales posibles (Yin et al. 2015). El rápido crecimiento de la complejidad de plantas de procesos modernos, tanto en términos de flujo de materiales como de intercambio de energía, ha incrementado sustancialmente el número de circuitos de control en lazo cerrado para mantener las condiciones de producción y calidad deseadas del producto (Åström and Hägglund 2001), Como resultado, los enfoques tradicionales basados en modelos, que requieren el conocimiento previo necesario del proceso o modelo físico obtenido a partir de principios primarios, se han vuelto poco prácticos, especialmente para la industria a gran escala (Yin et al. 2015).

Frente a esto, se puede optar por métodos inteligentes para expandir las capacidades de un controlador, y un PID mejorado incrementará la eficiencia (reducir materias primas, servicios públicos y desechos), flexibilidad, operabilidad, mantenimiento, rentabilidad y seguridad, que determina el cumplimiento y la competitividad. El uso innovador de la capacidad de desarrollo y expansión del PID es la clave para la fabricación sostenible (Åström and Hägglund 2001).

En plantas de control con una fuerte no linealidad, se han propuesto métodos de ajuste de parámetros PID que utilizan redes neuronales y algoritmos genéticos, aunque con ciertas limitaciones. Actualmente hay pocos esquemas de ajuste de parámetros PID de uso práctico para sistemas no lineales (Funes et al. 2015). Respecto de estos nuevos métodos, se muestran controladores sintonizados por medio de algoritmos evolutivos que superan de forma amplia los métodos clásicos (Marin Cano, Hernandez Rivero y Jimenez Builes 2018; Colorado Arellano et al. 2018). Así también, se implementaron algoritmos bio-inspirados obteniendo mejores resultados en comparación con los métodos tradicionales. Destacando su efectividad en plantas con retardos y de orden superior (López Mora 2014). En una realidad más cercana, en la escuela de Ingeniería Mecatrónica de la Universidad Nacional de Trujillo, Garcia Pereda (2018) aplicó un algoritmo de colonia de hormigas y en otro trabajo, Balcázar Llanos (2017), un algoritmo genético, para sintonizar controladores PID obteniéndose en ambos casos resultados sobresalientes frente a métodos tradicionales. En todos los casos mencionados anteriormente se necesitó realizar una identificación de la función de transferencia de la planta antes del proceso de sintonización. Sin embargo; la principal limitación de las funciones de transferencia es que solo pueden usarse para diseñar controladores de sistemas lineales o de sistemas no lineales que permiten construir sus aproximaciones lineales alrededor de un punto de equilibrio de interés (Åström y Murray 2006).

Por otro lado, debido al crecimiento significativo del grado de automatización en plantas modernas, se generan una gran cantidad de datos de proceso (Yin et al. 2015), que pueden ser aprovechados por los métodos desarrollados en el campo de machine learning y las capacidades computacionales actuales. Por ejemplo, Burgos Molina y Chacón Pacheco (2014) presentan el modelamiento de fenómenos dinámicos en el tiempo a través de una máquina de soporte vectorial. Así también, Bautista Thompson, Guzmán Ramírez and Figueroa Nazuno (2004) utilizan una máquina de soporte vectorial para abordar un problema de predicción de series de tiempo de múltiples puntos con resultados positivos. De la misma manera en otro estudio se describen las importantes ventajas teóricas y prácticas de la regresión de soporte vectorial en la identificación de sistemas de caja negra. La simplicidad de su implementación y su buen rendimiento respecto a la predicción de series temporales, hace de este método una alternativa atractiva frente a las técnicas estándar de identificación de sistemas (Gretton et al. 2002). Drezet y Harrison (1998) mencionan que las propiedades inherentes de las máquinas de soporte vectorial cumplen la mayoría de los requisitos para la identificación de sistemas lineales y no lineales de propósito general. Además, Schölkopf y Smola (2001) mencionan que la estimación regularizada se ha utilizado ampliamente en aplicaciones relacionadas a la estadística y el aprendizaje automático. Pero en los últimos años, se ha explotado en la identificación y predicción de sistemas no lineales, recurriendo principalmente a núcleos gaussianos o polinomiales.

Por lo tanto, se presenta un escenario idóneo en el que los procesos de sintonización de controladores usando algoritmos genéticos podrían utilizar modelos de machine learning que ofrezcan muchas más prestaciones y capacidades al momento de identificar una planta de control simple o compleja

## **Formulación del problema**

¿Cómo mejorar las prestaciones de los métodos tradicionales de identificación de sistemas, en procesos de sintonización de controladores PID basados en algoritmos genéticos?

## **Justificación del estudio**

### **Relevancia Tecnológica**

Las capacidades computacionales de la tecnología actual y el avance de métodos de optimización e identificación de sistemas, presentan un escenario ideal para abordar problemáticas entorno al control de procesos, como es la sintonización de controladores PID, desde nuevos enfoques que integren técnicas que se aplicaban comúnmente de forma separada y en otros contextos.

### **Relevancia Institucional**

Resolver problemas de control e identificación de sistemas, son campos todavía en desarrollo, por lo que dar soluciones desde nuevos enfoques como los mencionados en este trabajo, permitirá impulsar la producción científica en la Escuela de Ingeniería Mecatrónica.

### **Relevancia Social**

La integración de técnicas modernas, como las mencionadas en este trabajo, pueden ser un aporte para dar inicio a la exploración de nuevas soluciones a los problemas que los estudiantes o cualquier persona interesada en los sistemas de control, se enfrentan a menudo.

### **Relevancia Económica**

Todas las herramientas utilizadas en este trabajo son de uso y distribución libre, por lo que este trabajo permite la apertura a la colaboración y customización que las herramientas como el software privativo no admiten.

### **Relevancia Ambiental**

El desarrollo de mejores controladores implica optimizar los procesos de control, lo cual, a su vez, está ligado a un ahorro de recursos en las acciones industriales, científicas o didácticas, contribuyendo a la preservación del ambiente.

### **Antecedentes**

CHIA-JU WU propone el uso de una red neuronal para tomar el lugar de la planta de control en el procedimiento de ajuste de su controlador PID usando algoritmos genéticos. De esta manera, fuera de línea, las ganancias PID se determinan y posteriormente se aplican a la planta para el control en línea. Luego, se realiza un nuevo ajuste en línea de los parámetros PID aplicando algoritmos genéticos. Mostrando resultados alentadores pero señalando la necesidad de posteriores esfuerzos en el desarrollo de estos conceptos.

Mohd S. Saad, Hishamuddin Jamaluddin e Intan Z. M. Darus (2012) presentan la implementación de un sintonizador de controlador PID usando técnicas de algoritmos genéticos y evolución diferencial, cuyo desempeño es evaluado usando los criterios de la integral del error absoluto y el error cuadrático medio. Además, comparan el desempeño del controlador PID sintonizado, frente al obtenido mediante el método de Ziegler-Nichols, obteniendo resultados positivos.

Christoph Hametner, Christian H. Mayr, Martin Kozek and Stefan Jakubek presentan un método para diseñar controladores PID usando redes neuronales modulares LMN y algoritmos evolutivos, aunque indicando la limitación que supone el cálculo computacional de la computadora utilizada.

Simon Hissem, Mamadou Lamine Doumbia, Mohamed Keddar presentan un nuevo método de sistema de caja negra para aproximar sistemas de alto órden y diseñar controladores, sin necesidad del uso de funciones de transferencia, y el diseño se basa únicamente en elementos ponderados que caracterizan la respuesta de salida de estos sistemas. Desarrollaron y aplicaron un controlador PI al sistema real. Se investigó el rendimiento de este nuevo enfoque mediante el criterio del la integral del error cuadrático (ISE) y el método de aproximación de Pade, mostrando resultados positivos.

S. Strmcnik, M. Sega, J. PetrovCic y P. Tramte presentan, en relación con métodos de simulación y optimización, el uso parcial de un modelo de caja negra, demostrando ser una herramienta eficiente para el control, incluso, de algunos sistemas no lineales. Sin embargo, señalan que para el uso exitoso de esta herramienta se requiere al menos una comprensión cualitativa del sistema en consideración.

Michel Fliess, Cédric Join, Hebertt Sira-Ramire muestran un enfoque de control predictivo basado en modelos de caja negra, indicando su potencial uso en plantas industriales donde es difícil lograr una identificación paramétrica y que aún se debe seguir investigando para hacer que los controladores universales sea más precisos.

## **Objetivos**

### **General**

Usar algoritmos genéticos para sintonizar un controlador PID de una planta de control, a través de su abstracción en un modelo de machine learning entrenado con datos temporales.

### **Específicos**

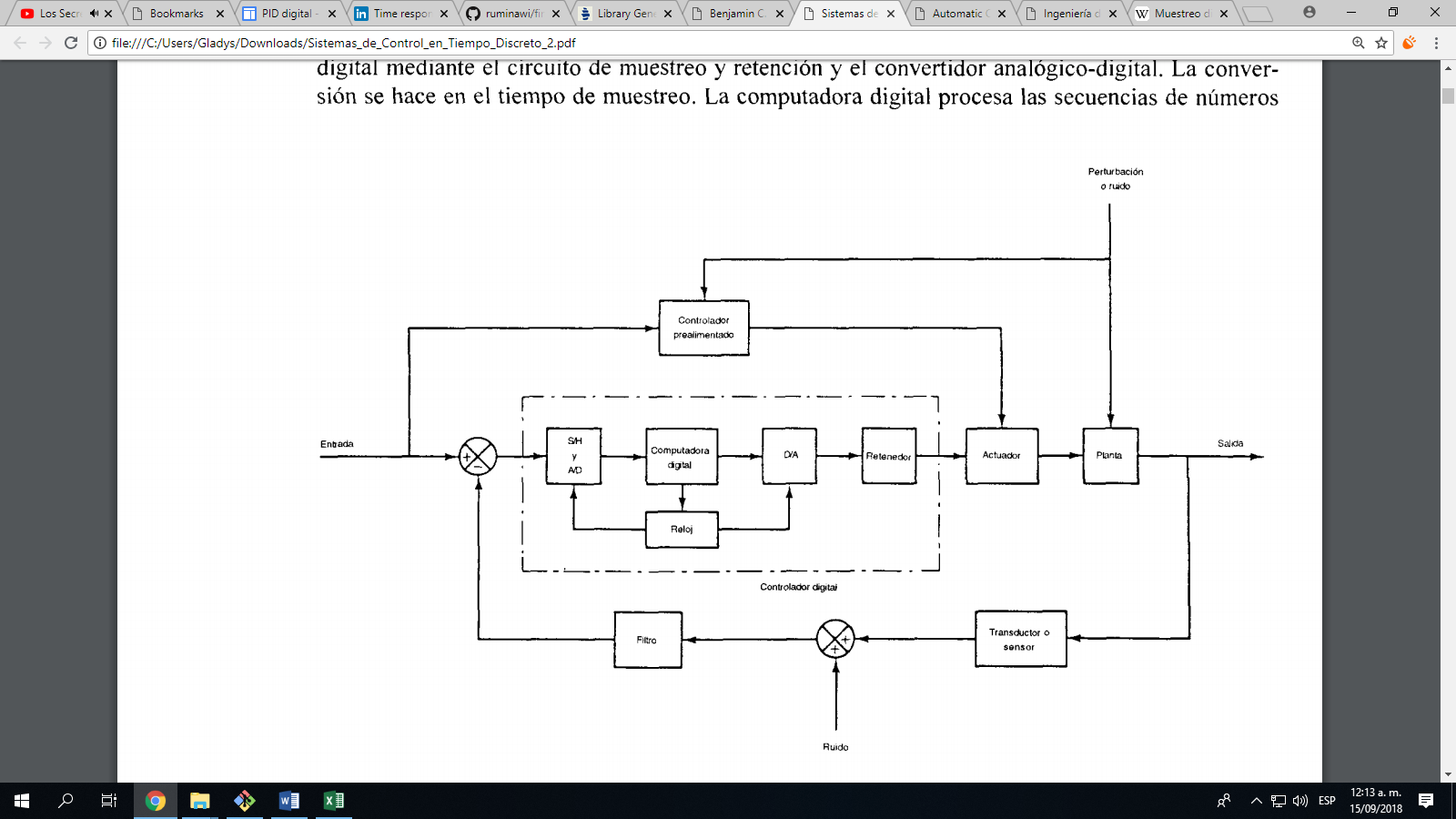
* Implementar la planta motor DC del sistema de control de velocidad.
* Elegir una señal de entrada y obtener un conjunto adecuado de datos experimentales en tiempo discreto para identificar la planta de control por medio de un modelo de machine learning.
* Elegir un método de regresión para abstraer la planta de control en un modelo de machine learning y definir sus restricciones.
* Implementar un controlador PID discreto y definir su evaluación de desempeño en base a la integral del tiempo multiplicada por valor absoluto del error y su ejecución sobre el modelo de machine learning.
* Sintonizar el controlador PID discreto aplicado al modelo de machine learning por medio de un algoritmo genético y comprobar su desempeño en la planta real.

## **Marco teórico**

## **Sistema de control digital**

La operación del controlador se maneja por el reloj. En dicho sistema de control digital, en algunos puntos del sistema pasan señales de amplitud variable ya sea en tiempo continuo o en tiempo discreto, mientras que en otros pasan señales codificadas en forma numérica, como se muestra en la figura.

***Figura 1.1:*** *Diagrama de bloques de un sistema de control digital que muestra las señales en forma binaria o gráfica.*



***Fuente:*** *(Ogata, 1995)*

La salida de la planta es una señal en tiempo continuo. La señal de error se convierte a forma digital mediante el circuito de muestreo y retención y el convertidor analógico-digital. La conversión se hace en el tiempo de muestreo. La computadora digital procesa las secuencias de números por medio de un algoritmo y produce nuevas secuencias de números. En cada instante de muestreo se debe convertir un número codificado (en general un número binario que consiste en ocho o más dígitos binarios) en una señal física de control, la cual normalmente es una señal en tiempo continuo o analógica. El convertidor digital-analógico y el circuito de retención convierten la secuencia de números en código numérico a una señal continua por secciones. El reloj en tiempo real de la computadora sincroniza los eventos. La salida del circuito de retención, una señal en tiempo continuo, se alimenta a la planta, ya sea de manera directa o a través de un actuador, para controlar su dinámica.

La operación que transforma las señales en tiempo continuo en datos en tiempo discreto se denomina muestreo o discretización. La operación inversa, que transforma datos en tiempo discreto en una señal en tiempo continuo, se conoce como retención de datos; ésta realiza la reconstrucción de la señal en tiempo continuo a partir de la secuencia de datos en tiempo discreto. Esto por lo regular se logra al utilizar alguna de las muchas técnicas de extrapolación. En la mayoría de los casos esto se realiza manteniendo constante la señal entre los instantes de muestreo sucesivos.

El circuito de muestreo y retención (S/H, del inglés Sample-and-Hold) y el convertidor analógico-digital (A/D) convierten la señal en tiempo continuo en una secuencia de palabras binarias codificadas numéricamente. Dicho proceso de conversión A/D se conoce como codificación. La combinación del circuito S/H y el convertidor analógico-digital se puede visualizar como un interruptor que cierra instantáneamente en cada intervalo de tiempo T y genera una secuencia de números en código numérico. La computadora digital procesa dichos números en código numérico y genera una secuencia deseada de números en código numérico. El proceso de conversión digital-analógico (D/A) se denomina decodificación. (Ogata, 1995)

## **Controlador automático**

Un controlador automático compara el valor real de la salida de una planta con la entrada de referencia (el valor deseado), determina la desviación y produce una señal de control que reduce la desviación a cero o a un valor pequeño. La manera en la cual el controlador automático produce la señal de control se denomina acción de control. La Figura 2-6 es un diagrama de bloques de un sistema de control industrial que consiste en un controlador automático, un actuador, una planta y un sensor (elemento de medición).

***Figura 1.2:*** *Diagrama de bloques de un sistema de control industrial, formado por un controlador automático, un actuador, una planta y un sensor (elemento de medición).*



***Fuente:*** *(Ogata, 2010)*

El controlador detecta la señal de error que, por lo general, está en un nivel de potencia muy bajo, y la amplifica a un nivel lo suficientemente alto. La salida de un controlador automático se alimenta a un actuador, como un motor o una válvula neumáticos, un motor hidráulico o un motor eléctrico. (El actuador es un dispositivo de potencia que produce la entrada para la planta de acuerdo con la señal de control, a fin de que la señal de salida se aproxime a la señal de entrada de referencia.) El sensor, o elemento de medición, es un dispositivo que convierte la variable de salida en otra variable manejable, como un desplazamiento, una presión o un voltaje, que pueda usarse para comparar la salida con la señal de entrada de referencia. Este elemento está en la trayectoria de realimentación del sistema en lazo cerrado. El punto de ajuste del controlador debe convertirse en una entrada de referencia con las mismas unidades que la señal de realimentación del sensor o del elemento de medición.

**Acción de control proporcional**

Para un controlador con acción de control proporcional, la relación entre la salida del controlador u(t) y la señal de error e(t) es:

donde Kp se considera la ganancia proporcional. Cualquiera que sea el mecanismo real y la forma de la potencia de operación, el controlador proporcional es, en esencia, un amplificador con una ganancia ajustable.

**Acción de control integral**

En un controlador con acción de control integral, el valor de la salida del controlador u(t) se cambia a una razón proporcional a la señal de error e(t). Es decir,

donde Ki es una constante ajustable.

**Acción de control proporcional-integral**

La acción de control de un controlador proporcional-integral (PI) mejora la estabilidad relativa y el error en estado estable al mismo tiempo, pero el tiempo de levantamiento se incrementa. Se define mediante

donde Ti se denomina tiempo integral.

**Acción de control proporcional – derivativa:**

La acción de control de un controlador proporcional-derivativa (PD) es en esencia anticipativa, mejora el amortiguamiento y tendrá efecto en el error de estado estable sólo si el error varía con respecto al tiempo. Se define mediante:

donde Td es el tiempo derivativo.

**Acción de control proporcional-integral-derivativa**

La combinación de la acción de control proporcional, la acción de control integral y la acción de control derivativa se denomina acción de control proporcional-integral-derivativa. Esta acción combinada tiene las ventajas de las acciones de control PI y PD. La ecuación de un controlador con esta acción combinada está dada por:

donde Kp es la ganancia proporcional, Ti es el tiempo integral y Td es el tiempo derivativo. (Ogata, 2010)

## **Controlador PID discreto**

Sistemas de control automático, Bnejamin C. Kuo, 7ma Ed., Prentice Hal Hispanoamericana S. A. Naucalpan de Juarez, Edo. De México.

El controlador PID en el dominio en tiempo continuo se describe como:

El componente proporcional se implementa en forma digital mediante una ganancia constante . Ya que un computadora digital o procesador tiene una longitud de palabra finita, la constante no puede realizarse con resolución infinita.

La derivada con respecto al tiempo de una función en se puede aproximar mediante la regla de diferencia hacia atrás, empleando los valores de medidos en y ; esto es:

La función de transferencia del diferenciador digital es:

Donde es la constante proporcional de la derivada del controlador. Y en general, la selección del periodo de muestreo es muy importante, el valor de T debe ser lo suficientemente pequeño, para que la aproximación digital sea exacta.

La regla de integración trapezoidal, aproxima el área bajo la función mediante una serie de trapezoides. Sea la integral de evaluado en designado como . Entonces:

La función de transferencia del integrador digital usando la regla trapezoidal es:

Donde es la constante proporcional de la derivada del controlador.

Al combinar la operación proporcional, integral (Método trapezoidal) y derivativa descritas anteriormente, el controlador digital PID es modelado mediante la siguiente función de transferencia:

Una vez que la función de transferencia de un controlador digital se determina, el controlador puede implementarse mediante un procesador digital o computadora. El operador se interpreta como un tiempo de retardo de segundos. En la práctica, el tiempo de retardo se implementa mediante el almacenamiento de una variable en alguna localidad de memoria en la computadora y entonces se saca después de que han pasado segundos. (Kuo and Aranda Pérez, 1996)

## **Filtro paso bajo**

La forma general en el dominio de Laplace de un filtro paso bajo de primer orden es:

Donde es la frecuencia de corte en y es la frecuencia de corte en .

Para implementar el filtro en un microcontrolador, se debe trabajar con la convolución del bloque del filtro, con un bloque de muestreo y retención de orden cero para evitar señales atrasadas:

Para poder utilizar el filtro en el programa, es necesario cambiar del dominio de Laplace al dominio Zeta:

Escribiendo la ecuación de la señal en tiempo discreto queda:

Donde es el tiempo de muestreo en .

## **Frecuencia de muestreo**

La frecuencia de muestreo es el número de muestras por unidad de tiempo que se toman de una señal en tiempo continuo para producir una secuencia de valores en tiempo discreto. Se expresa en Hercios (Hz).

La frecuencia de muestreo debe cumplir con el teorema de Nyquist – Shannon, que demuestra que la reconstrucción exacta de una señal [periódica](https://es.wikipedia.org/wiki/Onda_peri%C3%B3dica) continua en banda base a partir de sus muestras, es matemáticamente posible si la señal está limitada en banda y la tasa de muestreo es superior al doble de su ancho de banda.

Si la frecuencia más alta contenida en una señal analógica es y la señal se muestrea a , entonces se puede recuperar totalmente a partir de sus muestras mediante la siguiente función de interpolación:

Así, se puede expresar como:

Donde son las muestras de

La frecuencia es llamada la razón de muestreo de Nyquist. La mitad de su valor, , es llamada algunas veces la frecuencia de Nyquist. Si el criterio no es satisfecho, existirán frecuencias cuyo muestreo coincide con otras (el llamado aliasing).

No aporta nada incrementar la tasa de muestreo una vez que ésta cumple el criterio de Nyquist. En la práctica y dado que no existen los filtros analógicos pasa-bajo ideales, se debe dejar un margen entre la frecuencia máxima que se desea registrar y la frecuencia de Nyquist (frecuencia crítica) que resulta de la tasa de muestreo elegida (por ejemplo, para CD-Audio las frecuencias de los componentes a registrar y reproducir oscilan entre 20Hz y 20 kHz y la frecuencia crítica que se elige es de 22,05 kHz; puesto que la razón de muestreo de Nyquist debe ser mayor a 2 x 20KHz = 40 kHz, y se emplea la tasa de 441000 muestras por segundo, es decir 44.1 kHz; un margen del 10% aproximadamente para esta aplicación). Pero este margen es una necesidad que resulta de las limitaciones físicas de un filtro de reconstrucción (o filtro antialiasing) real, y no una consideración que contemple (o deba contemplar) el teorema. (Es.wikipedia.org, 2018)

Por otro lado, el periodo de muestreo y por lo tanto, la frecuencia de muestreo; debe tomar ciertas consideraciones adicionales para la implementación de un controlador PID digital:

La respuesta en lazo cerrado de un sistema controlado por un PID digital va a depender de este período de muestreo. Si este tiempo es demasiado alto, la estabilidad del sistema será menor y el sistema puede llegar a hacerse inestable y no ser controlable. Un método para estimar el período de muestreo consiste en calcular el período de oscilación del sistema en lazo cerrado con una ganancia que provoque oscilaciones. Se tomará el período de muestreo como la décima parte del tiempo o período de oscilación.

En el ejemplo que aparece a continuación se ha aumentado la ganancia proporcional hasta que se mantengan las oscilaciones en la respuesta al escalón. El período de oscilación es entonces de 5.6 segundos y por lo tanto el período de muestreo debe ser menor de 0.56 segundos.

***Figura 1.3:*** *Variación en la ganancia proporcional de un sistema de control de temperatura.*



***Fuente:*** *https://sites.google.com/site/picuino/digital\_pid#TOC-Per-odo-de-muestreo*

Si el sistema es sobre amortiguado y no presenta oscilaciones, el criterio para escoger el tiempo de muestreo partirá de la respuesta al escalón. Como regla general se acepta que T debe ser 10 veces menor que el tiempo de subida del sistema ante un escalón en lazo abierto. Este tiempo de subida se puede calcular como el tiempo que tarda el sistema en subir desde un 10% hasta un 90% del valor final. Por ejemplo, un sistema térmico que exhiba la respuesta al escalón que se muestra a continuación:

***Figura 1.4:*** *Respuesta de un sistema de control de temperatura frente a una entrada tipo escalón.*



***Fuente:*** *https://sites.google.com/site/picuino/digital\_pid#TOC-Per-odo-de-muestreo*

Este sistema tarda en subir desde el 10% hasta el 90% del valor final 21.5 - 3.5 = 18 segundos. Por lo tanto, para este sistema de ejemplo el tiempo de muestreo del controlador PID debe ser como máximo una décima parte de los 18 segundos:

En los dos casos se ha utilizado la misma planta para calcular el tiempo de muestreo. Como puede verse los resultados son muy diferentes. Con el segundo método el tiempo de muestreo es tres veces mayor que con el primero. Por lo tanto, el tiempo de muestreo depende también de la respuesta que se vaya a conseguir y del tipo de sistema. Siempre que se pueda utilizar el primer método, será preferible puesto que calcula tiempos menores y por lo tanto más seguros. (Pardo, 2018)

## **Modulación por ancho de pulso**

La modulación por ancho de pulsos (también conocida como PWM, siglas en inglés de pulse-width modulation) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada, por ejemplo), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el período. Expresado matemáticamente:

D es el ciclo de trabajo

es el tiempo en que la función es positiva (ancho del pulso)

T es el período de la función

***Figura 1.5:*** *Una señal de onda cuadrada de amplitud acotada mostrando el ciclo de trabajo D.*



***Fuente:*** <https://es.wikipedia.org/w/index.php?title=Modulaci%C3%B3n_por_ancho_de_pulsos> &oldid=110019730.

La construcción típica de un circuito PWM se lleva a cabo mediante un comparador con dos entradas y una salida. Una de las entradas se conecta a un oscilador de onda dientes de sierra, mientras que la otra queda disponible para la señal moduladora. En la salida la frecuencia es generalmente igual a la de la señal dientes de sierra y el ciclo de trabajo está en función de la portadora.

La principal desventaja que presentan los circuitos PWM es la posibilidad de que haya interferencias generadas por radiofrecuencia. Estas pueden minimizarse ubicando el controlador cerca de la carga y realizando un filtrado de la fuente de alimentación. (Es.wikipedia.org, 2018)

## **valor eficaz**

Se denomina valor eficaz al valor cuadrático medio de una magnitud eléctrica. El concepto de valor eficaz se utiliza especialmente para estudiar las formas de onda periódicas, a pesar de ser aplicable a todas las formas de onda, constantes o no. En ocasiones se denomina con el extranjerismo RMS (del inglés, root mean square).

El valor eficaz de la tensión es:

El significado físico del valor eficaz es designar el valor de una corriente rigurosamente constante que al circular sobre una determinada resistencia óhmica produciría los mismos efectos caloríficos que dicha corriente variable. De este modo, se establece un paralelismo entre cualquier tipo de corriente variable y la corriente continua que simplifica los cálculos con esta última.

Para un tren de pulsos (Una señal PWM típica):

***Figura 1.6:*** *Señal tipo pulso con periodo T*



***Fuente:*** *https://es.wikipedia.org/w/index.php?title=Valor\_eficaz&oldid=110312485*

Cuya fórmula es:

Su valor eficaz es:

Donde , está expresada en "por unidad" del periodo . (Es.wikipedia.org, 2018)

## **Análisis de la respuesta temporal:**

El análisis de respuesta temporal estudia la respuesta (De un sistema), es decir, el resultado como una función del tiempo.

La respuesta de tiempo total de un sistema de control consiste en respuesta transitoria (respuesta dinámica ) y respuesta en estado estable .

Donde:

* *(t) = respuesta de estado estacionario.*

Antes de proceder con el análisis de la respuesta temporal de un sistema de control, es necesario probar la estabilidad del sistema a través de pruebas indirectas. La naturaleza de la respuesta transitoria de un sistema es dependiente del sistema en sí mismo y no del tipo de entrada. Por lo tanto, se debe analizar la respuesta transitoria con alguna señal de excitación. La señal tipo escalón es la más usada para este propósito. (Bhattacharya, 2009)

## **Señales de excitación**

Como perturbaciones se pueden utilizar muchos tipos diferentes de señales, pero las de tipo escalón, PRBS (pseudo random binary sequence) y multisenos son las preferidas.

Además, las señales de excitación pueden dividirse ampliamente en tres categorías principales, llamadas señales de propósito general, señales de prueba optimizadas y señales dedicadas avanzadas, como se ilustra en la siguiente figura.

***Figura 1.7:*** *Input excitation types in system identification*



***Fuente:*** *(Ramkrishna 2016)*

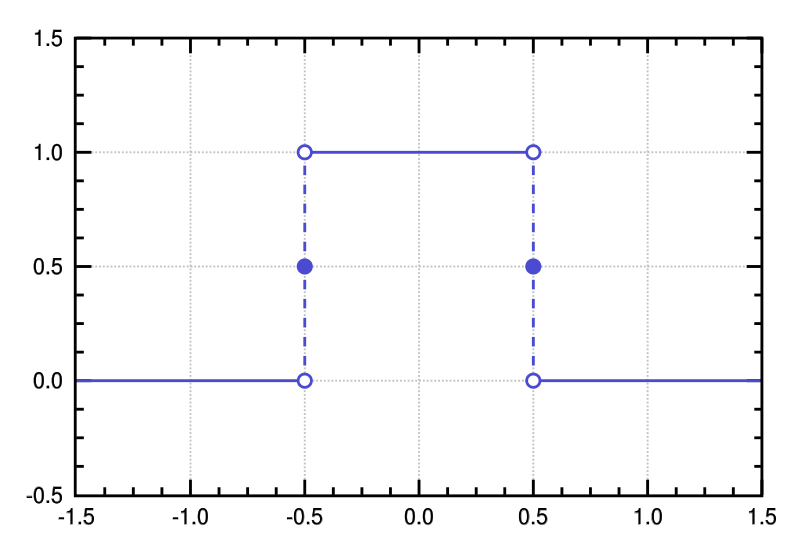
**Señal escalón:**

Una señal tipo escalón está definida por:

Donde es la amplitud de la señal. Debido a su simplicidad es usada extensivamente en la práctica. A menudo, las señales tipo escalón se usan con fines de pre identificación antes de aplicar señales avanzadas para captar la información básica del sistema, como la ganancia estática, las constantes de tiempo, las características de salida, etc. Típicamente, esta información del sistema es considerablemente útil para el diseño de experimentos avanzados (Ramkrishna 2016).

Los parámetros de diseño de una señal tipo escalón son la amplitud de la señal y la duración del escalón.

***Figura 1.8:*** *Parámetros de diseño de una señal tipo escalón.*



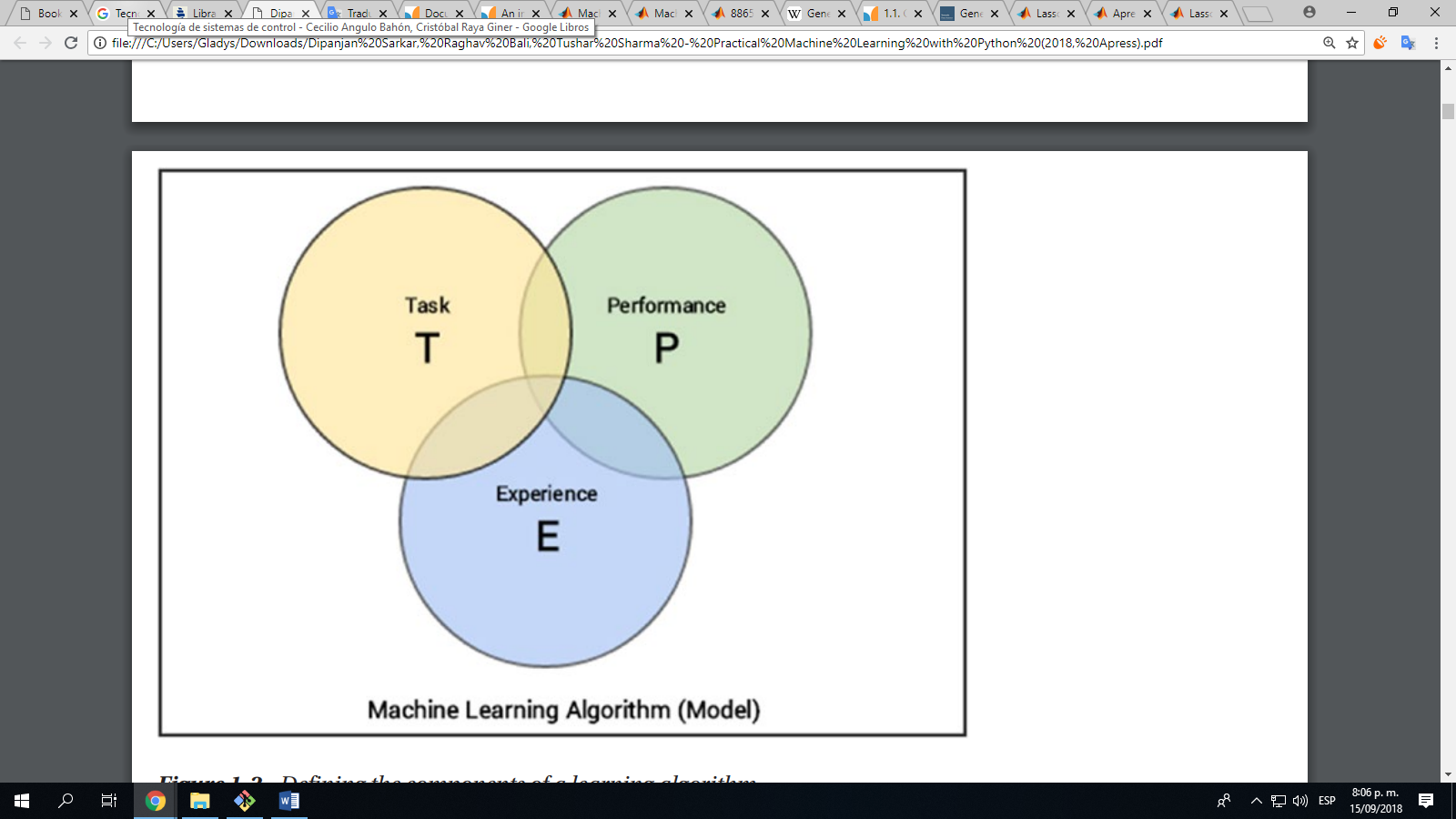
***Fuente:*** *(Ramkrishna 2016)*

## **Machine learning**

La idea de Machine Learning es que habrá algún algoritmo de aprendizaje que ayudará a una máquina a aprender a partir de unos datos.

El profesor Mitchell lo definió de la siguiente manera: "Se dice que un programa de computadora aprende de la experiencia E con respecto a algunas clases de tareas T y una medida de rendimiento P. Si su desempeño en las tareas T, medido por P, mejora con la experiencia E." Estos tres parámetros T, P y E son los componentes principales de cualquier algoritmo de aprendizaje.

***Figura 1.9:*** *Definiendo los componentes de un algoritmo de aprendizaje.*



***Fuente:*** *(Sarkar, BaliSharma 2018****)***

Se puede simplificar la definición como sigue. El machine learning es un campo que consiste de algoritmos de aprendizaje que:

* Mejoran su desempeño P
* Al ejecutar alguna tarea T
* Durante el tiempo sometido a una experiencia E

**Definición de la tarea, T**:

Desde el punto de vista del problema, la tarea, T, es básicamente el problema del mundo real que se debe resolver, que puede ser desde encontrar el mejor marketing o combinación de productos hasta predecir fallas en la infraestructura.

En el mundo del Machine learning, es mejor si puede definir la tarea de la forma más concreta posible, de modo que hable sobre cuál es el problema exacto que planea resolver y cómo puede definir o formular el problema en una tarea específica de Machine learning.

Volviendo a las tareas típicas que podrían clasificarse como tareas de Machine learning, la siguiente lista describe algunas tareas populares:

* Clasificación o categorización
* Regresión
* Detección de anomalías
* Anotación estructurada
* Traducción
* Agrupación o agrupamiento
* Transcripciones

**Definiendo la Experiencia, E**

El proceso de consumir un conjunto de datos que consiste en muestras de datos o puntos de datos tales que un algoritmo o modelo de aprendizaje aprende patrones inherentes se define como la experiencia, E que se gana con el algoritmo de aprendizaje. Cualquier experiencia que obtenga el algoritmo proviene de muestras de datos o puntos de datos y esto puede ocurrir en cualquier punto del tiempo. Puede alimentar las muestras de datos de una sola vez utilizando datos históricos o incluso proporcionar nuevas muestras de datos cada vez que se adquieren. Por lo tanto, la idea de que un modelo o algoritmo gane experiencia generalmente ocurre como un proceso iterativo, también conocido como entrenamiento del modelo. Cuando una máquina realmente aprende, se basa en datos que se le suministran de vez en cuando, lo que le permite adquirir experiencia y conocimiento sobre la tarea a resolver, tales como que puede usar esta experiencia, E, para predecir o resolver la misma tarea, T, en el futuro para puntos de datos no vistos anteriormente.

El aprendizaje automático se trata de aprender algunas propiedades de un conjunto de datos y aplicarlos a nuevos datos. Esta es la razón por la cual una práctica común en el machine learning para evaluar un algoritmo es dividir los datos disponibles en dos conjuntos, uno que llamamos **conjunto de entrenamiento** en el que aprendemos propiedades de datos y otro que llamamos el **conjunto de pruebas** en el que probamos estas propiedades.

**Definiendo el rendimiento, P**

El rendimiento, P, suele ser una medida cuantitativa o métrica que se utiliza para ver qué tan bien el algoritmo o modelo está realizando la tarea, T, con experiencia, E.

Las medidas de rendimiento típicas incluyen exactitud, precisión, recuperación, puntaje F1, sensibilidad, especificidad, tasa de error, tasa de clasificación errónea y muchas más. Las medidas de rendimiento generalmente se evalúan en muestras de datos de entrenamiento (usadas por el algoritmo para ganar experiencia, E) así como en muestras de datos que no se han visto o aprendido anteriormente, que generalmente se conocen como muestras de datos de prueba y validación. La idea detrás de esto es generalizar el algoritmo para que no se vuelva demasiado parcial solo en los puntos de datos de entrenamiento y tenga un buen rendimiento en el futuro en puntos de datos más nuevos. Se hablará más sobre capacitación, validación y datos de prueba cuando hablemos sobre la construcción y validación de modelos. (Sarkar, BaliSharma 2018)

En general, un problema de aprendizaje considera un conjunto de n muestras de datos y luego trata de predecir las propiedades de datos desconocidos. Si cada muestra es más que un número único y, por ejemplo, una entrada multidimensional (también conocida como datos multivariantes), se dice que tiene varios atributos o características (**features**).

Podemos separar los problemas de aprendizaje de acuerdo a los tipos de algoritmos que, en función de su salida, se utilizan como estrategia de resolución:

**Aprendizaje supervisado**, en el cual los datos vienen con atributos adicionales que queremos predecir. Este problema puede ser:

* **clasificación**: las muestras pertenecen a dos o más clases y queremos aprender de los datos ya etiquetados cómo predecir la clase de los datos sin etiqueta.
* **regresión**: si el resultado deseado consiste en una o más variables continuas, entonces la tarea se llama regresión. Un ejemplo de un problema de regresión sería la predicción de la longitud de un salmón en función de su edad y peso. En estadística, el análisis de la regresión es un proceso estadístico para estimar las relaciones entre variables. Incluye muchas técnicas para el modelado y análisis de diversas variables, cuando la atención se centra en la relación entre una variable dependiente y una o más variables independientes (o predictoras). Más específicamente, el análisis de regresión ayuda a entender cómo el valor de la variable dependiente varía al cambiar el valor de una de las variables independientes, manteniendo el valor de las otras variables independientes fijas.

**Aprendizaje no supervisado**, en el cual los datos de entrenamiento consisten en un conjunto de vectores de entrada x sin ningún valor objetivo correspondiente. El objetivo en tales problemas puede ser descubrir grupos de ejemplos similares dentro de los datos, donde se llama agrupamiento, o determinar la distribución de datos dentro del espacio de entrada, conocida como estimación de densidad, o proyectar los datos desde una dimensión alta. espacio hasta dos o tres dimensiones para visualización.

El proceso de aprendizaje tiene como resultado (O salida) un modelo para resolver una tarea dada. Un tipo es el modelo de regresión lineal llamado modelo lineal generalizado en el que se espera que el valor objetivo sea una combinación lineal de las variables de entrada. En noción matemática, si es el valor predicho:

A través del módulo, designamos el vector como coeficientes y como el termino independiente. («An introduction to machine learning with scikit-learn — scikit-learn 0.19.2 documentation» 2018)

## **Elastic Net**

Elastic net es un método de regresión regularizada que puede usarse como un modelo lineal generalizado (Modelo de regresión lineal) en ciertas bibliotecas de lenguajes de programación como python. Es recomendable usar Elastic net cuando se tenga varias variables altamente correlacionadas. Elastic net es un híbrido de regresión Ridge y regularización de Lasso. Al igual que Lasso, Elastic net puede generar modelos reducidos mediante la generación de coeficientes de valor cero. Los estudios empíricos han sugerido que la técnica Elastic net puede superar el Lasso en los datos con predictores altamente correlacionados. Lasso proporciona la regularización de Elastic net cuando establece el par nombre-valor de Alpha a un número estrictamente entre 0 y 1.

**Detalles de Lasso y Elastic net**

Lasso es una técnica de regularización para realizar regresiones lineales. Lasso incluye un término de penalización que restringe el tamaño de los coeficientes estimados. Por lo tanto, se parece a la regresión Ridge. Lasso es un estimador de contracción: genera estimaciones de coeficientes que son parciales. Sin embargo, un estimador de Lasso puede tener un error cuadrático medio menor que un estimador ordinario de mínimos cuadrados cuando lo aplica a datos nuevos. Lasso es una técnica de regularización. Usar Lasso para:

* Reducir el número de predictores en un modelo de regresión.
* Identificar predictores importantes.
* Seleccione entre los predictores redundantes.
* Produzca estimaciones de contracción con errores predictivos potencialmente más bajos que los mínimos cuadrados ordinarios.

A diferencia de la regresión Ridge, a medida que aumenta el término de penalización, Lasso establece más coeficientes a cero. Esto significa que el estimador de lazo es un modelo más pequeño, con menos predictores. Como tal, Lasso es una alternativa a la regresión por pasos y otras técnicas de selección de modelo y reducción de dimensionalidad.

**Definición de Lasso**

La técnica de Lasso resuelve este problema de regularización. Para un valor dado de λ, un parámetro no negativo, Lasso soluciona el problema

N es el número de observaciones.

es la respuesta en la observación i.

xi es data, un vector de valores p en la observación i.

λ es un parámetro de regularización positivo que corresponde a un valor de Lambda.

Los parámetros y son escalares y p-vector respectivamente.

A medida que λ aumenta, disminuye el número de componentes distintos de cero de β.

El problema de Lasso implica la norma L1 de β, en contraste con el algoritmo de red elástica.

**Definición de Elastic Net**

La técnica de Elastic net resuelve este problema de regularización. Para un α estrictamente entre 0 y 1, y un λ no negativo, Elastic net resuelve el problema

Donde

Elastic net es lo mismo que Lasso cuando α = 1. A medida que α se contrae hacia 0, Elastic net se aproxima a la regresión Ridge. Para otros valores de α, el término de penalización interpola entre la norma L1 de β y la norma L2 cuadrada de β. («Lasso and Elastic Net- MATLAB & Simulink- MathWorks America Latina» 2018)

## **Criterio de sintonía de controladores PID**

La intención de un sistema de control realimentado es minimizar el error en la salida, en función del tiempo transcurrido, después de haberse producido una alteración en el sistema. Ambos, la magnitud del error y el tiempo durante el cual existe contribuyen a la definición del control óptimo. Parece lógico proponer un parámetro por medio del cual se caracterice el tiempo de respuesta del sistema por la relación:

Donde es una función de ambos, error y tiempo.

Integrando con respecto al tiempo se obtiene un número característico del tiempo de respuesta del sistema. El criterio de comportamiento será cero solamente si el error es cero durante todo el tiempo, lo cual es imposible cuando se han introducido perturbaciones. Por tanto no desaparecerá, pero cuanto más pequeño sea su valor, mejor será el comportamiento del sistema.

Con el fin de completar la definición matemática del control óptimo, se debe proponer una función de forma que tenga las propiedades primero de ser cero solamente si el error es exactamente cero, y segundo de poseer un mínimo valor consistente con las nociones intuitivas de control óptimo. Puesto que la decisión final de lo que es mejor se basa en juicios o incuso en gustos personales, hay diferentes funciones , utilizadas para definir el óptimo.

La integral del cuadrado del error (ISE) es un criterio de comportamiento propuesto comúnmente para los sistemas de control.

El ISE (Integral Squared Error), es relativamente insensible a pequeños errores, pero los grandes errores contribuyen al fuertemente al valor de la integral. Consecuentemente, utilizando el ISE como criterio de comportamiento dará como resultado una respuesta con pequeños sobrepasamientos pero largos tiempos de estabilización, puesto que los pequeños errores a lo largo del tiempo contribuyen muy poco a la integral.

La integral del valor absoluto del error (Integral Absolute Error) ha sido propuesto con frecuencia como criterio de comportamiento:

Este criterio es más sensible a pequeños errores, pero menos sensible que el ISE a grandes errores.

Grahan y Lathrop (The synthesis of optimun transient response: criteria and estándard forms, Transaction IEEE, nov. 1953), introducen la integral del tiempo multiplicada por el valor absoluto del error (ITAE) como criterio de comportamiento:

ITAE (Integral Time Absolute Error), es insensible a los errores iniciales, y a veces inevitables, pero penalizan fuertemente los errores que permanecen a lo largo del tiempo. La respuesta óptima definida por ITAE, consecuentemente, mostrará tiempos cortos de respuesta total y mayores sobrepasamientos que los otros criterios. (Acedo Sánchez 2006)

## **Algoritmo genético**

Un algoritmo genético es una búsqueda heurística que está inspirada en la teoría de la evolución natural de Charles Darwin. Este algoritmo refleja el proceso de selección natural donde los individuos más aptos son seleccionados para la reproducción con el fin de producir descendencia de la próxima generación.

**Noción de selección natural**

El proceso de selección natural comienza con la selección de individuos más aptos de una población. Producen descendientes que heredan las características de los padres y se agregarán a la generación siguiente. Si los padres tienen una mejor condición física, sus hijos serán mejores que los padres y tendrán más posibilidades de sobrevivir. Este proceso continúa iterando y, al final, se encontrará una generación con las personas más aptas.

Esta noción se puede aplicar a un problema de búsqueda. Consideramos un conjunto de soluciones para un problema y seleccionamos el conjunto de las mejores.

Cinco fases se consideran en un algoritmo genético.

* Población inicial
* Función de la aptitud
* Selección
* Crossover
* Mutación

**Población inicial**

El proceso comienza con un conjunto de individuos que se llama Población. Cada individuo es una solución al problema que desea resolver. Un individuo se caracteriza por un conjunto de parámetros (variables) conocidos como Genes. Los genes se unen en una cuerda para formar un cromosoma (solución). En un algoritmo genético, el conjunto de genes de un individuo se representa utilizando una cuerda, en términos de un alfabeto. Por lo general, se utilizan valores binarios (cadena de 1s y 0s). Decimos que codificamos los genes en un cromosoma.

***Figura 1.10:*** *Población, cromosoma y gen.*

******

***Fuente:*** *https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3*

**Función de fitness**

La función de aptitud determina qué tan buena es una persona (la capacidad de una persona para competir con otras personas). Le da un puntaje de condición física a cada individuo. La probabilidad de que un individuo sea seleccionado para la reproducción se basa en su puntaje de condición física.

**Selección**

La idea de la fase de selección es seleccionar a las personas más aptas y dejarles pasar sus genes a la próxima generación. Se seleccionan dos pares de individuos (padres) en función de sus puntajes de aptitud. Las personas con una buena forma física tienen más posibilidades de ser seleccionadas para la reproducción.

**Crossover**

Crossover es la fase más importante en un algoritmo genético. Para cada par de padres que se aparearán, se elige al azar un punto de cruce dentro de los genes. Por ejemplo, considere el punto de cruce como 3 como se muestra a continuación.

***Figura 2.11:*** *Punto de crossover.*



***Fuente:*** *https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3.*

**Crossover**

Los descendientes se crean intercambiando los genes de los padres entre ellos hasta que se alcanza el punto de cruce.

***Figura 2.12:*** *Intercambio de genes entre los padres (Crossover).*



***Fuente:*** *https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3.*

Los nuevos descendientes se agregan a la población.

***Figura 2.13:*** *Nueva descendencia*



***Fuente:*** *https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3.*

**Mutación**

En ciertos descendientes nuevos formados, algunos de sus genes pueden ser sometidos a una mutación con una baja probabilidad aleatoria. Esto implica que algunos de los bits en la cadena de bits se pueden voltear.

***Figura 2.14:*** *Antes y después de una mutación.*

**

***Fuente:*** *https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3.*

La mutación ocurre para mantener la diversidad dentro de la población y prevenir la convergencia prematura.

**Terminación**

El algoritmo finaliza si la población ha convergido (no produce descendencia que sea significativamente diferente de la generación anterior).

Luego se dice que el algoritmo genético ha proporcionado una serie de soluciones a nuestro problema.

**Comentarios**

La población tiene un tamaño fijo. A medida que se forman las nuevas generaciones, mueren las personas con el menor nivel de aptitud física, lo que proporciona espacio para nuevos descendientes.

La secuencia de fases se repite para producir individuos en cada nueva generación que son mejores que la generación anterior. (Mallawaarachchi 2018)

## **ARDUINO**

Arduino, es una compañía open source y open hardware, así como un proyecto y comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales y dispositivos interactivos que puedan sensar y controlar objetos del mundo real. Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios. Los productos que vende la compañía son distribuidos como Hardware y Software Libre, bajo la Licencia Pública General Reducida de GNU (LGPL) o la Licencia Pública General de GNU (GPL),1​permitiendo la manufactura de las placas Arduino y distribución del software por cualquier individuo. Las placas Arduino están disponibles comercialmente en forma de placas ensambladas o también en forma de kits hazlo tú mismo (Del inglés DIY: "Do It Yourself"). («Es.wikipedia.org» 2018)

***Figura 1.15:*** *Arduino IDE****.***



***Fuente:*** *https://upload.wikimedia.org/wikipedia/commons/thumb/a/a1/Arduino\_IDE\_-\_Blink.png/1024px-Arduino\_IDE\_-\_Blink.png*

***Figura 1.16:*** *Arduino Uno R3*



***Fuente***: *https://cdn.sparkfun.com//assets/parts/6/8/1/6/11224-04.jpg*

## **The jupyter notebook**

Jupyter Notebook es una aplicación web de código abierto que permite crear y compartir documentos que contengan códigos, ecuaciones, visualizaciones y textos narrativos en vivo. Los usos incluyen: limpieza y transformación de datos, simulación numérica, modelado estadístico, visualización de datos, aprendizaje automático y mucho más. («Jupyter.org» 2018)

**Notebook document**

Notebook document (o "documents") son documentos producidos por la aplicación Jupyter Notebook, que contiene tanto código de computadora (por ejemplo, python) como elementos de texto enriquecido (párrafos, ecuaciones, figuras, enlaces, etc.). Los documentos del cuaderno son documentos legibles por humanos que contienen la descripción del análisis y los resultados (figuras, tablas, etc.) así como documentos ejecutables que se pueden ejecutar para realizar análisis de datos.

**Aplicación de Jupyter Notebook**

La aplicación Jupyter Notebook es una aplicación cliente-servidor que permite editar y ejecutar documentos portátiles a través de un navegador web. La aplicación Jupyter Notebook se puede ejecutar en un escritorio local que no requiere acceso a Internet (como se describe en este documento) o puede instalarse en un servidor remoto y acceder a través de Internet.

Además de mostrar/editar/ejecutar Notebook documents, la aplicación de Jupyter Notebook tiene un " Dashboard " (Notebook Dashboard), un "panel de control" que muestra los archivos locales y permite abrir Notebook documents o cerrar sus núcleos.

**Kernel**

Un kernel de notebook es un "motor de computación" que ejecuta el código contenido en un documento de Notebook. El núcleo ipython, por ejemplo, ejecuta código python. Existen núcleos para muchos otros lenguajes (kernels oficiales).

Cuando abre un Notebook document, el kernel asociado se inicia automáticamente. Cuando se ejecuta el Notebook (celda por celda o con el menú Celda -> Ejecutar todo), el kernel realiza el cálculo y produce los resultados. Dependiendo del tipo de cálculos, el kernel puede consumir CPU y RAM significativas. Tenga en cuenta que la memoria RAM no se libera hasta que el kernel se apaga.

**Notebook Dashboard**

El Notebook Dashboard es el componente que se muestra primero al iniciar la aplicación Jupyter Notebook. Notebook Dashboard se usa principalmente para abrir Notebook documents y para administrar los kernels en ejecución (visualizar y cerrar).

El panel de instrumentos del Notebook tiene otras características similares a un administrador de archivos, a saber, navegación de carpetas y cambio de nombre / eliminación de archivos. (Ingargiola y colaboradores 2015)

## **Marco conceptual**

## **Diagrama de bloques**

Representación gráfica, relacionada a un sistema, de las funciones que lleva a cabo cada componente y el flujo de señales. También se muestran las relaciones existentes entre los diversos componentes. («Es.wikipedia.org» 2018)

## **Sistema de control en lazo cerrado**

En un sistema de control en lazo cerrado, se alimenta al controlador la señal de error de actuación, que es la diferencia entre la señal de entrada y la señal de realimentación (que puede ser la propia señal de salida o una función de la señal de salida y sus derivadas y/o integrales), con el fin de reducir el error y llevar la salida del sistema a un valor deseado. El término control en lazo cerrado siempre implica el uso de una acción de control realimentado para reducir el error del sistema. («Es.wikipedia.org» 2018)

## **Variable controlada**

La variable controlada es la cantidad o condición que se mide y controla. Normalmente, la variable controlada es la salida del sistema. («Es.wikipedia.org» 2018)

## **Señal de control o variable manipulada**

La señal de control o variable manipulada es la cantidad o condición que el controlador modifica para afectar el valor de la variable controlada. («Es.wikipedia.org» 2018)

## **Plantas**

Una planta puede ser una parte de un equipo, tal vez un conjunto de los elementos de una máquina que funcionan juntos, y cuyo objetivo es efectuar una operación particular. Dentro de un sistema de control, una planta es cualquier objeto físico que se va a controlar. («Es.wikipedia.org» 2018)

## **Perturbaciones**

Una perturbación es una señal que tiende a afectar negativamente el valor de la salida de un sistema. Si la perturbación se genera dentro del sistema se denomina interna, mientras que una perturbación externa se genera fuera del sistema y es una entrada. («Es.wikipedia.org» 2018)

## **Control realimentado.**

El control realimentado se refiere a una operación que, en presencia de perturbaciones, tiende a reducir la diferencia entre la salida de un sistema y alguna entrada de referencia, y lo realiza tomando en cuenta esta diferencia. («Es.wikipedia.org» 2018)

## **Muestreo digital**

El muestreo digital es una de las partes del proceso de digitalización de las señales. Consiste en tomar muestras de una señal analógica a una frecuencia o tasa de muestreo constante, para cuantificarlas posteriormente. El muestreo está basado en el teorema de muestreo, que es la base de la representación discreta de una señal continua en banda limitada. Es útil en la digitalización de señales (y por consiguiente en las telecomunicaciones) y en la codificación del sonido en formato digital. («Es.wikipedia.org» 2018)

## **Bit**

Bit es el acrónimo de Binary digit.​ Un bit es un dígito del sistema de numeración binario. La capacidad de almacenamiento de una memoria digital también se mide en bits, pues esta palabra tiene varias acepciones. Un bit o dígito binario puede representar uno de esos dos valores: 0 o 1. («Es.wikipedia.org» 2018)

## **resolución digital**

La resolución viene determinada por la longitud de la palabra digital (número de bits), es decir por las agrupaciones de ceros y unos con que se va componiendo (codificando) la señal. («Es.wikipedia.org» 2018)

## **Hardware**

La palabra hardware en informática se refiere a las partes físicas tangibles de un sistema informático; sus componentes eléctricos, electrónicos, electromecánicos y mecánicos.1​ Los cables, así como los gabinetes o cajas, los periféricos de todo tipo, y cualquier otro elemento físico involucrado, componen el hardware o soporte físico; contrariamente, el soporte lógico e intangible es el llamado software. («Es.wikipedia.org» 2018)

## **Software**

Se conoce como software1​ al soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware. La interacción entre el software y el hardware hace operativo un ordenador (u otro dispositivo), es decir, el Software envía instrucciones que el Hardware ejecuta, haciendo posible su funcionamiento. («Es.wikipedia.org» 2018)

## **Velocidad angular**

La velocidad angular es una medida de la velocidad de rotación. Se define como el ángulo girado por una unidad de tiempo y se designa mediante la letra griega ω. Su unidad en el Sistema Internacional es el radián por segundo (rad/s). («Es.wikipedia.org» 2018)

## **Tiempo de establecimiento**

## En la teoría de control, el tiempo de establecimiento de un sistema dinámico tal como un amplificador u otro dispositivo de salida es el tiempo transcurrido desde la aplicación de una entrada escalonada instantánea ideal hasta el momento en que la salida del amplificador ha entrado y permanece dentro de una banda de error especificada. («Es.wikipedia.org» 2018)

## **Adquisición de datos**

La adquisición de datos o adquisición de señales consiste en la toma de muestras del mundo real (sistema analógico) para generar datos que puedan ser manipulados por un ordenador u otros dispositivos electrónicos (sistema digital). Consiste en tomar un conjunto de señales físicas, convertirlas en tensiones eléctricas y digitalizarlas de manera que se puedan ser procesadas por una computadora o PAC. Se requiere una etapa de acondicionamiento, que adecua la señal a niveles compatibles con el elemento que hace la transformación a señal digital. El elemento que hace dicha transformación es el módulo de digitalización o tarjeta de adquisición de datos (DAQ). («Es.wikipedia.org» 2018)

## **Comunicación serie**

El puerto serial envía y recibe bytes de información un bit a la vez. Aun y cuando esto es más lento que la comunicación en paralelo, que permite la transmisión de un byte completo por vez, este método de comunicación es más sencillo y puede alcanzar mayores distancias. Típicamente, la comunicación serial se utiliza para transmitir datos en formato ASCII. Para realizar la comunicación se utilizan 3 líneas de transmisión: Tierra (o referencia), Transmitir, Recibir. Debido a que la transmisión es asincrónica, es posible enviar datos por una línea mientras se reciben datos por otra. («Comunicación Serial: Conceptos Generales - National Instruments» 2018)

## **Velocidad de transmisión**

Indica el número de bits por segundo que se transfieren, y se mide en baudios (bauds). Por ejemplo, 300 baudios representan 300 bits por segundo. Cuando se hace referencia a los ciclos de reloj se está hablando de la velocidad de transmisión. Por ejemplo, si el protocolo hace una llamada a 4800 ciclos de reloj, entonces el reloj está corriendo a 4800 Hz, lo que significa que el puerto serial está muestreando las líneas de transmisión a 4800 Hz. («Comunicación Serial: Conceptos Generales - National Instruments» 2018)

## **UART**

UART es normalmente un circuito integrado individual usado para comunicaciones de un sistema de cómputo, son normalmente incluidas en microcontroladores. Las funciones principales de chip UART son: manejar las interrupciones de los dispositivos conectados al puerto serie y convertir los datos en formato paralelo, transmitidos al bus de sistema, a datos en formato serie, para que puedan ser transmitidos a través de los puertos y viceversa. («Transmisión Serial - USART - Sistemas Digitales» 2018)

## **Encoder**

Un codificador rotatorio, también llamado codificador del eje o generador de pulsos, suele ser un dispositivo electromecánico usado para convertir la posición angular de un eje a un código digital, lo que lo convierte en una clase de transductor. («Es.wikipedia.org» 2018)

# **MARCO METODOLÓGICO**

## **Hipótesis**

Es posible abstraer un sistema dinámico en un modelo de machine learning entrenado con datos temporales, que pueda ser usado por algoritmos genéticos para sintonizar un controlador PID de una planta de control.

## **Variables**

### **Variables independientes**

* Amplitud de señal de entrada.
* Duración de señal de entrada.
* Frecuencia de muestreo.
* Número de muestras.

### **Variables dependientes**

* Velocidad del motor.
* Ganancia proporcional.
* Ganancia integral.
* Ganancia derivativa.

***Tabla 2.1:*** *Variables*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable** | **Definición conceptual** | **Definición operacional** | **Indicador** | **Escala de medición** |
| Amplitud de señal de entrada. | Medida de la variación máxima del voltaje que varía periódicamente en el tiempo. | [Valor cuadrático medio de una señal PWM de excitación del sistema.](https://es.wikipedia.org/wiki/Media_cuadrática) | V | 0 – 12 |
| Duración de señal de entrada. | Tiempo que dura una señal de entrada. | Tiempo en que se mantiene la amplitud de una señal de entrada. | s | > 0 |
| Frecuencia de muestreo | [Número de muestras por unidad de tiempo que se toman de una señal continua para producir una señal discreta.](https://es.wikipedia.org/wiki/Señal_analógica) | Número de muestras por segundo de los valores de manipulación del duty cycle de la señal PWM y los valores de velocidad motor. | Hz | > 0 |
| Número de muestras | Cantidad de porciones extraídas de un conjunto por métodos que permiten considerarla como representativa de él. | Cantidad de datos obtenidos por el proceso de muestreo digital. | n° | > 0 |
| Velocidad del motor | Magnitud física que expresa el ángulo descrito en la unidad detiempo por el radio de un cuerpo que gira en torno de un eje. | Magnitud física que expresa la cantidad de radianes por segundo recorridos por el eje del motor. | Rad/s | > 0 |
| Ganancia proporcional | Valor del controlador que puede reducir, pero no eliminar, el error en estado estacionario. | Salida del controlador es proporcional a la señal de error. | nº | 0 – 1000 |
| Ganancia integral | Valor del controlador que elimina errores estacionarios. | Salida del controlador que es la integral de la señal de error. | nº | 0 – 1000 |
| Ganancia derivativa | Valor del controlador que anticipa el efecto de la acción proporcional para estabilizar la variable controlada después de cualquier perturbación. | Salida del controlador que es la derivada de la señal de error. | nº | 0 – 1000 |

## **Metodología**

### **Tipo de estudio**

# Exploratorio

### **Diseño**

# Experimental.

## **Población y muestra**

### **Población**

Motores como plantas de control de velocidad.

### **Muestra**

Motor de voltaje de operación nominal de 12VDC con encoder interno de voltaje de operación de 5VDC como planta de control de velocidad.

## **Método de investigación**

Cuantitativo.

## **Técnicas e instrumentos de recolección de datos**

### **Técnicas**

# Revisión de bibliografía especializada para explorar señales de excitación sobre diversos sistemas de control.

# Estudio del comportamiento del sistema de control frente a diversos estímulos.

### **Instrumentos**

# Jupyter Notebook

# Plataforma Arduino

### **Métodos de análisis de datos**

# Pre procesamiento de la señal de salida de la planta de control: aplicación de filtrado digital sobre los datos provenientes del sensor para su posterior procesamiento.

# **DESARROLLO Y RESULTADOS**

## **Implementación del sistema de control**

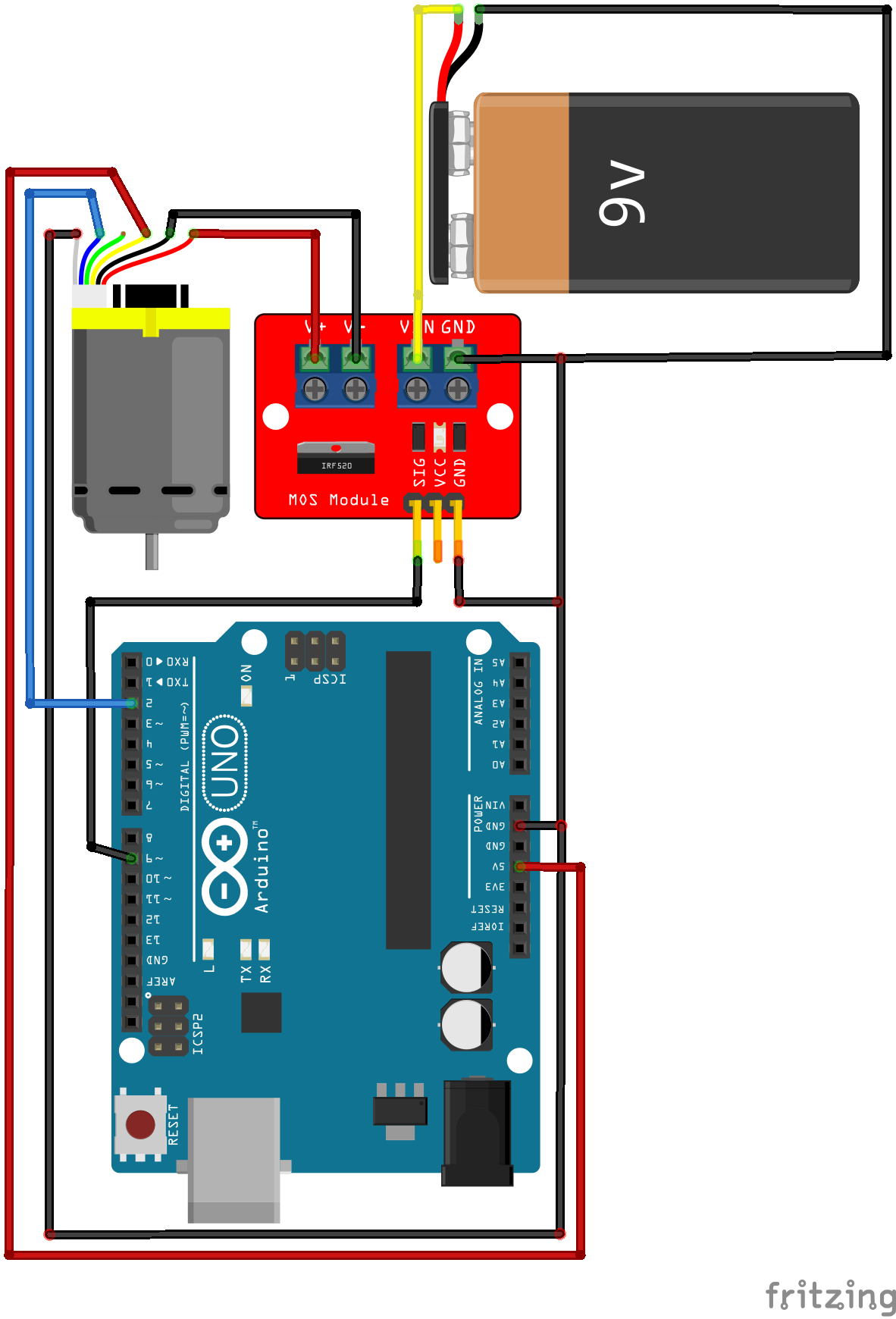
***Tabla 2.1:*** Elementos de software utilizados:

|  |  |
| --- | --- |
| **ELEMENTO** | **DESCRIPCIÓN** |
| Arduino IDE | Entorno Open Source de programación para la placa Arduino |
| Jupyter Notebook | Aplicación web Open Source de múltiples usos como: limpieza y transformación de datos, simulación numérica, modelado estadístico, visualización de datos y aprendizaje automático. |

***Tabla 2.2:*** Elementos de hardware utilizados:

|  |  |  |
| --- | --- | --- |
| **ELEMENTO** | **ESPECIFICACIONES** | **DESCRIPCIÓN** |
| Computadora | Procesador 4 núcleos mínimo | Elemento receptor de datos para su posterior procesamiento. |
| Placa Arduino Uno R3 | Microcontrolador: ATmega328P Voltaje de Operación: 5V Voltaje de alimentación: 6-20V Corriente máxima entrada/salida: 40mA | Tarjeta de adquisición de datos y elemento controlador |
| [Driver Mosfet IRF520](https://naylampmechatronics.com/drivers/239-driver-mosfet-irf520.html) | Voltaje de salida: 0-24 V DC Voltaje de control: 5V TTL Corriente máxima: 9 A Corriente nominal: 6 A | Amplifica la salida del controlador y actúa sobre la planta de control |
| Motor DC | Operación nominal: 6 a 12 VDC | Planta de control |
| Encoder interno | Voltaje de Operación: 5V  4 pulsos por vuelta | Elemento sensor |
| Fuente de voltaje | Voltajes de salida: 5 y 12 VDC | Alimentación del sistema de control |

***Figura 3.1:*** Disposición de los elementos de hardware del sistema de control. Elaborado en Fritzing.



**12 VDC**

**IRF520**

**Motor DC y encoder**

**Placa Arduino**

**Fuente de alimentación**

### **Etapa de adquisición y procesamiento de datos:**

En esta etapa, la placa Arduino funcionará como tarjeta de adquisición de datos de las señales de entrada y salida de la planta de control. La placa se comunicará por medio del puerto serial a la laptop. En la laptop se realizará el procesamiento de datos que resultará en la sintonización del controlador PID del sistema de control.

***Figura 3.2:*** *Esquema de la disposición de elementos de software y hardware.*



### **Etapa de control:**

En esta etapa, ya con el controlador PID sintonizado e implementado en la placa Arduino, esta funcionará como elemento controlador del sistema de control.

***Figura 3.3:*** *Esquema de la disposición de elementos de software y hardware.*



## **Elección de una señal de entrada y obtención de un conjunto adecuado de datos experimentales en tiempo discreto para la identificación de la planta de control por medio de un modelo de machine learning.**

El sistema estará sujeto a perturbaciones repentinas. Por lo tanto, una señal de entrada de tipo escalón será apropiada para obtener la mayor información de la planta de control.

Por otro lado, el motor admite un valor máximo de voltaje de 12 voltios, que es cuando alcanza su máxima velocidad. Para manipular la velocidad (rad/s) y simular una señal de entrada tipo escalón con amplitud variable, lo que se varió fue el valor cuadrático medio del voltaje del motor, que puede ser manipulado por una señal PWM; de esta forma, una señal PWM de 0% de duty cycle (Correspondiente al valor 0 programado en el IDE de Arduino para un pin PWM) permite obtener un valor de voltaje promedio de 0, y una señal PWM de 100% de duty cycle (Correspondiente al valor 255 programado en el IDE de Arduino para un pin PWM) permite obtener el valor de voltaje cuadrático medio de 12 V.

Es así que, mediante la realización de pruebas experimentales con señales de entrada de tipo escalón, con valores distintos de amplitud y duración (Número de ciclos de 32.64ms del timer del microcontrolador), fue posible determinar que el sistema se estabiliza frente a un cambio en la excitación, en menos de 4.896 s (150 ciclos del timer), y se observó que esto sucede sin importar la amplitud de la señal de entrada.

***Figura 3.4:*** *Ejemplo de una señal de excitación para el sistema (Color azul) y su respuesta (Color naranja) con valores de amplitud normalizados*



Con las consideraciones anteriores, se determinaron los parámetros de diseño de la señal de entrada:

* **Tipo:**

Escalón

* **Amplitud:**

0 – 12 V (Voltaje RMS gobernado por valores entre 0 y 255 programados en el IDE de Arduino)

* **Duración:**

4.896 s

***Figura 3.5:*** *Señal de entrada para el sistema (color azul) y su velocidad (color naranja) en una ventana de tiempo con valores de amplitud normalizados.*



Tanto para la etapa de adquisición de datos como para la etapa de control, se definió un tiempo de muestreo 0.22848 (7 ciclos del timer). Además, debido al ruido en la salida del sistema, fue necesario usar un filtro digital pasa bajo de primer orden con una frecuencia de corte y el tiempo de muestreo definido :

Se tomaron más de 17000 muestras que fueron guardados en la hoja de cálculo “motor. xlms”.

A continuación, se muestran los diagramas de flujo del proceso de muestreo digital:

***Figura 3.6:*** *Implementación del muestreo digital en el microcontrolador.*



***Figura 3.7:*** *Implementación de la interrupción por timer en el microcontrolador.*



***Figura 3.8:*** *Almacenamiento de datos en memoria y creación de la hoja de cálculo mediante un programa en Python usando Jupyter Notebook.*



## **Elección del modelo de machine learning y definición de sus restricciones para abstraer el sistema**

***Figura 3.9:*** *Vista de los datos guardados en la hoja de cálculo.*







En donde la columna “u” representa el vector de valores entre 0 y 255 que gobiernan el voltaje RMS que recibe el motor y la columna “y”, el vector de salidas del sistema (Rad/s).

Para la identificación de la planta de control por medio de un modelo de regresión a partir de los datos obtenidos en la hoja de cálculo, se determinaron los parámetros de los conjuntos de entrada y salida para entrenamiento y test del modelo de machine learning en base a cinco muestras (Correspondientes a una ventana de tiempo de 1.1424 s). Quedando:

**Parámetros del conjunto de entrada:**

* **X1:** Un valor de la columna “u”
* **X2:** Valor de la columna “u” anterior a X1
* **X3:** Valor de la columna “y” relacionado a X2
* **X4:** Valor de la columna “u” anterior a X2
* **X5:** Valor de la columna “y” relacionado a X4
* **X6:** Valor de la columna “u” anterior a X4
* **X7:** Valor de la columna “y” relacionado a X6
* **X8:** Valor de la columna “u” anterior a X6
* **X9:** Valor de la columna “y” relacionado a X8

**Parámetros del conjunto de salida:**

* **Y1:** Valor de la columna “y” relacionado a X1

***Tabla 2.3:*** *Parámetros del conjunto de entrada*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **X1** | **X2** | **X3** | **X4** | **X5** | **X6** | **X7** | **X8** | **X9** |
| u(k) | u(k-1) | y(k-1) | u(k-2) | y(k-2) | u(k-3) | y(k-3) | u(k-4) | y(k-4) |

***Tabla 2.4:*** *Parámetros del conjunto de salida*

|  |
| --- |
| **Y1** |
| y(k) |

***Figura 3.10:*** *Vista del conjunto de entrada generado para entrenar el modelo de machine learning.*







Como se puede observar, el conjunto original de datos se vio comprometido por los parámetros elegidos para el conjunto de entrada, quedando cuatro datos menos debido a la eliminación de datos indeterminados. Afectando también al conjunto de salida que será igual a los valores de la columna “y” pero eliminando los últimos cuatro valores.

El modelo de machine learning utilizado fue el modelo de regresión lineal “Elastic Net” de la biblioteca “Scikit Learn” de python.

***Figura 3.11:*** *Diagrama de flujo del proceso de generación de los conjuntos de entrada y salida.*



***Figura 3.12:*** *Diagrama de flujo del proceso de entrenamiento y test del modelo*.



Los valores máximo y mínimo de la columna “y” correspondiente a la velocidad (Rad/s) definirán las restricciones del modelo entrenado para evitar valores de salida ajenos a la planta de control real:

* **Velocidad máxima:** 395.369 rad/s
* **Velocidad mínima:** 0 rad/s

## **Implementación del controlador PID discreto para definir su evaluación de desempeño, en base a la integral del tiempo multiplicado por el** **valor absoluto del error y su ejecución sobre el modelo de machine learning**

Se implementó un bucle que ejecuta mil veces la acción de un controlador PID discreto sobre el modelo de machine learning. Puesto que el tiempo de muestreo elegido fue de 0.22848 s (7 ciclos del timer del microcontrolador), las mil iteraciones evalúan el desempeño del controlador durante , en base a la integral del tiempo multiplicado por el valor absoluto del error.

El tiempo de muestreo máximo recomendado para la implementación del controlador, según la referencia teórica, es la décima parte del tiempo requerido para la estabilización de la señal de salida del sistema: 4.896 s (150 ciclos del timer). Es decir, 0.4896 s. Por lo tanto, un tiempo de muestreo de 0.22848 es un tiempo adecuado.

El formato de la entrada para usar el modelo de machine learning es análoga a los parámetros del conjunto de entrada:

***Tabla 2.5:*** *Formato de una entrada.*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **u(k)** | **u(k-1)** | **y(k-1)** | **u(k-2)** | **y(k-2)** | **u(k-3)** | **y(k-3)** | **u(k-4)** | **y(k-4)** |
| pid(k) | pid (k-1) | vel(k-1) | pid (k-2) | vel (k-2) | pid (k-3) | vel (k-3) | pid (k-4) | vel (k-4) |

El formato de la salida generada por el modelo de machine learning es análoga a los parámetros del conjunto de salida:

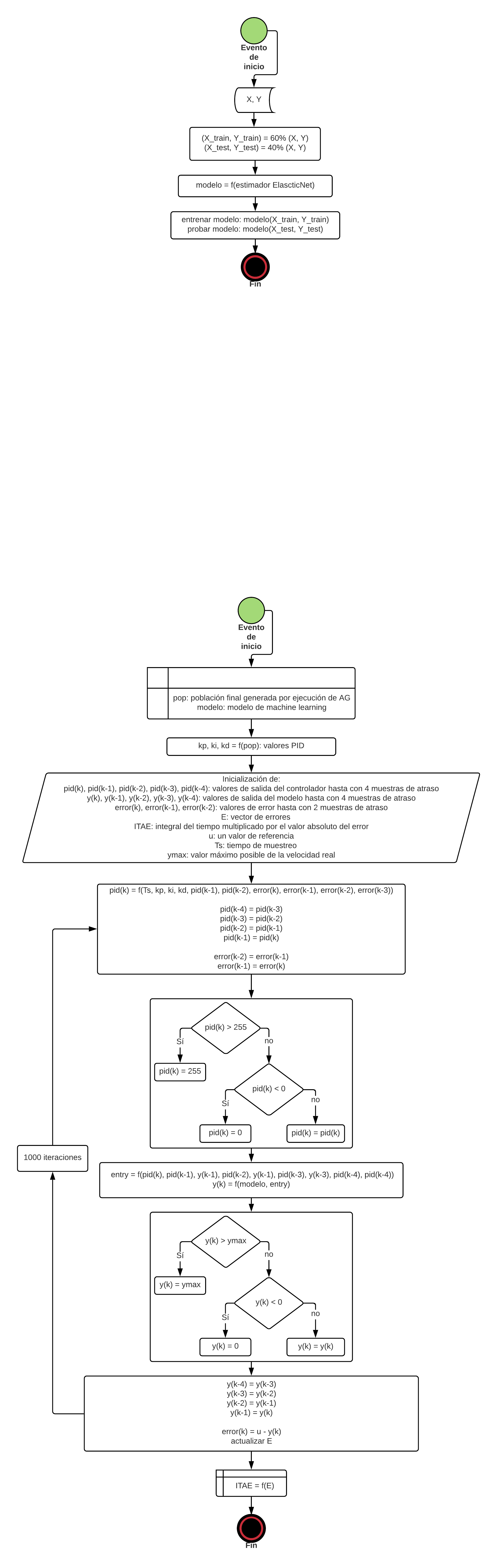
***Tabla 2.6:*** *Formato de una salida.*

|  |
| --- |
| **y(k)** |
| vel(k) |

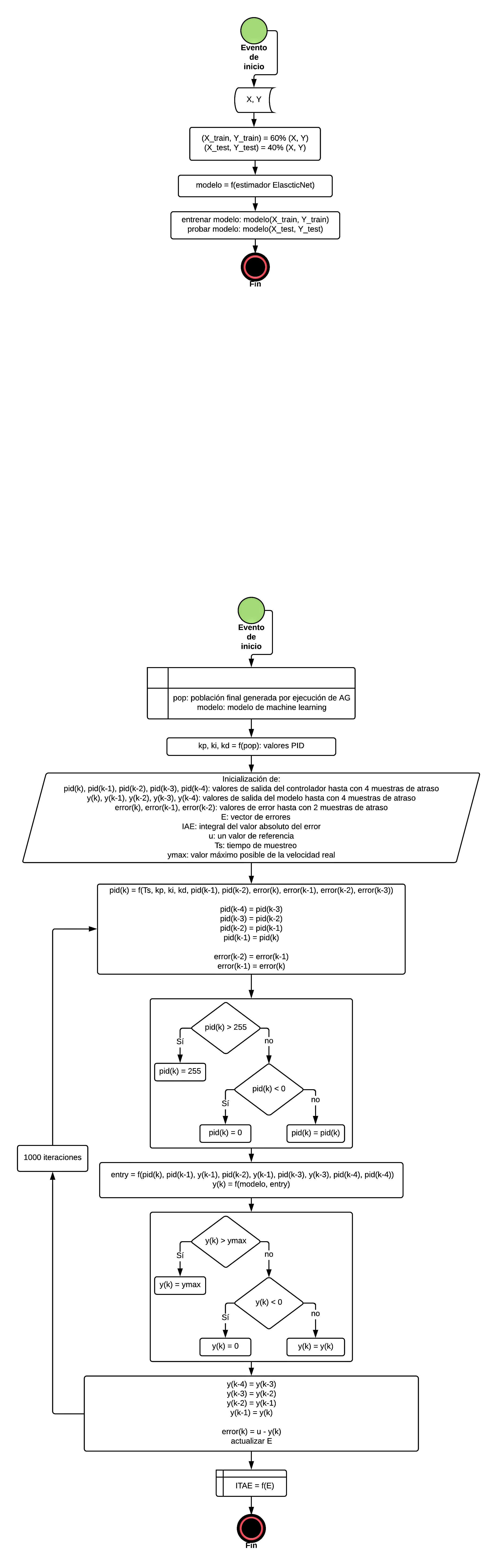
Donde pid se refiere al valor de salida del controlador y vel, al valor de la velocidad.

A continuación, se muestra el diagrama de flujo de la función para evaluar el desempeño del controlador:

***Figura 3.13:*** *Diagrama de flujo de la evaluación de la performance del controlador.*



***Figura 3.14:*** *Diagrama de flujo de la evaluación de la performance del controlador. Continuación*.



## **Sintonización del controlador PID por medio de un algoritmo genético y prueba en la planta real**

Se definieron los parámetros propios del algoritmo genético: recombinación en 0.7, mutación en 0.1, tamaño de la población en 40 individuos y número de generaciones en 40.

Además, los cromosomas (individuos) usados por el algoritmo genético serán cadenas binarias, las cuales serán decodificadas por la función de evaluación de desempeño del controlador para obtener los valores de las ganancias proporcional, integral y derivativa. Para esto, se debe definir la resolución digital de la cadena binaria de los individuos. Además, también definir los valores decimales máximos y mínimos posibles que podrían adoptar. La cantidad de bits de cada ganancia (Y, en consecuencia, la resolución digital de cada individuo) depende de la precisión que se busca o necesita. De forma general, para una ganancia si se elige una precisión de bits, el algoritmo genético podrá buscar entre valores de un rango de a definido previamente.

Para este trabajo, la resolución digital escogida fue de 30 bits, con 10 bits para cada ganancia y todos los valores máximos fueron definidos en 10 y los mínimos, en 0. Por ejemplo, se muestra el mejor individuo obtenido luego de ejecutar el algoritmo genético:

AQUÍ UNA IMAGEN DEL SERIAL PLOTTER DEL POCESO DE CONTROL

***Figura 3.15:*** *Diagrama de flujo del proceso de sintonización del controlador mediante algoritmos genéticos.*



***Figura 3.16:*** *Diagrama de flujo del controlador PID digital en la placa Arduino.*



***Figura 3.17:*** *Diagrama de flujo del controlador PID digital en la placa Arduino (Interrupciones).*



# **DISCUCIÓN**

## **El sistema de control implementado**

* El encoder de la planta de control envía 4 pulsos por vuelta. Aunque dicha cantidad resultó suficiente para implementar el sistema de control del presente trabajo, mientras más pulsos sean enviados por vuelta, mejor será la medición en sistemas de control de velocidad, posición, etc.
* Debido al microcontrolador de la placa Arduino utilizada, los tiempos de interrupción que se pueden implementar en el reloj (timer) ya están definidos. Por lo tanto, los posibles tiempos de muestreo, ya sea para fines de adquisición de datos o control, solo podrán ser múltiplos del tiempo de interrupción seleccionado. En este trabajo, el tiempo de interrupción seleccionado para el reloj del microcontrolador es de 32.64 ms, por ejemplo.

## **La señal de entrada para el sistema y el conjunto de datos experimentales en tiempo discreto**

* El sistema de control estudiado no estuvo sujeto a entradas que cambien de forma gradual en un determinado tiempo, ni tampoco a entradas de choque; solo estuvo sometido a cambios repentinos. De otra manera, la señal de entrada no sólo debería ser de tipo escalón, puesto que no se obtendría un conjunto de datos que refleje de forma exitosa el comportamiento del sistema.
* Ya que los datos obtenidos en el proceso de muestreo digital, corresponden a un tiempo de muestreo determinado, el modelo de machine learning que se entrenó con los datos mencionados, se podría entender prácticamente como un modelo de la planta de control en el dominio discreto que corresponderá a un igual tiempo de muestreo.

## **El modelo de machine learning**

* Los parámetros para conformar el conjunto de entradas fueron seleccionados partiendo de la consideración de que las plantas de control de velocidad de motores DC, como el utilizado en este trabajo, son a menudo de segundo o tercer orden en el dominio discreto al aplicar métodos tradicionales de identificación de sistemas. Esto permite deducir que un modelamiento matemático del sistema depende de estados actuales y atrasados, en este caso, posiblemente hasta en tres muestras. Para el modelo de machine learning obtenido, se halló una mejor performance usando desde estados actuales hasta atrasados en cuatro muestras. Esto se observa en la comprobación del funcionamiento del modelo con el conjunto de prueba, el cual arrojó más de un 96 % de éxito en la identificación de la planta de control.
* El tipo de modelo de machine learning fue elegido debido a que los parámetros del conjunto de entradas están correlacionados unos con otros; una característica claramente presente en el funcionamiento de un sistema dinámico.

## **Implementación del controlador PID discreto y definición de su evaluación de desempeño en base a la integral del tiempo multiplicado por el valor absoluto del error y su ejecución sobre el modelo de machine learning**

* La ejecución del controlador PID discreto se implementó correspondiendo al tiempo de muestreo de los datos con el que fue entrenado el modelo de machine learning. Este tiempo de muestreo tiene restricciones adicionales que debe cumplir, definidas en la referencia teórica.
* El resultado de la evaluación de desempeño del controlador dado por la integral del tiempo multiplicado por el valor absoluto del su error, es un criterio puramente cuantitativo con el cual no se puede caracterizar al funcionamiento del controlador en base a parámetros cualitativos muy ligados al proceso o a los elementos físicos del sistema de control.

## **Sintonización del controlador PID por medio de un algoritmo genético**

* Las probabilidades de mutación y recombinación se definieron para que el algoritmo genético permita que la gran mayoría de los individuos de una generación sea una composición mejorada de la generación anterior; pero dejando abierta la posibilidad de obtener individuos mejorados radicalmente debido a variaciones aleatorias en sus características. Los valores escogidos para el tamaño de la población y el número de generaciones permiten al algoritmo genético recorrer por todo el rango definido sin problemas.
* La cantidad de bits de cada ganancia (Y, en consecuencia, la resolución digital) se eligió para tener individuos con un amplio número de elementos de búsqueda para el algoritmo genético dentro del rango dado.

# **CONCLUSIONES**

# El envío de 4 pulsos por vuelta desde el encoder interno de la planta hacia la tarjeta de adquisición de datos programada con el tiempo de muestreo de 0.22848s y un filtro digital pasa bajos de frecuencia de corte 0.1 Hz fueron suficientes para lograr implementar el sistema de control de velocidad.

# La elección de una señal tipo escalón, con una duración de 4.896s antes del siguiente cambio en su amplitud a algún valor entre 0 y 255 para manipular la señal PWM de excitación de la planta, permitió obtener un conjunto de datos que realmente reflejó en gran magnitud el comportamiento real del sistema.

# El modelo de machine learning basado en el método de regresión lineal Elastic Net logró identificar la planta en más del 96% según la comprobación de los datos del conjunto de prueba.

# Bajo el supuesto de que el modelo de machine learning obtenido, era prácticamente una abstracción que reflejaba el comportamiento de la planta en tiempo discreto correspondiendo a un tiempo de muestreo de 0.22848, y teniendo en cuenta sus restricciones de velocidad máxima de 395.369 rad/s y mínima de 0 rad/s; se logró implementar con éxito un controlador PID discreto con el mismo tiempo de muestreo para evaluar su desempeño en base al criterio cuantitativo que proporciona la integral del tiempo multiplicado por el valor absoluto del error.

# Los parámetros escogidos para el algoritmo genético permitieron hallar con éxito una tendencia rápida hacia la solución. Además, al escoger una cantidad de 10 bits para cada ganancia y un rango de valores entre 0 y 10, se proporcionó 1024 valores de búsqueda para el algoritmo genético, lo que significa una precisión de 0.0097.

# **SUGERENCIAS**

# La planta de control utilizada en este trabajo cumple con características idóneas para implementar un sistema de control de velocidad, pero si se busca implementar un sistema de control de posición angular entonces se debe buscar un sistema motor dc con encoder interno que envíe muchos más pulsos por vuelta o acoplar un encoder externo y su disco respectivo, según la precisión angular que se busca.

# Es fundamental elegir una señal o las señales de entrada adecuadas para garantizar el éxito de la identificación del sistema dinámico estudiado a través de un modelo de machine learning. Esto se logra tomando en cuenta todas las perturbaciones a las que podría estar sometida la planta de control y estudiando su comportamiento frente a ellas; sin importar si la planta de control es un sistema lineal simple de una entrada y una salida o es un sistema complejo no lineal de múltiples entradas y salidas.

# Los parámetros seleccionados para conformar el conjunto de entradas para entrenar el modelo de machine learning dependen directamente del tiempo de establecimiento de la planta y la cantidad de variables de entrada del sistema de control. Es así que, si un sistema tiene un tiempo de establecimiento relativamente prolongado, será necesario aumentar la cantidad de estados anteriores (número de atrasos en las muestras) para cada una de las variables de entradas y salidas consideradas en la selección de parámetros.

# No se usaron métodos analíticos para seleccionar los rangos de búsqueda de los parámetros PID. A pesar de que los métodos analíticos pueden ser muy eficientes en sistemas simples, mientras la complejidad de un sistema va aumentando, su análisis también. Es aquí donde se muestra la ventaja principal de incorporar técnicas de machine learning y algoritmos genéticos en problemas de control, puesto que se pueden implementar pruebas para estimar la performance del controlador y dar así con valores adecuados para sintonizarlo con éxito. Por lo que el usuario podría establecer categorías de búsqueda para las ganancias, por ejemplo, con 4 categorías: [0:1], [1:10], [10, 100] y [100, 1000] se tendrían 4 posibles rangos de búsqueda para cada ganancia Kp. Ki y Kd; lo que daría lugar a 4 x 4 x 4 = 64 pruebas de las que se obtendría los mejores valores de sintonización entre 0 y 1000. Y para aprovechar mejor las capacidades computacionales de la tecnología actual, se podría evaluar cada controlador en menos iteraciones e iniciar realizando un control proporcional (El cual necesitaría solo 4 pruebas) y continuar con el resto de ganancias. Y esto sería la base para automatizar el proceso de sintonización siempre y cuando los valores que generen un controlador adecuado, estén entre 0 y 1000.

# **BIBLIOGRAFÍA**

KOCUR, M., KOZAK, S. and DVORSCAK, B. 2014. Design and Implementation of FPGA - digital based PID controller. Proceedings of the 2014 15th International Carpathian Control Conference (ICCC). pp. 1. DOI 10.1109/carpathiancc.2014.6843603. IEEE

VILANOVA, R. and VISIOLI, A. 2012. PID Control in the Third Millennium. 1. Dordrecht: Springer. ISBN 9781447124252, 1447124251.

ÅSTRÖM, K. and HÄGGLUND, T. 2001. The future of PID control. Control Engineering Practice. Vol. 9, no. 11, pp. 1163, 1173. DOI 10.1016/s0967-0661(01)00062-4. Elsevier BV

YIN, S., LI, X., GAO, H. and KAYNAK, O. 2015. Data-Based Techniques Focused on Modern Industry: An Overview. IEEE Transactions on Industrial Electronics. Vol. 62, no. 1, pp. 657. DOI 10.1109/tie.2014.2308133. Institute of Electrical and Electronics Engineers (IEEE)

FUNES, E., ALLOUCHE, Y., BELTRÁN, G. and JIMÉNEZ, A. 2015. A Review: Artificial Neural Networks as Tool for Control Food Industry Process. Journal of Sensor Technology. Vol. 05, no. 01, pp. 34. DOI 10.4236/jst.2015.51004. Scientific Research Publishing, Inc.

MARIN CANO, A., HERNANDEZ RIVERO, J. and JIMENEZ BUILES, J. 2018. Tuning Multivariable Optimal PID Controller for a Continuous Stirred Tank Reactor Using an Evolutionary Algorithm. IEEE Latin America Transactions. Vol. 16, no. 2, pp. 422. DOI 10.1109/tla.2018.8327395. Institute of Electrical and Electronics Engineers (IEEE)

LÓPEZ MORA, C. 2014. Evaluación de desempeño de dos técnicas de optimización bio-inspiradas: Algoritmos Genéticos y Enjambre de Partículas. Revista Tekhnê [en línea]. Vol. 11, no. 1, pp. 56. [Consulta: 1 septiembre 2018]. ISSN 1692-8407. Disponible en: http://revistas.udistrital.edu.co/ojs/index.php/tekhne/article/view/8951. Tekhnê

GARCIA PEREDA, J. 2018. Empleo del algoritmo de colonia de hormigas para la sintonización de un controlador pid aplicado a un circuito rc ﻿. [en línea]. Vol. 1, no. 1, pp. 97. [Consulta: 1 septiembre 2018]. Disponible en: http://dspace.unitru.edu.pe/handle/UNITRU/898. Repositorio Institucional de la Universidad Nacional de Trujillo

BALCÁZAR LLANOS, J. 2017. Uso De Algoritmos Genéticos Para La Sintonización De Un Controlador Pid Aplicado Al Control De Nivel Utilizando El Entrenador De Control De Procesos “Dl 2314” Del Instituto Superior Tecnológico Publico Nueva Esperanza. [en línea]. Vol. 1, no. 1, pp. 96. [Consulta: 1 septiembre 2018]. Disponible en: http://dspace.unitru.edu.pe/handle/UNITRU/9482. Repositorio Institucional de la Universidad Nacional de Trujillo

ÅSTRÖM, K. and MURRAY, R. 2006. Feedback Systems: An Introduction for Scientists and Engineers [en línea]. 2. S.l.: Karl J. Åström and Richard M. Murray. [Consulta: 1 septiembre 2018]. Disponible en: http://www.cds.caltech.edu/~murray/books/AM08/pdf/am06-preface\_16Sep06.pdf.

BURGOS MOLINA, C. and CHACÓN PACHECO, M. 2014. ENTRENAMIENTO MUESTRAL DE MODELOS DINÁMICOS CON SVM. [en línea]. pp. 7. [Consulta: 1 septiembre 2018]. Disponible en: http://www.jcc2014.ucm.cl/jornadas/EVENTOS/ECC%202014/ECC-2.pdf. Jornadas Chilenas de Computación 2014

SCHÖLKOPF, B. and SMOLA, A. 2001. Learning with kernels. 1. Cambridge, Mass.: MIT Press.

BAUTISTA THOMPSON, E., GUZMÁN RAMÍREZ, E. and FIGUEROA NAZUNO, J. 2004. Predicción de Múltiples Puntos de Series de Tiempo Utilizando Support Vector Machines. Computación y Sistemas [en línea]. Vol. 7, no. 3, pp. 154. [Consulta: 1 septiembre 2018]. DOI ISSN 1405-5546. Disponible en: http://www.scielo.org.mx/scielo.php?script=sci\_abstract&pid=S1405-55462004000100002&lng=es&nrm=iso. Scielo

GRETTON, A., DOUCET, A., HERBRICH, R., RAYNER, P. and SCHOLKOPF, B. 2002. Support vector regression for black-box system identification. Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing (Cat. No.01TH8563). pp. 341, 344. DOI 10.1109/ssp.2001.955292. IEEE

DREZET, P. and HARRISON, R. 1998. Support vector machines for system identification. UKACC International Conference on Control (CONTROL '98). pp. 692. DOI 10.1049/cp:19980312. IEE

OGATA, K. 1995. Discrete time control systems. 2. London, NJ: Prentice-Hall International.

OGATA, K. 2010. Modern control engineering. 5. Upper Saddle River, NJ: Prentice-Hall/Pearson.

KUO, B. and ARANDA PÉREZ, G. 1996. Sistemas de control automático. 7. México [etc.]: Prentice Hall Hispanoamericana.

Bhattacharya, S. 2009. Control systems engineering. 1. Delhi: Dorling Kindersley.

SARKAR, DIPANJAN, BALI, RAGHAV and SHARMA, TUSHAR, 2018, Practical machine learning with Python. 1. Bangalore, Karnataka, India: Raghav Bali.

RAMKRISHNA, GHOSH, 2016, Input designs for identification of Ill-conditioned multivariable systems. 1. Finland: Åbo Akademi University.

ACEDO SÁNCHEZ, J. 2006. Instrumentación y control avanzado de procesos. 1. Madrid: Díaz de Santos.

GRETTON, A., DOUCET, A., HERBRICH, R., RAYNER, P. and SCHOLKOPF, B. [sin fecha]. Support vector regression for black-box system identification. Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing (Cat. No.01TH8563) [en línea]. pp. 1. [Consulta: 17 septiembre 2018]. DOI 10.1109/ssp.2001.955292. Disponible en: https://ieeexplore.ieee.org/document/955292/. IEEE

YU WANG 2017. A new concept using LSTM Neural Networks for dynamic system identification. 2017 American Control Conference (ACC). pp. 1-2. DOI 10.23919/acc.2017.7963782. IEEE

S. SAAD, M., JAMALUDDIN, H. and Z. M. DARUS, I. 2012. PID Controller Tuning Using Evolutionary Algorithms. [en línea]. Vol. 7, no. 4, pp. 1. [Consulta: 17 septiembre 2018]. Disponible en: https://[www.researchgate.net/](http://www.researchgate.net/)publication/279901140\_PID\_-controller\_tuning\_using\_evolutionary\_algorithms. WSEAS Transactions on Systems and Control.

MALLAWAARACHCHI, V. 2018. Introduction to Genetic Algorithms — Including Example Code. Towards Data Science [en línea]. [Consulta: 17 septiembre 2018]. Disponible en: https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3.

INGARGIOLA, A. y colaboradores 2018. 1. What is the Jupyter Notebook? — Jupyter/IPython Notebook Quick Start Guide 0.1 documentation. Jupyter-notebook-beginner-guide.readthedocs.io [en línea]. [Consulta: 17 septiembre 2018]. Disponible en: https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what\_is\_jupyter.html.

Comunicación Serial: Conceptos Generales - National Instruments. Digital.ni.com [en línea] 2018.

Transmisión Serial - USART - Sistemas Digitales. Sistdig.wikidot.com [en línea] 2018.

An introduction to machine learning with scikit-learn — scikit-learn 0.19.2 documentation. Scikit-learn.org [en línea] 2018.

Lasso and Elastic Net- MATLAB & Simulink- MathWorks America Latina. La.mathworks.com [en línea] 2018.

Project Jupyter. Jupyter.org [en línea] 2018.

Colaboradores de Wikipedia. Arduino [en línea]. Wikipedia, La enciclopedia libre, 2018 [fecha de consulta: 17 de septiembre del 2018]. Disponible en: https://es.wikipedia.org/w/index.php?title=Arduino&oldid=110567587.

Colaboradores de Wikipedia. Arduino [en línea]. Wikipedia, La enciclopedia libre, 2018 [fecha de consulta: 1 de junio del 2018]. Disponible en: https://es.wikipedia.org/w/index.php?title=Arduino&oldid=110567587.

Pardo, C. (2018). PID digital - Picuino. [online] Sites.google.com. Disponible en: https://sites.google.com/site/picuino/digital\_pid#TOC-Per-odo-de-muestreo [fecha de consulta: 1 de junio del 2018].

Colaboradores de Wikipedia. Modulación por ancho de pulsos [en línea]. Wikipedia, La enciclopedia libre, 2018 [fecha de consulta: 1 de junio del 2018]. Disponible en: https://es.wikipedia.org/w/index.php?title=Modulaci%C3%B3n\_por\_ancho\_de\_pulsos&oldid=110019730.

Colaboradores de Wikipedia. Valor eficaz [en línea]. Wikipedia, La enciclopedia libre, 2018 [fecha de consulta: 1 de junio del 2018]. Disponible en: https://es.wikipedia.org/w/index.php?title=Valor\_eficaz&oldid=110312485.

Colaboradores de Wikipedia. Codificador rotatorio [en línea]. Wikipedia, La enciclopedia libre, 2018 [fecha de consulta: 1 de junio del 2018]. Disponible en: https://es.wikipedia.org/w/index.php?title=Codificador\_rotatorio&oldid=105727872.

Colaboradores de Wikipedia. Adquisición de datos [en línea]. Wikipedia, La enciclopedia libre, 2018 [fecha de consulta: 1 de junio del 2018]. Disponible en: https://es.wikipedia.org/w/index.php?title=Adquisici%C3%B3n\_de\_datos&oldid=110268301.

Colaboradores de Wikipedia. Adquisición de datos [en línea]. Wikipedia, La enciclopedia libre, 2018 [fecha de consulta: 1 de junio del 2018]. Disponible en: https://en.wikipedia.org/wiki/Settling\_time.

Colaboradores de Wikipedia. Adquisición de datos [en línea]. Wikipedia, La enciclopedia libre, 2018 [fecha de consulta: 1 de junio del 2018]. Disponible en: https://es.wikipedia.org/wiki/Velocidad\_angular.

Colaboradores de Wikipedia. Adquisición de datos [en línea]. Wikipedia, La enciclopedia libre, 2018 [fecha de consulta: 1 de junio del 2018]. Disponible en: https://es.wikipedia.org/wiki/Software.

Colaboradores de Wikipedia. Adquisición de datos [en línea]. Wikipedia, La enciclopedia libre, 2018 [fecha de consulta: 1 de junio del 2018]. Disponible en: https://es.wikipedia.org/wiki/Resoluci%C3%B3n\_digital.

Colaboradores de Wikipedia. Adquisición de datos [en línea]. Wikipedia, La enciclopedia libre, 2018 [fecha de consulta: 1 de junio del 2018]. Disponible en: https://es.wikipedia.org/wiki/Arduino.

Colaboradores de Wikipedia. Adquisición de datos [en línea]. Wikipedia, La enciclopedia libre, 2018 [fecha de consulta: 1 de junio del 2018]. Disponible en: https://es.wikipedia.org/wiki/Bit.

Colaboradores de Wikipedia. Adquisición de datos [en línea]. Wikipedia, La enciclopedia libre, 2018 [fecha de consulta: 1 de junio del 2018]. Disponible en: https://es.wikipedia.org/wiki/Hardware.

# **ANEXOS**