



UNIVERSIDAD SURCOLOMBIANA

GESTIÓN DE BIBLIOTECAS

CARTA DE AUTORIZACIÓN



CÓDIGO

AP-BIB-FO-06

VERSIÓN

1

VIGENCIA

2014

PÁGINA

1 de 2

Neiva, 29 de julio de 2022

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Neiva

El (Los) suscrito(s):

Oscar Oriol Toledo, con C.C. No. 7730613

Dairo Lizcano Sánchez, con C.C. No. 1076984489

Autor(es) de la tesis y/o trabajo de grado titulado “Diseño e implementación de un emulador de comportamiento dinámico para plantas lineales”, presentado y aprobado en el año 2022 como requisito para optar al título de Ingeniero Electrónico;

Autorizo (amos) al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que, con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

- Los usuarios puedan consultar el contenido de este trabajo de grado en los sitios web que administra la Universidad, en bases de datos, repositorio digital, catálogos y en otros sitios web, redes y sistemas de información nacionales e internacionales “open access” y en las redes de información con las cuales tenga convenio la Institución.
- Permite la consulta, la reproducción y préstamo a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato Cd-Rom o digital desde internet, intranet, etc., y en general para cualquier formato conocido o por conocer, dentro de los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia.
- Continúo conservando los correspondientes derechos sin modificación o restricción alguna; puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación del derecho de autor y sus conexos.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “Los derechos morales sobre el trabajo son propiedad de los autores”, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



UNIVERSIDAD SURCOLOMBIANA
GESTIÓN DE BIBLIOTECAS

CARTA DE AUTORIZACIÓN



CÓDIGO	AP-BIB-FO-06	VERSIÓN	1	VIGENCIA	2014	PÁGINA	2 de 2
--------	--------------	---------	---	----------	------	--------	--------

EL AUTOR/ESTUDIANTE:

Firma: Oscar Oscar Toledo
CC. 7730.613

EL AUTOR/ESTUDIANTE:

Firma: Dany
CC. 1070.984.489.

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO

CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	1 de 3
--------	--------------	---------	---	----------	------	--------	--------

TÍTULO COMPLETO DEL TRABAJO: Diseño e implementación de un emulador de comportamiento dinámico para plantas lineales.

AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
Toledo Laiseca	Oscar Oriol
Lizcano Sánchez	Dairo

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre
Sendoya	Diego Fernando

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre

PARA OPTAR AL TÍTULO DE: Ingeniero Electrónico

FACULTAD: Ingeniería

PROGRAMA O POSGRADO: Ingeniería Electrónica

CIUDAD: Neiva **AÑO DE PRESENTACIÓN:** 2022 **NÚMERO DE PÁGINAS:** 53

TIPO DE ILUSTRACIONES (Marcar con una X):

Diagramas Fotografías Grabaciones en discos Ilustraciones en general Grabados
Láminas Litografías Mapas Música impresa Planos Retratos Sin ilustraciones Tablas o Cuadros

SOFTWARE requerido y/o especializado para la lectura del documento:

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO

CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	2 de 3
--------	--------------	---------	---	----------	------	--------	--------

MATERIAL ANEXO: CD CON LA TESIS

PREMIO O DISTINCIÓN (En caso de ser LAUREADAS o Meritoria):

PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

Español:	Inglés
App Designer	App Designer
DAC	DAC
Emulador Función de trasferencia	Emulator Transfer Function
IDE Arduino	IDE Arduino
Microcontrolador	Microcontroller
Plantas Dinámicas	Dynamyc Plants
Resolución	Resolution

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)

Con el presente trabajo de grado se pretende exponer el estudio realizado para el diseño e implementación de un emulador de plantas dinámicas. Este estudio tiene como finalidad mejorar el entorno y las prácticas educativas para un óptimo aprendizaje en el área de control de procesos de los estudiantes de Ingeniería Electrónica de la Universidad Surcolombiana. El software se diseñó en el entorno de desarrollo integrado de Arduino y la interfaz gráfica de usuario en Matlab. En el App Designer de Matlab se desarrolló una aplicación independiente que presenta la interfaz de comunicación donde el usuario digita los coeficientes de la función de transferencia a emular, luego los datos recibidos se envían al Arduino donde se hace la respectiva discretización. El hardware consta de tres etapas, la primera mejora la resolución del dispositivo y se construyó por medio de arreglos de amplificadores operacionales, la segunda consta del microcontrolador Arduino Uno, donde se encuentra la mayor parte de la programación del dispositivo, la tercera fase consistió en la conversión de la señal digital a analógica implementada con el dispositivo DAC MCP4725. Esto en conjunto permitió la realización de un práctico y funcional dispositivo que proporcionará a los estudiantes la oportunidad de emular una mayor cantidad de procesos.

ABSTRACT: (Máximo 250 palabras)

With this degree work it is intended to expose the study carried out for the design and implementation of a dynamic plant emulator, this study aims to improve the environment and educational practices for optimal learning in the area of process control of students of Electronic Engineering of the Surcolombiana University.

The software was designed in the Arduino integrated development environment and the Designer App. In the Designer App the communication interface was built where the user inscribes the transfer function to emulate, then the received data is sent to the Arduino IDE where the respective discretization is done. The hardware consists of three stages, the first stage, improves the resolution of the device and was built by means of arrangements of operational amplifiers, the second consists of the Arduino One microcontroller, where most of the programming of the device is located, the third phase It consists of the conversion of the digital signal to analog implemented with the MCP4725 DAC device and all this together allowed the realization of a practical and functional device that will provide students with the opportunity to control a greater number of processes.

APROBACION DE LA TESIS

Nombre Presidente Jurado:

Firma:

Nombre Jurado: Faiber Robayo

Firma: 

Nombre Jurado: Fernand Díaz

Firma:

DISEÑO E IMPLEMENTACIÓN DE UN EMULADOR DE COMPORTAMIENTO
DINÁMICO PARA PLANTAS LINEALES

DAIRO LIZCANO SÁNCHEZ
OSCAR ORIOL TOLEDO LAISECA

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA (HUILA)
2022

**DISEÑO E IMPLEMENTACIÓN DE UN EMULADOR DE COMPORTAMIENTO
DINÁMICO PARA PLANTAS LINEALES**

Autores:
**DAIRO LIZCANO SÁNCHEZ
OSCAR ORIOL TOLEDO LAISECA**

Proyecto de Grado para optar al título de **INGENIERO
ELECTRÓNICO**

Director:
DIEGO FERNANDO SENDOYA
Ingeniero Electrónico
Especialista en Automatización Industrial y Magister en Ingeniería de
Control Industrial

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA (HUILA)
2022

Nota de Aceptación

Director

Evaluador 1

Evaluador 2

DEDICATORIA

Este trabajo está dedicado a mis padres que han sido el eje fundamental para mi formación integral y a mis hijos que son mi motor y motivación.

Dairo Lizcano Sánchez

Este trabajo está dedicado a mis padres, quienes desde siempre me han brindado su apoyo incondicional, especialmente en los momentos de dificultad, que es cuando su voz de aliento se ha hecho más fuerte. A mis hijos que han sido mi motivación más grande en cada meta propuesta.

Oscar Oriol Toledo

AGRADECIMIENTOS

Agradecemos a cada una de las personas que apoyaron este arduo proceso. A nuestros padres, por brindarnos apoyo económico y moral. A nuestro director de tesis, el Ingeniero Diego Fernando Sendoya, que ha estado presente en cada etapa de este trabajo, brindando los conocimientos necesarios; y, en general a cada uno de los profesores del programa de Ingeniería Electrónica que también han contribuido en nuestra formación como ingenieros; así como la Universidad Surcolombiana, que nos acogió y formó como profesionales.

TABLA DE CONTENIDO

	Pág.
DEDICATORIA	4
AGRADECIMIENTOS.....	5
LISTA DE FIGURAS	8
LISTA DE TABLAS	9
LISTA DE ANEXOS	10
RESUMEN.....	11
ABSTRACT.....	12
INTRODUCCIÓN.....	13
1. JUSTIFICACIÓN	14
2. OBJETIVOS	15
2.1. OBJETIVO GENERAL.....	15
2.2. OBJETIVOS ESPECÍFICOS.....	15
3. PLANTEAMIENTO DEL PROBLEMA	16
4. MARCO TEÓRICO.....	17
4.1 ANTECEDENTES DE LA INVESTIGACIÓN	17
4.2 ELEMENTOS TEÓRICOS UTILIZADOS PARA EL SOFTWARE.....	18
4.3 ELEMENTOS TEÓRICOS UTILIZADOS PARA EL HARDWARE.....	20
5. METODOLOGÍA.....	23
5.1 ADECUACIÓN DE CIRCUITOS A LA ENTRADA Y LA SALIDA PARA MEJOR RESOLUCIÓN DEL DISPOSITIVO	23
5.1.1 Adecuación de Circuito de Entrada.....	24
5.1.2 Adecuación de Circuito a la Salida.....	24
5.2 PROGRAMACIÓN EN MATLAB.....	25
5.2.1 Función Menu.....	25
5.2.2 Función Disc.....	26
5.2.3 Función Enviar.....	26
5.3 INTERFAZ GRÁFICA	27
5.4 RESTRICCIONES DE USO PARA EL DISPOSITIVO.....	28
5.5 COMUNICACIÓN SERIAL ARDUINO-MATLAB.....	29
5.6 PROCEDIMIENTO PARA DISCRETIZAR LA FUNCIÓN DE TRANSFERENCIA	29
5.7 PROCESO DE PROGRAMACIÓN EN ARDUINO.....	31
5.7.1 Comunicación.....	32
5.7.2 Diagrama de Flujo.:	32
5.8 PLANEACIÓN DE LA PRÁCTICA DE LABORATORIO	36
5.8.1 Titulo de la Práctica	36
5.8.2 Introducción.....	36
5.8.3 Objetivo de la Práctica.....	36
5.8.3.1 Objetivo General..	36
5.8.3.2 Objetivos Específicos.....	36
5.8.4 Metodología.....	36

5.8.5 Recursos, Materiales y Equipos	37
5.8.6 Desarrollo de la Práctica..	37
6. RESULTADOS	39
6.1 PLANTA DE PRIMER ORDEN.....	39
6.2 PLANTAS DE SEGUNDO ORDEN	40
6.2.1 Motor DC sobre plataforma ARM.	40
6.2.2 Modelo Dinámico de un Servomotor DC.	41
6.3 PLANTAS DE TERCER ORDEN.....	42
6.4 PLANTAS DE CUARTO ORDEN	43
7. ANÁLISIS DE RESULTADOS.....	45
BIBLIOGRAFIA.....	47
ANEXOS.....	49

LISTA DE FIGURAS

Figura 4- 1 Distribución de pines en Arduino Uno.....	21
Figura 4- 2 Conexión con Pines MCP4725 con Arduino.....	22
Figura 5- 1 Secuencia para la elaboración del dispositivo.....	23
Figura 5- 2 Arreglo de amplificadores para la entrada del emulador	24
Figura 5- 3 Arreglo de amplificadores para la salida del emulador.	24
Figura 5- 4 Diagrama de flujo para la función menu.	25
Figura 5- 5 Interfaz de Usuario del Emulador.	27
Figura 5- 6 Cuadro de dialogo orientado	27
Figura 5- 7 Recuadro de ayuda	28
Figura 5- 8 Diagrama de flujo para la programación en Arduino.	33
Figura 5- 9 Conexiones entre el Arduino Uno y el DAC.....	37
Figura 5-10 Esquema para el lazo de realimentación.....	38
Figura 6- 1 Representación Gráfica al Sistema de un Tanque Agitado.....	39
Figura 6- 2 Respuesta a un Sistema de Primer Orden (Tanque Agitado).....	40
Figura 6- 3 Respuesta a un Sistema de Segundo Orden (Motor DC sobre Plataforma ARM).....	41
Figura 6- 4 Respuesta a un Sistema de Segundo Orden (Servomotor DC)	42
Figura 6- 5 Respuesta a un Sistema de Tercer Orden (Planta de velocidad para una transmisión hidrostática)	43
Figura 6- 6 Respuesta a un Sistema de Cuarto Orden	44

LISTA DE TABLAS

Tabla 1 Características del Arduino Uno	21
Tabla 2 Diferencia de Muestras Finitas según el Grado de la Derivada	30
Tabla 3 Matriz de Pascal	31

LISTA DE ANEXOS

Anexo 1 Algoritmo en Arduino	49
Anexo 2 Diagrama Electrónico.....	511
Anexo 3 Pistas para el Diagrama Impreso.....	52

RESUMEN

Con el presente trabajo de grado se pretende exponer el estudio realizado para el diseño e implementación de un emulador de plantas dinámicas. Este estudio tiene como finalidad mejorar el entorno y las prácticas educativas para un óptimo aprendizaje en el área de control de procesos de los estudiantes de Ingeniería Electrónica de la Universidad Surcolombiana. El software se diseñó en el entorno de desarrollo integrado de Arduino y la interfaz gráfica de usuario en Matlab. En el App Designer de Matlab se desarrolló una aplicación independiente que presenta la interfaz de comunicación donde el usuario digita los coeficientes de la función de transferencia a emular, luego los datos recibidos se envían al Arduino donde se hace la respectiva discretización. El hardware consta de tres etapas, la primera mejora la resolución del dispositivo y se construyó por medio de arreglos de amplificadores operacionales, la segunda consta del microcontrolador Arduino Uno, donde se encuentra la mayor parte de la programación del dispositivo, la tercera fase consistió en la conversión de la señal digital a analógica implementada con el dispositivo DAC MCP4725. Esto en conjunto permitió la realización de un práctico y funcional dispositivo que proporcionará a los estudiantes la oportunidad de emular una mayor cantidad de procesos.

Palabras clave: App Designer, DAC, Emulador, Función de transferencia, IDE Arduino, Microcontrolador, Plantas dinámicas, Resolución.

ABSTRACT

With this degree work it is intended to expose the study carried out for the design and implementation of a dynamic plant emulator, this study aims to improve the environment and educational practices for optimal learning in the area of process control of students of Electronic Engineering of the Surcolombiana University. The software was designed in the Arduino integrated development environment and the Designer App. In the Designer App the communication interface was built where the user inscribes the transfer function to emulate, then the received data is sent to the Arduino IDE where the respective discretization is done. The hardware consists of three stages, the first stage, improves the resolution of the device and was built by means of arrangements of operational amplifiers, the second consists of the Arduino One microcontroller, where most of the programming of the device is located, the third phase It consists of the conversion of the digital signal to analog implemented with the MCP4725 DAC device and all this together allowed the realization of a practical and functional device that will provide students with the opportunity to control a greater number of processes.

Keywords: App Designer, DAC, Dynamic plants, Emulator, IDE Arduino, Microcontroller, Resolution, Transfer function.

INTRODUCCIÓN

Este trabajo de investigación muestra el desarrollo de un prototipo portable que emula procesos por medio del comportamiento dinámico de funciones de transferencia lineales.

Con la intención de mejorar las prácticas pedagógicas de los estudiantes de Ingeniería Electrónica y ante la carencia de algunos elementos indispensables para la dinamización en el proceso de enseñanza, se pretende implementar un dispositivo que se sumará a los instrumentos y equipos electrónicos utilizados para las prácticas de laboratorio concernientes a automatización y control de procesos.

El dispositivo permitirá al estudiante de ingeniería electrónica la emulación de plantas lineales desde el mismo módulo.

El software se diseñó mediante el entorno de programación visual App Designer de Matlab y el entorno de desarrollo de Arduino, mientras que el hardware consta de unas etapas amplificadoras, la placa de microcontrolador Arduino Uno y el Convertidor Digital-Análogico MCP 4725.

La construcción del emulador cuenta con las siguientes fases de desarrollo: en primera medida se implementaron circuitos que permitieron la mejora en el rango de voltajes admitidos por el microcontrolador por medio de amplificadores operacionales. En segunda medida se creó una interfaz gráfica que permite una fácil interacción con el usuario y se utilizó la comunicación serial para el proceso de envío de datos entre Arduino y Matlab. Como tercera medida se hicieron algunas restricciones que el usuario debe tener en cuenta a la hora de utilizar el dispositivo. Como cuarta medida se muestra claramente el proceso de discretización de una función de transferencia lineal por el método de Euler hacia atrás con su respectiva programación en Arduino. Como quinta medida se propone una práctica de laboratorio con el fin de que los estudiantes puedan elaborar el dispositivo y como sexta y última medida se dan a conocer los resultados obtenidos y se hace su respectivo análisis.

1. JUSTIFICACIÓN

El ser humano ha impuesto una hegemonía con respecto a los demás seres vivos en el planeta. Haciendo uso de los diferentes objetos de su entorno ha logrado crear artefactos que en otros tiempos eran inimaginables, mejorando la calidad de vida de cada individuo de su especie. Este proceso ha sido el resultado del trabajo de mentes brillantes que con el paso de los años han aportado su grano de arena al desarrollo de la humanidad. Cada proceso tecnológico lleva consigo la importancia de la exactitud y la precisión en la medición de variables. No se puede construir un edificio sin el estudio milimétrico de cada una de sus estructuras, sin que se analicen variables como la presión, la altura, el peso, entre otras.

Uno de los más grandes propósitos que se han planteado las sociedades ha sido controlar las distintas variables que se encuentran en el entorno y usarlas para su beneficio. En el camino han creado sistemas que miden, identifican y controlan variables como la velocidad, la temperatura, la posición y otras tantas.

En las prácticas educativas que se llevan a cabo en pro de formar estudiantes en las áreas de automatización y control, se hace necesaria la interacción con este tipo de sistemas. Los establecimientos educativos optan por adquirir plantas que permitan a los estudiantes aprender a identificar, medir y controlar procesos muy parecidos a los que presenta la industria.

Con el fin de mejorar el ambiente para los estudiantes de Ingeniería Electrónica se crea un prototipo portable que permitirá que la persona a instruirse pueda emular plantas lineales mediante su modelo matemático representado con la función de transferencia.

Se pretende que los estudiantes de Ingeniería Electrónica de la Universidad Surcolombiana puedan contar con un dispositivo de fácil manejo, portable y muy versátil, que les permita emular plantas lineales.

Se ha propuesto que los profesores del área de Control de la Universidad utilicen este proyecto de grado para que los estudiantes en sus prácticas de laboratorio puedan implementar el emulador, que se ha desarrollado con un lenguaje sencillo para garantizar su fácil asequibilidad en cuanto a economía se refiere.

2. OBJETIVOS

2.1. OBJETIVO GENERAL

Diseñar e implementar un prototipo que permita emular el comportamiento dinámico de plantas lineales.

2.2. OBJETIVOS ESPECÍFICOS

- Diseñar un software que sea capaz de emular el funcionamiento de una planta real.
- Implementar un hardware que me brinde las características necesarias para la implementación del software a trabajar.
- Proponer prácticas de laboratorio: una a lazo abierto para verificar el funcionamiento del emulador con respecto a plantas reales y otra práctica de laboratorio a lazo cerrado para verificar el sistema controlado.

3. PLANTEAMIENTO DEL PROBLEMA

Conociendo la importancia de la automatización en todos los ámbitos de producción en el mundo para el manejo de procesos, se da cuenta que de allí se desglosa un abanico de variables a controlar, entre ellas, la temperatura, la velocidad, la presión, el movimiento, entre otras. El control de estas variables en las plantas hace que los procesos sean idóneos para que la producción se haga de manera eficiente. Si no se utilizara el control en las plantas industriales, probablemente se necesitaría un operario para cada proceso; y esto no sería suficiente, dado que las mediciones y los actuadores tienen que operar instantáneamente en el proceso y se mide de manera milimétrica en cada variable, técnicas que tal vez ningún individuo podría realizar.

Utilizar el control en plantas industriales para el mejoramiento de procesos es la meta de muchos Ingenieros Electrónicos, dado que con ello se mejora la eficiencia de los mismos. El reto de los maestros, es enseñar y mostrar estos procesos de control lo cual se vuelve complicado, dado que las plantas existentes en el laboratorio de control empleadas para la práctica de los estudiantes son escasas, lo que provoca aglomeraciones de estudiantes al momento de interactuar con ellas. Esto implica que no se logren óptimos resultados al realizar las prácticas, haciéndose demorado y en ocasiones tedioso. Por otro lado, siendo las plantas existentes elementos reales, con el tiempo muestran descalibración y desgaste; lo que hace que su funcionamiento no sea el mejor para la práctica.

FORMULACIÓN DEL PROBLEMA

En consecuencia, se plantea la siguiente pregunta de investigación. ¿Es posible crear un dispositivo emulador de plantas dinámicas que ayude a potenciar el aprendizaje en los estudiantes de Ingeniería Electrónica?

4. MARCO TEÓRICO

Este apartado está orientado a recopilar las consideraciones que diversos autores han planteado en torno a los conceptos que soportan el presente proyecto de investigación.

4.1 ANTECEDENTES DE LA INVESTIGACIÓN

A continuación, se menciona de dos investigaciones encontradas y que se relacionan con esta propuesta investigativa. Un primer trabajo es el de Esteban Velo Sánchez¹, de la Universidad de Coruña, quien a través de su investigación “Desarrollo y testeo de un emulador de plantas industriales basado en Arduino”, creó un dispositivo portable que emulaba algunas plantas que se encontraban preestablecidas dentro del microcontrolador del Arduino, estas se podían escoger por medio de un teclado portátil y el menú se podía visualizar en una pantalla LCD, el proceso de discretización lo hicieron por el método de la transformada bilineal, lo que hace que el dispositivo sea bastante robusto, ya que no depende del tiempo de muestreo aplicado.

Como segundo trabajo, se presenta el expuesto por: Brayan David Castro², de la Universidad de Pamplona, a través del trabajo de investigación llamado: “Desarrollo de un sistema portátil emulador de dinámicas de procesos de primer y segundo orden integrado con sistemas de control clásico”. Este trabajo se enfocó en emular plantas de primer y segundo orden. Las plantas de primer orden se emulaban con filtros pasa bajo a los que se les acondicionaba un circuito seguidor de tensión en su etapa final, debido a la poca corriente de la carga del filtro, mientras que, para las plantas de segundo orden, fue creado con un arreglo de cuatro amplificadores operacionales que simulaban sistemas con diferente factor de amortiguamiento.

Con respecto a las anteriores propuestas investigativas, se presentó un prototipo totalmente diferente, en cuanto a estructura de software y hardware. La principal ventaja con respecto a la primera investigación es que el prototipo expuesto en esta, puede emular mayor cantidad de plantas y además su software se diseñó en dos etapas, la primera se presentó mediante una aplicación en el entorno de diseño interactivo App Designer, que permite instalarse en cualquier ordenador sin necesidad de tener otros software instalados y en segunda medida se presenta el software creado para el microcontrolador utilizando el método de discretización Euler hacia atrás. Cabe mencionar que el dispositivo tiene mejoras en cuanto a la interacción con el usuario, debido a que este puede digitar diferentes funciones de transferencia, mientras que la propuesta de Velo Sánchez, solo permite escoger algunas que ya están preestablecidas, además la aplicación también le permite al usuario visualizar en tiempo real la respuesta a las plantas sin ningún sistema de adquisición de datos.

Con respecto al segundo trabajo se tiene más ventajas, ya que el dispositivo propuesto por Brayan Castro solo puede emular algunas plantas de primer y segundo orden, mientras que el emulador propuesto en esta investigación puede emular plantas hasta de cuarto orden, siempre y cuando se cumpla con las restricciones propuestas en el parágrafo 5.4, además el prototipo tiene un excelente interfaz de usuario y es de fácil manejo.

Con lo anterior expuesto queda claro que el prototipo propuesto es totalmente diferente a los presentados en otras investigaciones y cuenta con algunas ventajas que quedan evidenciadas a lo largo del libro.

¹ VELO, Esteban. Desarrollo y testeo de un emulador de plantas industriales basado en Arduino. Universidad de Coruña. España. (2014).

² CATRO, Brayan. Desarrollo de un sistema portátil emulador de dinámicas de procesos de primer y segundo orden integrado con sistemas de control clásico. Revista Ingeniería e Innovación. Universidad de Pamplona, Norte de Santander, Colombia. (2021).

4.2 ELEMENTOS TEÓRICOS UTILIZADOS PARA EL SOFTWARE

Teniendo en cuenta que lo que se pretende mediante el presente proyecto investigativo, es emular el comportamiento dinámico de plantas lineales, es pertinente iniciar por desarrollar el concepto de emulador, que tiene como propósito desempeñar la misma funcionalidad de determinado dispositivo o software. Es de esa manera como se responde al concepto de emulación, que para Roldán “consiste en la imitación del comportamiento de un ente software o hardware en otro ente distinto al que se intenta imitar, de manera que el usuario tenga una experiencia de interacción similar a la del ente original”³. Para este punto el autor refiere dos conceptos que suelen ser objeto de confusión, emulador y simulador. El primero imita el funcionamiento interno y externo del sistema; mientras que el segundo, centra sus esfuerzos solo en imitar el comportamiento interno⁴, en otras palabras, el primero puede imitar hardware y software mientras que el segundo, el software.

Ahora, como el objeto a emular son las plantas dinámicas, es oportuno aludir a las aproximaciones teóricas que destacados estudiosos han aportado en torno a estas. Uno de ellos es Ogata⁵, para quien una planta es un conjunto de elementos de determinada máquina que funcionan juntos y que tienen como objetivo una función en particular. Los sistemas dinámicos son aquellos cuyo estado evoluciona con el tiempo. En otro de sus trabajos, Ogata⁶ menciona que un sistema es dinámico cuando su salida en el presente depende de una entrada en el pasado; es decir que la salida del sistema no anticipa valores futuros de la entrada. Cabe aclarar que las plantas dinámicas pueden clasificarse como lineales e invariantes en el tiempo (LTI) y no lineales. Dada la complejidad de las últimas, el presente trabajo se enfoca en plantas lineales que, de acuerdo con Perez et al⁷, deben cumplir con los principios de escalamiento y superposición. En el primer caso, si las entradas son multiplicadas por un factor, las salidas deben multiplicarse igualmente por este. En el segundo, la adición de dos o más acciones es la suma del efecto de cada acción. Además, un sistema es lineal si la salida sigue fielmente los cambios producidos en la entrada.

Hasta el momento han quedado dilucidados los conceptos más relevantes en lo respectivo a emuladores y plantas dinámicas. Sin embargo, es importante hacer énfasis en que llevar a cabo una emulación de dicha índole requiere necesariamente realizar un modelamiento matemático, que, para algunos autores “es una función donde se acopla a un proceso en términos matemáticos que permite entender de una manera clara, examina y hace predicciones sobre ellos”⁸, para otros como Hughes⁹, son aquellos que se construyen para desarrollar procesos de inferencia sobre ciertos aspectos de sistemas reales previamente observados. Estos modelos se fundamentan en leyes físicas como las de Newton, leyes de termodinámica y de fluidos, entre otras. En este punto vale la pena resaltar dos aspectos. Primero, estos modelos matemáticos no son una representación precisa de un proceso, ya que sería imposible explicar un sistema completamente con una fórmula matemática exacta¹⁰. Segundo, “Los Sistemas Lineales Invariantes en el Tiempo están escritos por una ecuación diferencial lineal con coeficientes constantes”¹¹. Con esta ecuación y mediante la transformada

³ ROLDAN, Ángel. Emulador de GAMEBOY para dispositivos móviles. Universidad de Castilla-La Mancha. Castilla. 2017. p.14.

⁴ Ibid.p.14.

⁵ OGATA, Katsuhiko. Ingeniería de control moderna. 5º Ed. Madrid: Pearson Educación S.A, 2010. p.3.

⁶ OGATA, Katsuhiko. Dinámica de sistemas. Naucalpan de Juárez: Prentice-Hall Hispanoamericana S.A, 1987. p.1.

⁷ PÉREZ, Mario Alberto; PEREZ HIDALGO, Analía y PÉREZ BERNQUER, Elisa. Introducción a los sistemas de control y modelado matemático para sistemas lineales invariantes en el tiempo Universidad Nacional de San Juan. Facultad de Ingeniería. Departamento de Electrónica y Automática. .2008. p.2.

⁸ MELINA S,2017 citado por PARRA Irvin y HOYOS Víctor. Modelamiento matemático en el proceso de cocción del pan,2018. p.24.

⁹ HUGHERS, GR citado por IZQUIERO, Luis, et al. Modelado de sistemas complejos mediante simulación basada en agentes y mediante dinámica de sistemas. En: Empiria.Vol.,16 (jul-dic,2008); 86.p.

¹⁰ PARRA, Irvin y HOYOS, Víctor, OB.CIT., p.24.

¹¹ UNIVERSIDAD PEDAGÓGICA Y TECNOLÓGICA DE COLOMBIA. Modelamiento de Sistemas LTI. 2022.

de Laplace de la entrada y la transformada de la salida se genera una expresión llamada Función de Transferencia, que expresado en otros términos por Martín, “se define como el cociente entre la transformada de Laplace de la salida y la transformada de Laplace de la entrada, bajo la suposición de que las condiciones iniciales sean cero y se puede representar de la siguiente manera”¹²

$$G(s) = \frac{X(s)}{U(s)} \quad (4.1)$$

Donde $G(s)$ representa la función de transferencia. Si se toma el denominador y se hace $U(s) = 0$, a esto se le denomina ecuación característica, donde la potencia más alta determina el orden del sistema y las raíces se conocen como los polos de la función de transferencia. De la misma manera, las raíces de $X(s) = 0$ son los ceros de la función de transferencia. Se hace referencia a estos parámetros, ya que son de vital importancia a la hora de analizar la estabilidad de los sistemas¹³.

Otra definición de función de transferencia es planteada por Chávez quien establece que,

es una representación conveniente de un sistema dinámico invariante en el tiempo lineal. Matemáticamente, la función de transferencia es una función de variables complejas. Para sistemas de dimensión finita, la función de transferencia es simplemente una función racional de una variable compleja. La función de transferencia se puede obtener mediante inspección o mediante simples manipulaciones algebraicas de las ecuaciones diferenciales que describen los sistemas. Las funciones de transferencia pueden describir sistemas de muy alto orden, incluso sistemas de dimensión infinita gobernados por ecuaciones diferenciales parciales.¹⁴

En el proyecto es de vital importancia la función de transferencia, debido a que esta se toma como base para saber la dinámica de los procesos que se piensan emular, así que el usuario del dispositivo deberá tenerla en cuenta como uno de los parámetros más significativos. Además, los coeficientes del numerador y el denominador de la función de transferencia son algunos de los parámetros que se ingresan manualmente desde la interfaz creada en Matlab. Ahora, para que el microcontrolador pueda reconocer las funciones de transferencia digitadas por el usuario, es necesario convertirlas a ecuaciones en diferencia.

Una ecuación en diferencia relaciona dos o más momentos de la variable dependiente con respecto a la variable independiente t que representa el tiempo, $F(t, x_t, x_{t+1}, x_{t+2}, \dots)$. También se pueden relacionar dos variables dependientes de la siguiente manera:

$$a_1x_{t+1} + a_2y_{t+1} + a_3x_t + a_4y_t = a_5 \quad (4.2)$$

Donde a_n son coeficientes constantes y $n = 1, 2, 3, \dots$ y se tienen dos incógnitas (x_t, y_t) ¹⁵.

La anterior es una expresión que relaciona distintas sucesiones, donde una de ellas es desconocida.

Una sucesión es una función $U(n)$ que está definida en el dominio de todos los enteros no negativos y se expresa como $\{U(n)\}$. Y sus valores $U(n)$, $n = 0, 1, 2, 3, \dots$ se ordenan de la siguiente manera: $U(0), U(1), U(2), \dots$

¹² MARTIN, Daprotis. Función transferencia y respuesta impulsiva. Bahía Blanca: 2011. p.2.

¹³ MONATERIO, Félix y Gutiérrez Álvaro. Introducción a los Sistemas Electrónicos de Control, 2021. p.2.

¹⁴ CHAVEZ, Washintong. Desarrollo de aplicaciones prácticas de sistemas dinámicos en Matlab/Simulink y Octave para la asignatura de control y automatismo. Guayaquil. Universidad Católica de Santiago de Guayaquil, 2020. p.15.

¹⁵ FERNÁNDEZ, Javier. Ecuaciones y Sistemas en Diferencias. Aplicaciones De La Dinámica De Poblaciones. Sevilla. Universidad de Sevilla. p.8.

Para Navarrete¹⁶, cada $U(n)$ se denota como término de la sucesión, donde la n es un número que representa el orden de los términos, así que se podría decir que una sucesión es un acomodo lineal de una infinidad de números ordenados por cierta regla.

En este proyecto, el método que se utiliza para hallar la ecuación en diferencia es el Método de Euler Hacia Atrás, el cual “se considera una aproximación hacia atrás de la derivada”¹⁷

$$y'(t) \approx \frac{y(t) - y(t-\tau)}{\tau} \quad (4.3)$$

Cuando se tiene el modelado matemático en ecuaciones diferenciales se reemplaza el término en cada derivada. Este proceso se mostrará con más detalle en el apartado 5.6 del presente proyecto.

4.3 ELEMENTOS TEÓRICOS UTILIZADOS PARA EL HARDWARE

Hasta el momento se han esbozado los aportes teóricos más importantes en cuanto al software del dispositivo. A continuación, se presenta una aproximación teórica hacia el hardware.

Parte importante en el hardware del dispositivo es la tarjeta de desarrollo, que en este caso es la tarjeta Arduino Uno, caracterizada por ser uno de los microcontroladores más usados en el mundo por su versatilidad y fácil programación a la hora de interactuar con sensores y actuadores, ya sea de naturaleza digital o analógica. Además, su bajo costo y la naturaleza open source han originado una gran cantidad de usuarios, que fomentan el intercambio de ideas y conocimiento. En términos de Ledesma, Cardoso y Torres,

Arduino es entonces una herramienta de procesamiento digital parecido a una computadora. Contiene puertos de entrada y salida digital a los cuales se les puede conectar: botones, leds, teclados o sensores digitales. Además de estos sensores mencionados, también tiene entradas analógicas que nos permite medir las señales de sensores analógicos. Para poder visualizar la información que arrojan estos sensores, la tarjeta Arduino Uno cuenta con un puerto de comunicación serial que, mediante un Puerto USB de una computadora, permite enviar y recibir mensajes digitales¹⁸.

Las principales características del Arduino Uno se estipulan en la tabla 4-1.

¹⁶ NAVARRETE MOLANO, Jenny. Introducción a las ecuaciones en diferencia. Bogotá. 2003.8p. Trabajo de grado (Matemático). Fundación Universitaria Konrad Lorenz. Facultad de Matemáticas.

¹⁷ UNIVERSIDAD POLITÉCNICA DE CATALUNYA (España), Métodos Numéricos para Ecuaciones Diferenciales Ordinarias, p.9.

¹⁸ LEDESMA, Norma; CARDOZO, Berenice y TORRES Oliva. Modelo de aprendizaje para arduino uno básico. En: Revista de Cómputo Aplicado.Vol.,3 (abr-jun,2019); 16p.

Características del Arduino Uno	
Velocidad	16 MHz
14 pines	Entrada/ Salida Digital
6 canales	Entrada analógica
1 puerto	UART
1 puerto	I2C
1 puerto	SPI
1 procesador	ATmega328P de 8 bits
Memoria flash	32 KB
RAM	2 KB
EEPROM	1 KB
Voltaje	5 voltios

Tabla 4-1 Características del Arduino Uno

El procesador de Arduino Uno es un ATmega328P el cual corre a 20 MIPS (millones de instrucciones por segundo)¹⁹. Debido a que cada ciclo del software recibe y envía información y, además, debe ejecutar procedimientos matriciales complejos, se procedió a hacer pruebas con diferentes plantas y diferentes tiempos de muestreo con el fin de determinar cuál era el mejor, con lo cual se llegó a conclusión que el tiempo de muestreo para el emulador es de 10 ms.

La distribución de pines en la tarjeta de Arduino Uno se representa en la figura 4-1.

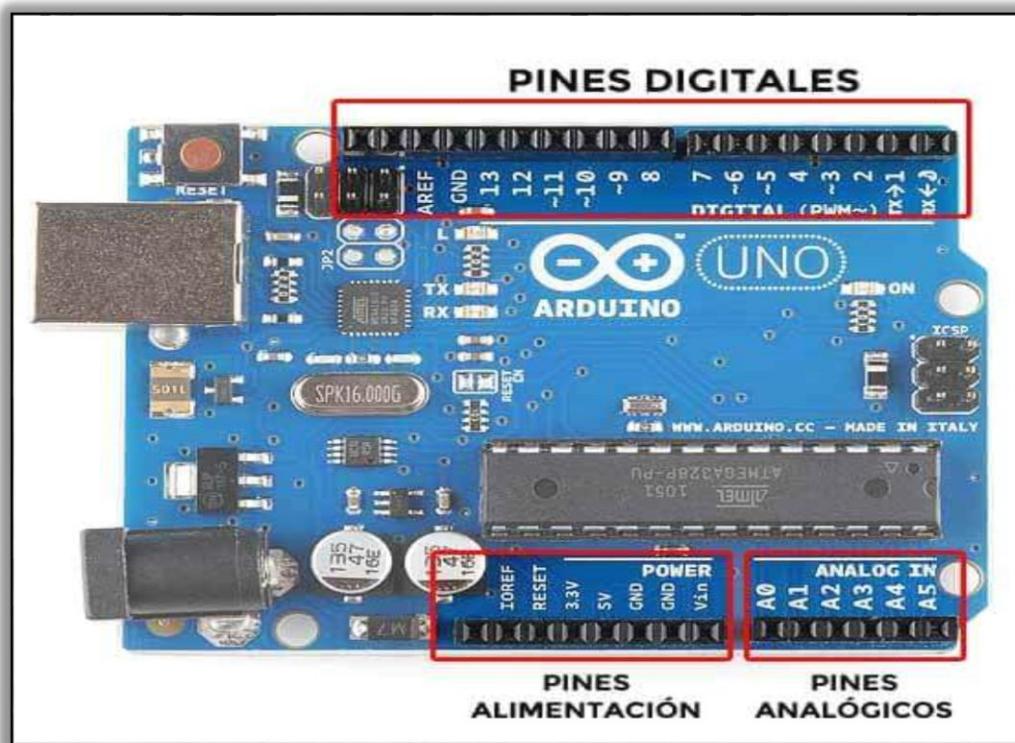


Figura 4- 1 Distribución de pines en Arduino Uno

Como el Arduino solo tiene salidas digitales y lo que se propone es presentar la señal de forma analógica, se hace necesario el uso de un dispositivo que convierta la señal digital en analógica. Es en este punto que adquiere fuerza otro dispositivo importante en el Hardware, el DAC, que para Huircán²⁰, es un dispositivo que “convierte las

19 HERRAMIENTAS TECNOLÓGICAS PROFESIONALES.

²⁰ HUIRCAN, Juan Ignacio. Conversores Análogo-Digitales y Digital-Análogo: Conceptos Básicos. En: Revista Academia. (2007) 3p.

señales digitales en cantidades eléctricas analógicas relacionadas en forma directa con el número de entradas codificadas digitalmente”.

Para el trabajo se hizo uso del dispositivo MCP4725, que como se menciona en Microchip Technology Inc.²¹, cuenta con una resolución de 12 bits, un pin para la dirección de selección, maneja Voltajes de 2,7V a 5,5V y una velocidad de trasmisión de hasta 3,4 Mbps.

En la figura 4-2 se muestra el esquema para el montaje del MCP4725 en compatibilidad con el Arduino.

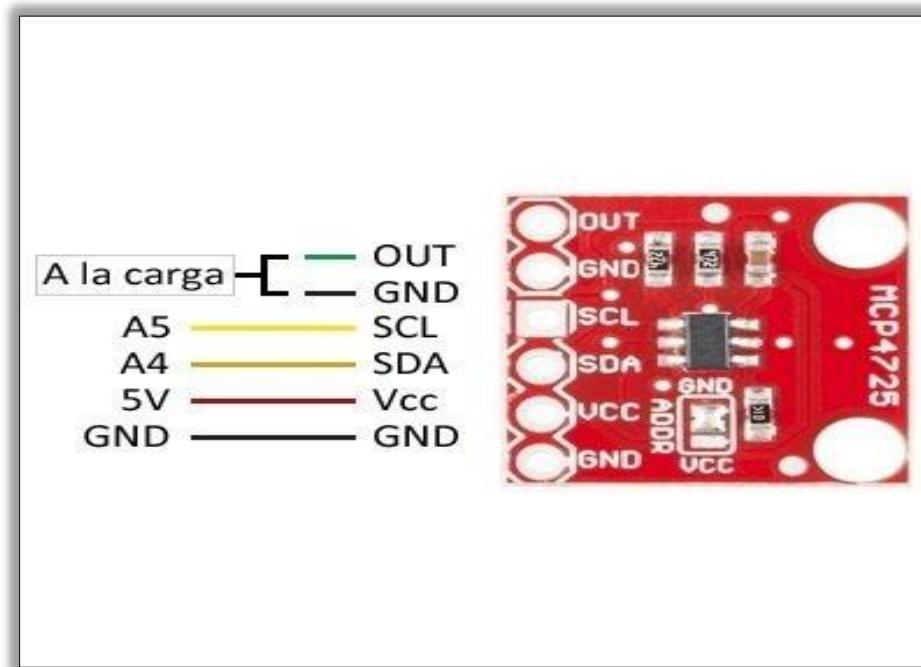


Figura 4- 2 Conexión con Pines MCP4725 con Arduino

²¹ MICROSHIP TECHNOLOGY INC. 12-Bit Digital-to-Analog Converter with EEPROM Memory in SOT-23-6.2007.

5. METODOLOGÍA

En este apartado se presenta el procedimiento utilizado para la consecución del prototipo, explicando cada etapa llevada a cabo para la creación del mismo.

La secuencia de flujo de información en el dispositivo se ilustra en la figura 5-1

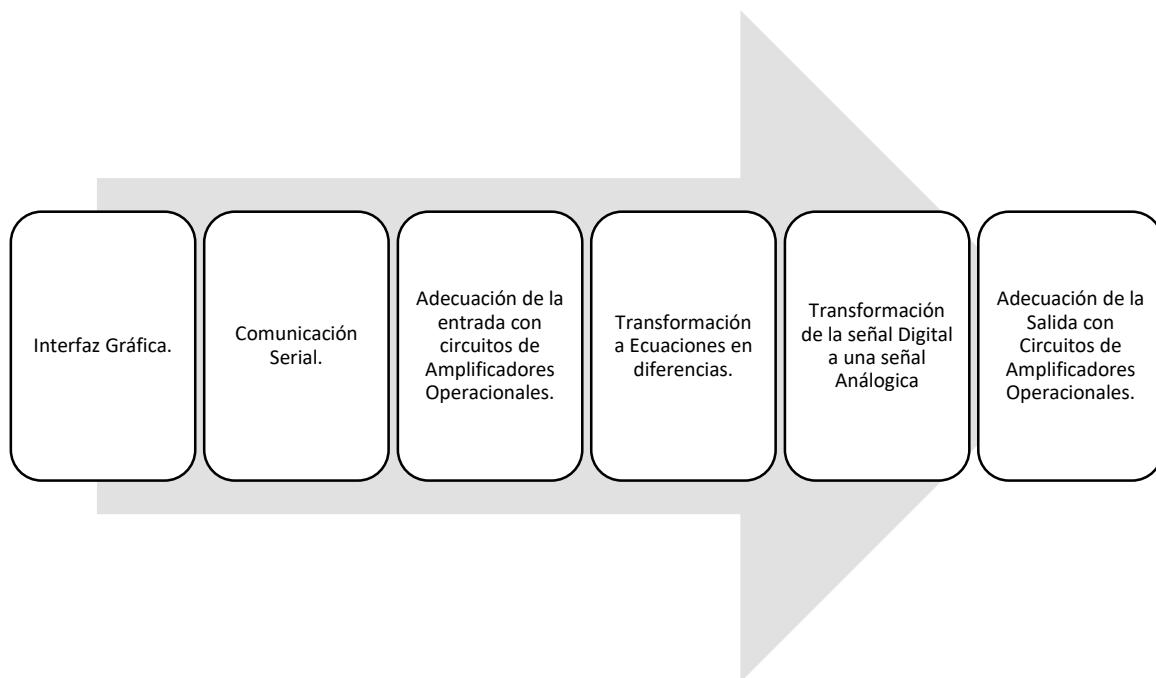


Figura 5- 1 Secuencia para la elaboración del dispositivo

En base a la figura 5-1 se muestra la secuencia de flujo de información en el dispositivo. En primera medida, se creó un interfaz en el App Designer de Matlab, donde el usuario puede interactuar con el dispositivo, en esta sección el usuario tiene la opción de introducir funciones de transferencia y el intervalo de tiempo que desea que esta se ejecute. Luego se estableció la comunicación serial de Arduino con el ordenador y con el Convertidor Digital-Analógico mediante el bus de transmisión I2C²². Debido a que el microcontrolador solo maneja entradas entre 0 y 5 voltios, se decidió hacer un arreglo de Amplificadores Operacionales con el fin de que se recibiera un rango mayor de voltajes de entrada. Una vez el Arduino ha recibido los datos ingresados por el usuario, la función de transferencia se debe convertir a ecuaciones en diferencias. El proceso cíclico generado en Arduino proporciona valores que son enviados al convertidor, con el fin de que lo que perciba el usuario sea una respuesta analógica y, como paso final, se hizo otro arreglo de Amplificadores Operacionales, con el cual se buscó escalar la salida del DAC para mostrar voltajes que estuvieran en el mismo rango de valores que en la entrada.

5.1 ADECUACIÓN DE CIRCUITOS A LA ENTRADA Y LA SALIDA PARA MEJOR RESOLUCIÓN DEL DISPOSITIVO

Como la tarjeta de desarrollo usada solo recibe voltajes de cero a cinco voltios, se optó por plantear dos circuitos que le permitieran al prototipo recibir y entregar un rango más amplio de voltajes.

²² MORENO, Antonio. BUS I2C. Universidad de Córdoba. España. (2004). p.183.

5.1.1 Adecuación de Circuito de Entrada. En Jen-Hsing²³ se muestra un arreglo de amplificadores operacionales, que permiten tener un rango de valores que van desde -10 a 10 voltios en la entrada. Este arreglo se presenta en la Figura 5-2.

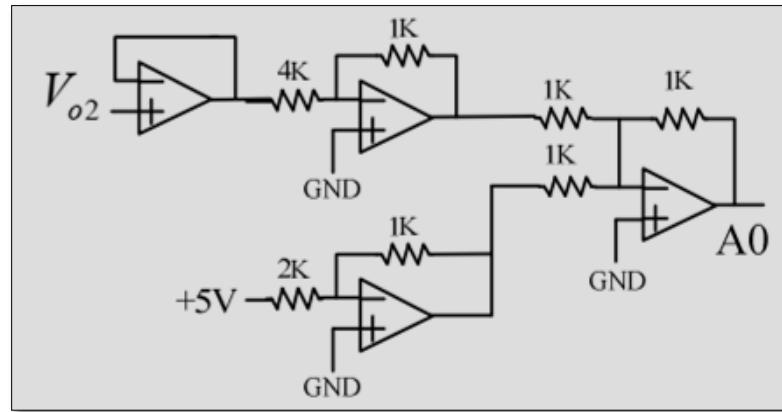


Figura 5- 2 Arreglo de amplificadores para la entrada del emulador

La siguiente ecuación relaciona la entrada con la salida del circuito.

$$A_0 = -\left(-\frac{V_{o2}}{4} - \frac{5}{2}\right) = \frac{V_{o2}}{4} + 2.5 \quad (5.1)$$

Teniendo en cuenta esta relación, cuando el voltaje de entrada es igual a 10V el voltaje en la salida A_0 es igual a 5V y cuando la entrada al circuito es igual a -10V se tiene que A_0 es igual a 0V. Este escalado permite que el voltaje a la entrada del microcontrolador sea el idóneo y admita un amplio rango de valores posibles de voltajes en la entrada del emulador.

Es importante hablar también de los rangos de voltaje en la salida de la planta, porque en general, las dinámicas de los sistemas tienden a seguir la señal que se le haya ingresado a la entrada. Así que si se tiene un rango de -10 a 10 V en la entrada, se hace necesario que la salida tenga el mismo rango de voltajes. Esto se hace posible por medio de un arreglo de amplificadores, que se presenta en la figura 5-3.

5.1.2 Adecuación de Circuito a la Salida. A la salida del DAC se tienen valores entre 0 a 5 V, esta salida se representa como V_8 en el circuito y se conecta a la entrada del arreglo de amplificadores que tienen como relación de entrada y salida la ecuación 5.2.

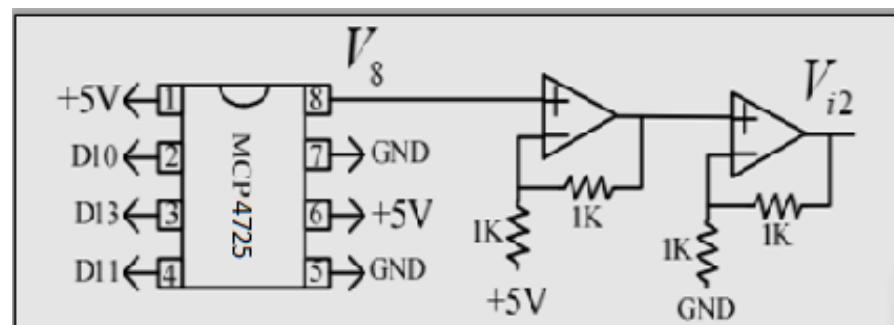


Figura 5- 3 Arreglo de amplificadores para la salida del emulador.

$$V_{i2} = 4V_8 - 10 \quad (5.2)$$

Lo que hace que cuando el voltaje a la salida del DAC sea igual a cero $V_{i2} = -10$ V y cuando $V_8 = 5$ V se tiene que $V_{i2} = 10$ V.

²³ JEN-HSING, Li. Control System Laboratory with Arduino. En: IEEE, (2018); p.2

Con lo anterior expuesto se mejora el rango de valores admisibles a la entrada y salida del dispositivo.

Posteriormente se procede a diseñar el interfaz de usuario, que viabiliza la comunicación entre el operador y el emulador. Para ello se utilizó el App Designer de Matlab.

Es fundamental que quien se interese por manipular el dispositivo de manera adecuada, posea conocimientos básicos en control de procesos.

5.2 PROGRAMACIÓN EN MATLAB

Para el diseño de la programación en Matlab se utilizaron tres funciones, una función menu, para la adquisición y ordenamiento de los datos en un vector que se llamó rx, una llamada disc, que se encarga de graficar la función de transferencia ingresada por el usuario y una última llamada enviar, que se encarga de la comunicación con el Arduino y sirve para el envío y recepción de datos.

5.2.1 Función menu. Esta función toma los datos extraídos de la interfaz gráfica, los ordena de manera adecuada para los procesos en Arduino y luego este vector es llamado desde la función enviar. En la figura 5-4 se muestra un diagrama de flujo que representa dicha función.

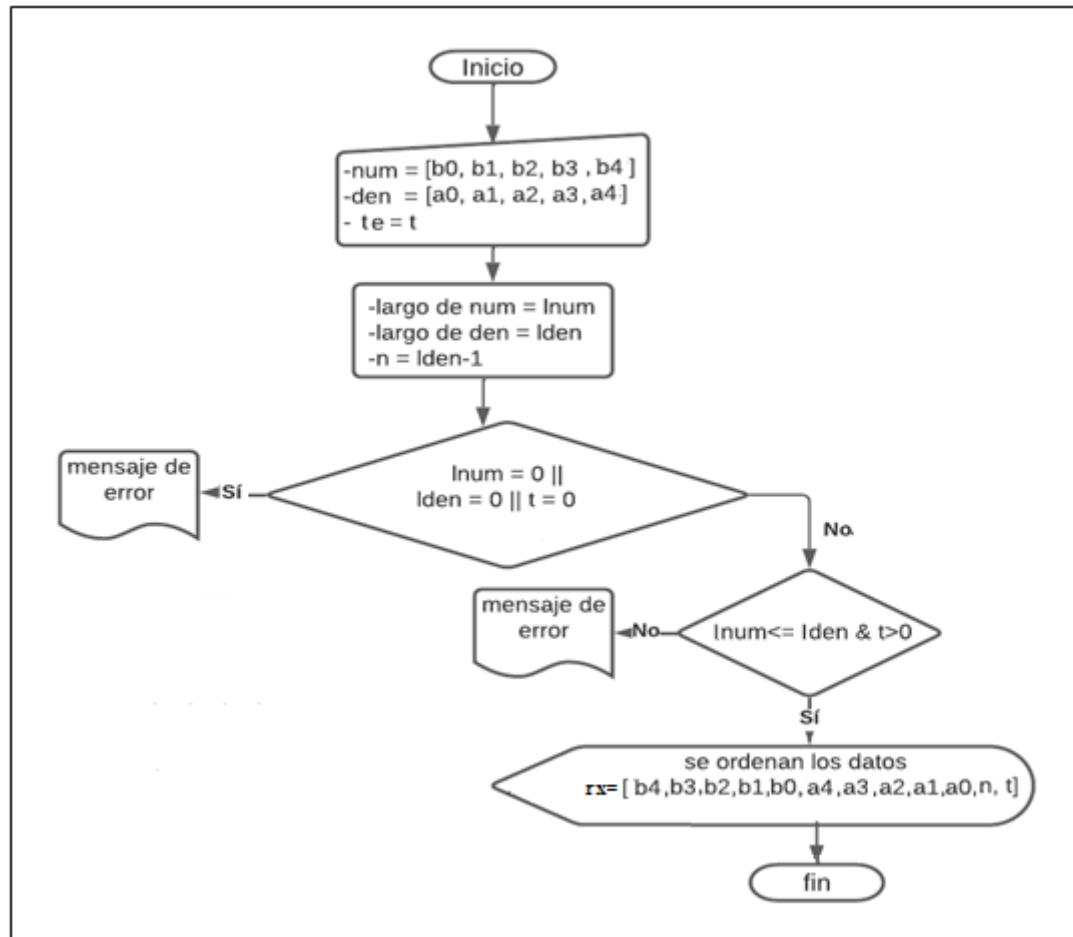


Figura 5- 4 Diagrama de flujo para la función menu.

En donde num representa los coeficientes del numerador, den representa los coeficientes del denominador, te es el tiempo de ejecución del sistema, n es el orden de la planta y en algunas partes del libro rx se toma como datos. En esta función también se hacen algunas restricciones que se presentarán en el parágrafo 5.4, si la persona que use el prototipo pasa por alto estas restricciones se le hará saber por medio de un recuadro de dialogo para que reconsidera los datos que ha ingresado.

5.2.2 Función Disc. Esta función toma los valores del Numerador, Denominador y tiempo de ejecución para generar la Función de Transferencia en Matlab y simular la gráfica correspondiente a dicha función. También recibe los valores del prototipo en tiempo real y los grafica simultáneamente.

A continuación, se presenta un resumen del seudocódigo de esta función:

Disc (num, den, te):

- Halla orden del sistema $n = \text{longitud del denominador} - 1$
- Simular función de transferencia en el tiempo de ejecución
- Recibir datos del emulador en tiempo real.
- Graficar resultados

5.2.3 Función Enviar. En este punto se inicializa la comunicación serial, buscando y abriendo el puerto disponible, por lo que hay que tener en cuenta estar usando un solo Arduino. Además, esta función luego de enviar los datos, lee por el mismo puerto serial los resultados del algoritmo de discretización que ejecuta Arduino. Finalmente cierra la comunicación y borra el contenido del puerto.

En el siguiente seudocódigo muestra cómo se concibió dicha función:

Enviar (datos):

- Hallar la cantidad de datos.
- Extraer el tiempo de ejecución.
- Calcular la cantidad de muestras.
- Buscar puerto serial habilitado para comunicación.
- Borrar la información que se encuentre en el puerto.
- Establece parámetros de comunicación, velocidad 9600 baudios y con retorno de carro y nueva línea.
- Abrir puerto.
- Enviar byte de prueba.
- Por medio de un ciclo enviar uno a uno los elementos del vector datos.
- Inicializar vector para almacenar las lecturas de la salida.
- Por medio de un ciclo limitado por la variable muestras, leer cada dato del puerto
- Cerrar, limpiar y eliminar el puerto.

5.3 INTERFAZ GRÁFICA



Figura 5- 5 Interfaz de Usuario del Emulador.

En el App Designer se creó un interfaz, donde se disponen cuatro espacios para que el usuario pueda digitar por medio del ordenador. En el primer espacio se registra el step que se ha usado en el emulador, en los dos espacios siguientes se registra el numerador y denominador de la función de transferencia. Cada espacio se ha configurado para recibir vectores de hasta 5 elementos. El formato de ingreso de estos campos es similar a la escritura que se usa para las funciones de transferencia en Matlab; es decir que se disponen los coeficientes del numerador y denominador entre corchetes y separados por espacio o coma, en el orden del máximo grado al grado 0. Si por ejemplo uno de ellos es de orden cúbico, pero no tiene valor cuadrático, se llena esta posición con un cero de la siguiente manera $cs^3 + as + x$ y se debe ingresar como $[c \ 0 \ a \ x]$ o $[c, 0, \ a, \ x]$; también es posible obviar los corchetes, pero emplearlos puede evitar posibles errores de digitación. Es importante completar el campo del tiempo de ejecución, ya que Matlab le asigna un valor por defecto que puede ser aleatorio, generando errores en la ejecución del programa; de igual forma, para simplificar este proceso, al colocar el cursor en dichos espacios, se presentará un cuadro de dialogo que explicará la forma adecuada de escribir los coeficientes, como se ilustra en la figura 5-6.

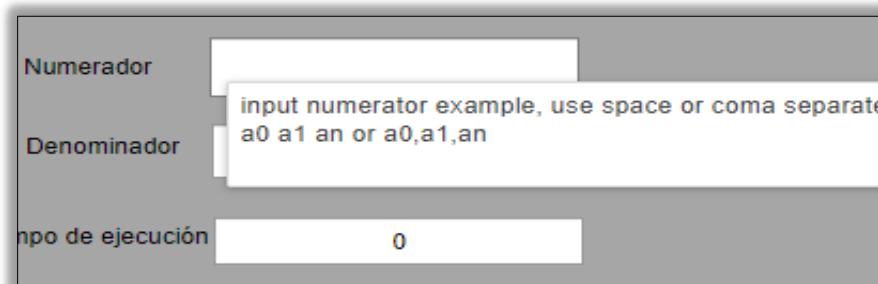


Figura 5- 6 Cuadro de dialogo orientado

En la parte superior de la interfaz gráfica se encuentra un botón de ayuda, que brinda una explicación clara sobre la manera más pertinente de digitar la información requerida, como se muestra en la figura 5-7.

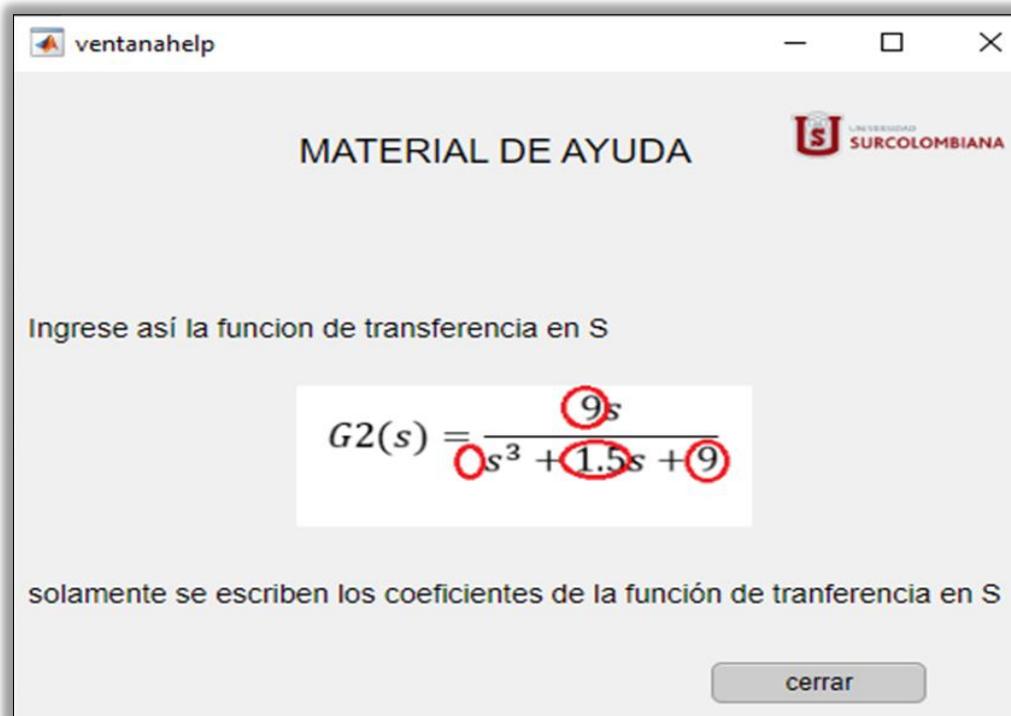


Figura 5- 7 Recuadro de ayuda

5.4 RESTRICCIONES DE USO PARA EL DISPOSITIVO

Para que se envíen los datos desde la interfaz gráfica del dispositivo al Arduino, debe cumplirse con algunas especificaciones muy precisas para que el funcionamiento sea correcto. Estas especificaciones se muestran a continuación:

- El orden del numerador debe ser menor que el orden del denominador.
- Tanto los coeficientes del numerador como los del denominador deben estar correctamente digitados.
- El tiempo de ejecución debe ser mayor que cero.
- Los espacios en donde se digita la información no deben estar vacíos al momento de ejecutar.
- No se podrán emular funciones de transferencia que superen el orden 4.
- Según Ogata, “para reconstruir una señal original a partir de una señal muestreada, existe una frecuencia mínima que la operación de muestreo debe satisfacer. Dicha frecuencia se especifica en el teorema de muestreo.”²⁴

Teorema de muestreo: si w_s , definida como $\frac{2\pi}{T}$, donde T es el periodo de muestreo, es mayor que $2w_1$, o

$$w_s > 2w_1$$

Donde w_1 es la componente de mas alta frecuencia presente en la señal en tiempo continuo. Entonces se puede deducir que si se cumple este

²⁴ KATSUHIKO, Ogata, Sistemas de Control en Tiempo Discreto, segunda edición: México. Prentice Hall Hispanoamérica S.A. 1996. P. 90.

teorema se puede reconstruir completamente dicha señal a partir de la señal muestreada. Por lo cual no todas las funciones de transferencia se podrán emular en el dispositivo propuesto en esta investigación.

El tiempo de muestreo que se eligió para el emulador fue de 0.01s debido a la cantidad de instrucciones que debe procesar el microprocesador. Además, fue un tiempo prudente para la visualización en tiempo real. Si se hubiese escogido un tiempo menor, se hace muy demorada la respuesta del sistema.

5.5 COMUNICACIÓN SERIAL ARDUINO-MATLAB

En el momento en que el usuario ha escrito de manera adecuada los coeficientes de la función, tiempo de ejecución y además haya tenido en cuenta las advertencias anteriormente presentadas, procede a pulsar el botón de ejecutar.

Por el puerto serial se envían 12 valores desde Matlab a Arduino, esta información se ha organizado de la siguiente manera: cinco coeficientes del numerador (B), cinco coeficientes del denominador (A), el orden del sistema (n), y el tiempo de ejecución (t). La información que se envía al Arduino desde la aplicación, está contenida en un vector llamado rx descrito a continuación:

$$rx = [b_4, \quad b_3, \quad b_2, \quad b_1, \quad b_0, \quad a_4, \quad a_3, \quad a_2, \quad a_1, \quad a_0, \quad n, \quad t]$$

Para leer esta información en Arduino por puerto serial, se debe tener en cuenta que tanto Matlab como Arduino deben estar configurados a una misma velocidad de transmisión y recepción de datos serial, que en este caso será de 9600 baudios.

5.6 PROCEDIMIENTO PARA DISCRETIZAR LA FUNCIÓN DE TRANSFERENCIA

Para que se dé la evaluación numérica y la posterior implementación en el Arduino, es necesario transformar la función de transferencia a ecuaciones en diferencia.

En este proyecto el proceso discretización se llevó a cabo en la tarjeta de desarrollo, con el fin de que el usuario pudiera escribir y editar una mayor cantidad de procesos sin necesidad de tener que recurrir a bases de datos con funciones de transferencia preestablecidas. Despues de algunas pruebas con los distintos métodos de discretización, se decidió que el método que más facilitaba la programación era el método de Euler hacia atrás, este método tiene la propiedad de preservar la estabilidad numérica, es decir, al emular un sistema estable con el método de Euler hacia atrás, la solución numérica resulta estable²⁵. Un factor muy importante que se tuvo en cuenta al momento de elegir el método de discretización para el emulador.

Tomando en consideración lo expuesto por Rovveto, “el método de Euler utiliza la pendiente (primera derivada) para predecir un nuevo valor de la función y luego extrapolar en forma lineal sobre el tamaño del paso, la pendiente se aproxima a partir de un intervalo inicial y con ello se obtiene un nuevo punto en el que se reevalúa la pendiente y así sucesivamente”²⁶.

Tomando como referencia a Ayala²⁷, se puede hallar la pendiente en cada punto usando una aproximación de las derivadas, por medio del método de las Diferencias Finitas.

²⁵ INTRODUCCIÓN A LOS MÉTODOS DE INTEGRACIÓN NUMÉRICA DE ECUACIONES DIFERENCIALES ORDINARIAS (ARGENTINA).

²⁶ CHAPRA Y CANALE citado por ROVETTO, Carlos. Métodos Numéricos para Ingenieros. Universidad Tecnológica de Panamá, (junio de 2021); p.91.

²⁷ AYALA, Tita Paola, Análisis del Método de Disparo Lineal y los Métodos de Diferencias Finitas para problemas de valores en la frontera. En: Universidad Nacional José Faustino Sánchez Carrión, Perú.2021. p.23.

Para el proceso anteriormente descrito, se reemplaza cada derivada de la ecuación diferencial por su equivalente en diferencias finitas. En la tabla 5-1 se presenta dicha equivalencia.

Grado Derivada	Derivadas
0	$x(t - 1)$
1	$\frac{x(t - 1) - x(t - 2)}{T}$
2	$\frac{x(t - 1) - 2x(t - 2) + x(t - 3)}{T^2}$
3	$\frac{x(t - 1) - 3x(t - 2) + 3x(t - 3) - x(t - 4)}{T^3}$

Tabla 5-1 Diferencia de Muestras Finitas según el Grado de la Derivada

Se procede a discretizar una función de transferencia de orden dos, con el fin de que sea más claro el método que se usó.

La función de transferencia se representa de la siguiente forma:

$$\frac{Y(s)}{U(s)} = G(s) = \frac{b_0s^2 + b_1s + b_2}{a_0s^2 + a_1s + a_2} \quad (5.3)$$

Separando la salida de la entrada tenemos:

$$Y(s)[a_0s^2 + a_1s + a_2] = U(s)[b_0s^2 + b_1s + b_2] \quad (5.4)$$

Lo que representado en el tiempo sería:

$$a_0y(t)'' + a_1y(t)' + a_2y(t) = b_0u(t)'' + b_1u(t)' + b_2u(t) \quad (5.5)$$

Reemplazamos las expresiones que contienen derivadas teniendo en cuenta la tabla 5-1, se debe tener en cuenta que los sistemas a evaluar son causales, por consiguiente, la primera muestra la tenemos en el tiempo t .

$$\begin{aligned} a_0 \frac{y(t) - 2y(t - 1) + y(t - 2)}{T^2} + a_1 \frac{y(t) - y(t - 1)}{T} + \\ \dots a_2y(t) = b_0 \frac{u(t - 1) - 2u(t - 2) + u(t - 3)}{T^2} + \\ \dots b_1 \frac{u(t - 1) - u(t - 2)}{T} + b_2u(t - 1) \end{aligned} \quad (5.6)$$

Se reorganiza para despejar $y(t)$

$$\begin{aligned} a_0[y(t) - 2y(t - 1) + y(t - 2)] + a_1T[y(t) - y(t - 1)] + \\ \dots a_2T^2y(t) = b_0[u(t - 1) - 2u(t - 2) + u(t - 3)] + \\ \dots b_1T[u(t - 1) - u(t - 2)] + b_2T^2u(t - 1) \end{aligned} \quad (5.7)$$

Luego:

$$\begin{aligned}
 & b_0[u(t-1) - 2u(t-2) + u(t-3)] + \\
 & \dots b_1T[u(t-1) - u(t-2)] + b_2T^2u(t-1) - \\
 y(t) = & \frac{\dots [a_0[-2y(t-1) + y(t-2)] + a_1T[y(t-1)]]}{[a_2T^2 + a_1T + a_0]} \tag{5.8}
 \end{aligned}$$

Para que la programación fuese menos compleja, se representó el procedimiento anterior de forma matricial. Para ello se incluye la matriz de pascal que se denota con la letra P y es de tamaño $(n+1) \times (n+1)$, donde n es el orden del sistema. En este caso se ha dispuesto que las funciones de transferencia tengan como máximo un cuarto orden, lo que indica que P será de tamaño 5×5 .

	1	0	0	0	0
	1	-1	0	0	0
$P =$	1	-2	1	0	0
	1	-3	3	-1	0
	1	-4	6	-4	1

Tabla 5-2 Matriz de Pascal

Una función de transferencia se puede discretizar utilizando las operaciones matriciales indicadas. A continuación, se muestra este procedimiento para una función de transferencia de orden dos.

$$y(t) = \frac{\left[\begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix} \cdot \begin{bmatrix} T^2 \\ T^1 \\ T^0 \end{bmatrix} \right]^T \cdot \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \cdot \begin{bmatrix} u(t-1) \\ u(t-2) \\ u(t-3) \end{bmatrix} - \left[\begin{bmatrix} a_1 \\ a_0 \end{bmatrix} \cdot \begin{bmatrix} T^1 \\ T^0 \end{bmatrix} \right]^T \cdot \begin{bmatrix} -1 & 0 \\ -2 & 1 \end{bmatrix} \cdot \begin{bmatrix} y(t-1) \\ y(t-2) \end{bmatrix}}{\begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix}^T \cdot \begin{bmatrix} T^2 \\ T^1 \\ T^0 \end{bmatrix}}$$

Los vectores propuestos cuando se operan con el punto chico (.), se multiplican término a término, generando un vector del mismo tamaño que los vectores operados. Cuando los vectores o matrices son operados con el punto grande (•), se debe hacer una multiplicación matricial de dichos arreglos. Después de ejecutar adecuadamente las operaciones se conseguirá la misma ecuación que se presentó con el método de Euler hacia atrás.

5.7 PROCESO DE PROGRAMACIÓN EN ARDUINO

Para el proceso de programación en Arduino se tuvo en cuenta varios aspectos importantes:

En primera medida la comunicación que se da entre Arduino-Matlab y Arduino-DAC, en segunda medida la declaración de variables, vectores y matrices necesarios para procesar la información digitada por el usuario. Por lo general esta información solo se procesará en la primera iteración del void loop. Como tercera medida se procedió a programar el proceso explicado en el parágrafo 5.6 del documento, que consiste en discretizar la señal. Para ello se dividieron los procesos y se utilizaron funciones independientes en cada una de las operaciones, almacenando sus resultados en variables que posteriormente se operan para tener en definitiva la ecuación en diferencia que se necesita.

5.7.1 Comunicación. Para la comunicación serial entre Arduino y Matlab, se verifica que la velocidad de transmisión de datos esté sincronizada. Para este caso se utilizó 9600 baudios.

Para la comunicación del DAC con Arduino Uno se hizo uso del protocolo de transmisión I2C por medio de sus pines SDA y SCL que a su vez van interconectados a los pines A4 y A5 del Arduino respectivamente. Para hacer la comunicación Arduino-DAC es necesario descargar la librería Adafruit MCP4725 en Arduino, puesto que esta permitirá la conexión con este dispositivo en específico.

Como paso siguiente el Arduino recibe el vector de datos que se ha enviado desde el ordenador y que consta de cuatro parámetros diferentes que representan: los coeficientes del numerador, denominador, orden del sistema y tiempo de ejecución.

5.7.2 Diagrama de Flujo. El siguiente diagrama de flujo muestra cómo se diseñó la programación en el entorno de Arduino:

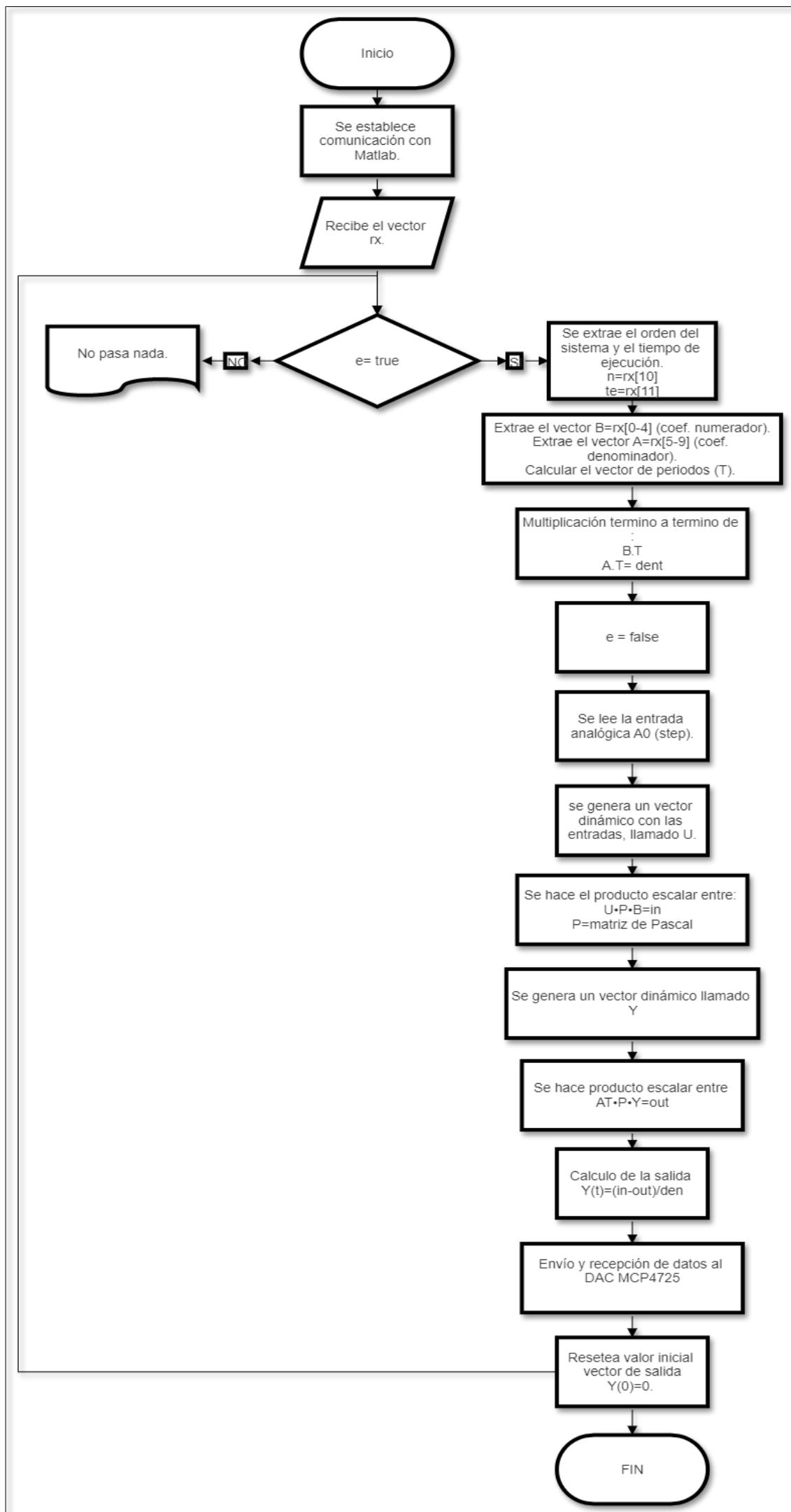


Figura 5- 8 Diagrama de flujo para la programación en Arduino.

En la parte inicial del fluograma se establece un condicional con la variable lógica *e*. Cuando esta es verdadera se leen los datos recibidos y se almacenan en el vector *rx*.

$$rx = [b_4, b_3, b_2, b_1, b_0, a_4, a_3, a_2, a_1, a_0, n, t] \quad (5.9)$$

Posteriormente se procede a extraer de este el orden del sistema y el tiempo de ejecución. Despues, a través de la función llenar, creamos los vectores *A* y *B* que se forman a través de lo que se ha leído en *rx*.

$$B = \begin{bmatrix} b_n \\ b_{n-1} \\ \vdots \\ b_1 \\ b_0 \end{bmatrix}; A = \begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_1 \\ a_0 \end{bmatrix} \quad (5.10)$$

Donde *n* es el orden del sistema.

Luego, por medio de un ciclo for se genera un vector de periodos que se llamó *T*.

$$T = \begin{bmatrix} T^n \\ T^{n-1} \\ \vdots \\ T^0 \end{bmatrix} \quad (5.11)$$

A continuación, se procede a hacer algunos cálculos preliminares como la multiplicación punto a punto entre los vectores *B* y *T*.

$$\begin{bmatrix} b_n \\ b_{n-1} \\ \vdots \\ b_0 \end{bmatrix} \cdot \begin{bmatrix} T^n \\ T^{n-1} \\ \vdots \\ T^0 \end{bmatrix} = \begin{bmatrix} b_n T^n \\ b_{n-1} T^{n-1} \\ \vdots \\ b_0 T^0 \end{bmatrix} = BT \quad (5.12)$$

El producto escalar de *A* y *T*

$$\begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{bmatrix} \cdot \begin{bmatrix} T^n \\ T^{n-1} \\ \vdots \\ T^0 \end{bmatrix} = [a_n T^n + a_{n-1} T^{n-1} + \dots + a_0 T^0] = den \quad (5.13)$$

Esta operación representa el denominador de la ecuación en diferencia que se obtendrá al final.

A y *T* multiplicados punto a punto desde la segunda posición de cada vector.

$$\begin{bmatrix} a_{n-1} \\ a_{n-2} \\ \vdots \\ a_0 \end{bmatrix} \cdot \begin{bmatrix} T^{n-1} \\ T^{n-2} \\ \vdots \\ T^0 \end{bmatrix} = \begin{bmatrix} a_{n-1}T^{n-1} \\ a_{n-2}T^{n-2} \\ \vdots \\ a_0T^0 \end{bmatrix} = AT \quad (5.14)$$

Luego, lo que se hace es actualizar el tiempo, haciendo que el tiempo pasado sea igual al tiempo actual. Después, se lee la entrada analógica y se almacena en el vector U en la posición cero. Después se van desplazando y almacenando respectivamente las entradas en dicho vector de la siguiente manera.

$$U = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}; \quad U = \begin{bmatrix} u(t-1) \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}; \quad U = \begin{bmatrix} u(t-1) \\ u(t-2) \\ \vdots \\ 0 \\ 0 \end{bmatrix}; \quad U = \begin{bmatrix} u(t-1) \\ u(t-2) \\ \vdots \\ u(t-n) \\ u(t-(n+1)) \end{bmatrix} \quad (5.15)$$

El paso a seguir es hacer el producto escalar entre el vector de entradas, el vector BT , la matriz de Pascal y el vector U .

$$in = \begin{bmatrix} b_nT^n \\ b_{n-1}T^{n-1} \\ \vdots \\ b_0T^0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 1 & -3 & 3 & -1 & 0 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix} \cdot \begin{bmatrix} u(t-1) \\ u(t-2) \\ \vdots \\ u(t-n) \\ u(t-(n+1)) \end{bmatrix} \quad (5.16)$$

De la misma forma que se procedió con el vector U , se van almacenando y desplazando respectivamente las salidas, como se muestra a continuación.

$$Y = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}; \quad Y = \begin{bmatrix} y(t) \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}; \quad Y = \begin{bmatrix} y(t) \\ y(t-1) \\ \vdots \\ 0 \\ 0 \end{bmatrix}; \quad Y = \begin{bmatrix} y(t) \\ y(t-1) \\ \vdots \\ y(t-n) \\ y(t-(n+1)) \end{bmatrix} \quad (5.17)$$

Luego se procede a hacer el producto escalar entre el vector de salidas, el vector AT y la matriz de pascal desde su segunda fila y su segunda columna, como se muestra a continuación:

$$out = \begin{bmatrix} a_{n-1}T^{n-1} \\ a_{n-2}T^{n-2} \\ \vdots \\ a_0T^0 \end{bmatrix} \cdot \begin{bmatrix} -1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -3 & 3 & -1 & 0 \\ -4 & 6 & -4 & 1 \end{bmatrix} \cdot \begin{bmatrix} y(t-1) \\ y(t-2) \\ \vdots \\ y(t-n) \\ y(t-(n+1)) \end{bmatrix} \quad (5.18)$$

Con las variables in , out y den , el proceso de discretización esta casi terminado. Como último paso se halla como tal la ecuación en diferencia que se denoto con la letra Y de la siguiente manera:

$$Y = \frac{in - out}{den} \quad (5.19)$$

El dispositivo Convertidor Digital Analógico utilizado tiene una resolución de 12 bits, es decir que puede manejar valores que van desde 0 a 4095, por ello se acondiciona el valor de la salida con el siguiente factor de conversión:

$$v_o = \frac{4095 * Y}{5.0} \quad (5.20)$$

En cada iteración se resetea el valor inicial haciendo $Y = 0$ con el fin de que el resultado no se sume al valor anterior y de esta manera la gráfica emulada se aproxime a la real. Cada iteración se va almacenando en una variable contadora y se repite el ciclo hasta que se cumpla con el número de muestras indicadas.

5.8 PLANEACIÓN DE LA PRÁCTICA DE LABORATORIO

Para mostrar la funcionalidad del dispositivo, se diseña una práctica de laboratorio que permita a los estudiantes de Ingeniería Electrónica su elaboración. La práctica quedó diseñada de la siguiente manera:

5.8.1 Título de la Práctica. Emulador para Plantas Dinámicas Lineales

5.8.2 Introducción. Con el fin de mejorar la respuesta de los artefactos creados por el hombre, se han dispuesto elementos de control que permiten manejar variables como la velocidad, la presión, la temperatura, entre otras.

Cada sistema creado se puede representar con ecuaciones matemáticas. Por medio de dichas ecuaciones se puede hacer una aproximación del comportamiento dinámico de un proceso, es decir que, con el modelado matemático de un sistema, es probable determinar qué tan rápido y qué tan estable es ante cambios del medio. En el momento en que los estudiantes quieran instruirse en este ámbito, van a tener que manejar plantas dinámicas para aplicarles mejoras por medio de componentes electrónicos. Con el fin de que los estudiantes encuentren mayor variedad de procesos a la hora de aplicar controladores, en esta práctica de laboratorio se pretende hacer un dispositivo que emule dichas plantas.

5.8.3 Objetivo de la Práctica

5.8.3.1 Objetivo General. Crear un emulador para plantas dinámicas lineales.

5.8.3.2 Objetivos Específicos.

- ✓ Implementar el software proporcionado para la ejecución del emulador.
- ✓ Implementar el hardware que permita la ejecución del software dispuesto.
- ✓ Comparar la respuesta del emulador a funciones de trasferencia con una planta de segundo orden y compararla con la simulación en Matlab.
- ✓ Analizar la respuesta del emulador ante una planta de segundo orden a lazo abierto y a lazo cerrado.

5.8.4 Metodología. En primera medida se implementa el Hardware necesario para la construcción del emulador. Este consta de unas etapas amplificadoras que mejoran la resolución del dispositivo y a su vez se interconectan con el Arduino Uno y el Conversor Digital-Analógico.

En este punto es necesario mencionar la figura 5-2, donde se presenta un arreglo de Amplificadores Operacionales que proporcionará un rango de voltajes más amplio a la entrada, y a la salida de dicho arreglo posibilitará un voltaje admisible para el Arduino.

En la figura 5-9 se muestran las conexiones adecuadas entre el Arduino y el DAC.

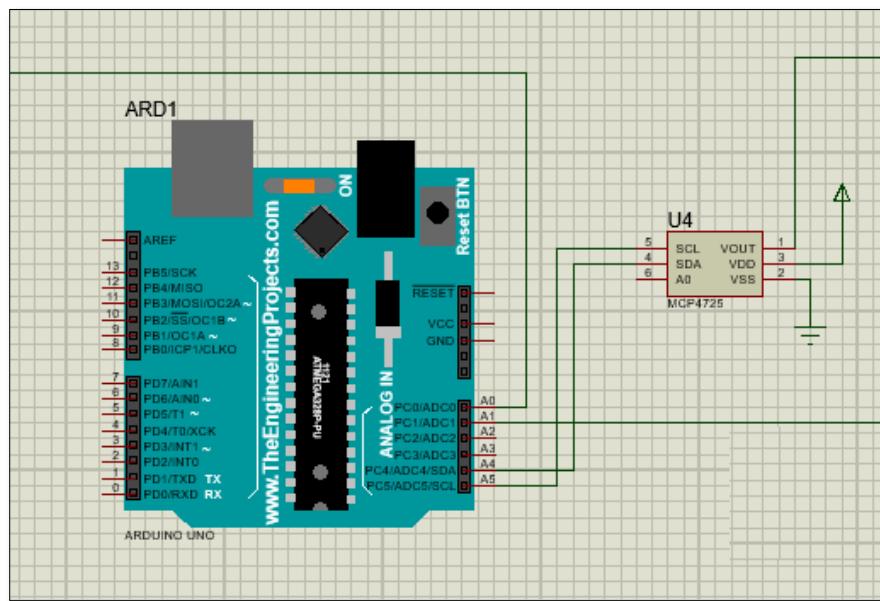


Figura 5- 9 Conexiones entre el Arduino Uno y el DAC.

Para ejecutar un siguiente paso, es importante hacer uso de la figura 5-3 en la que se representa la etapa final. En ella se muestra un arreglo de Amplificadores Operacionales que reciben el voltaje de salida del DAC y lo escalan de manera inversa al arreglo de amplificadores en la entrada.

Cabe mencionar que para la construcción del dispositivo es necesario contar como mínimo contar con Matlab 2019a.

Luego se procede a abrir la aplicación que presenta un interfaz gráfico descrito en la figura 5-4. El archivo será proporcionado por el docente a cargo por medio de una Memoria Flash (USB), que contiene el instalador del aplicativo.

5.8.5 Recursos, Materiales y Equipos

- ✓ 1 Arduino Uno
- ✓ DAC MCP4725
- ✓ Ordenador mínimo con Matlab 2019a.
- ✓ Fuentes de voltaje de 5V, 12V y -12V
- ✓ 2 chips de amplificadores TL084
- ✓ Resistores de 1kΩ, 2kΩ y 5kΩ, 40kΩ, 8kΩ, 20kΩ y 10kΩ
- ✓ 1 condensador de 10 µF
- ✓ Jumpers macho-hembra

5.8.6 Desarrollo de la Práctica. Despues de tener el hardware, se sube el código al Arduino y se abre la aplicación establecida.

La interfaz de Matlab dispone de cuatro campos para ser ingresados por el usuario, los cuales se escriben por medio de teclado. En primera medida está el valor del step. En segunda medida los coeficientes del numerador y coeficientes del denominador de la función de transferencia que pueden recibir vectores de hasta 5 elementos. El formato de ingreso de estos campos es similar a la escritura que se usa para las funciones de transferencia en Matlab; es decir que se disponen los coeficientes del numerador y denominador separados por espacio o coma, en el orden del máximo grado al grado cero. Si por ejemplo uno de ellos es de orden cúbico, pero no tiene valor cuadrático, se llena esta posición con un cero de la siguiente manera $cs^3 + as + x$ y se debe ingresar como $[c \ 0 \ a \ x]$. En el cuarto espacio se escribe el tiempo que se desea ejecutar el proceso.

Despues de tener bien escrita la función de transferencia a emular y el tiempo de ejecución se da clic en el botón ejecutar, para hacer el proceso de emulación.

Se procede a emular la función de transferencia de segundo orden mostrada en el parágrafo 6.2. que representa un motor DC sobre plataforma ARM.

$$G(s) = \frac{2.45s + 177}{s^2 + 3.163s + 23.16}$$

Usando un step de 1V y un tiempo de ejecución de 5 segundos comparar la gráfica emulada con la simulada en Matlab. Comparar el tiempo de establecimiento, el sobre pico y el tiempo pico. Además, comentar si el sistema es sobreamortiguado, subamortiguado o críticamente amortiguado y si tiene error en estado estacionario, ¿Qué magnitud tiene el error en estado estacionario?

Como paso a seguir, se cierra el lazo de Realimentación. Para esta fase de la práctica, se utilizó un arreglo de amplificadores, que tiene en su primera etapa un circuito que compara la señal de referencia con la señal de salida del emulador y luego se coloca un controlador analógico como se muestra en la figura 5-10. En este proceso el emulador actúa como planta, así que el controlador se conectará directamente a la entrada del emulador.

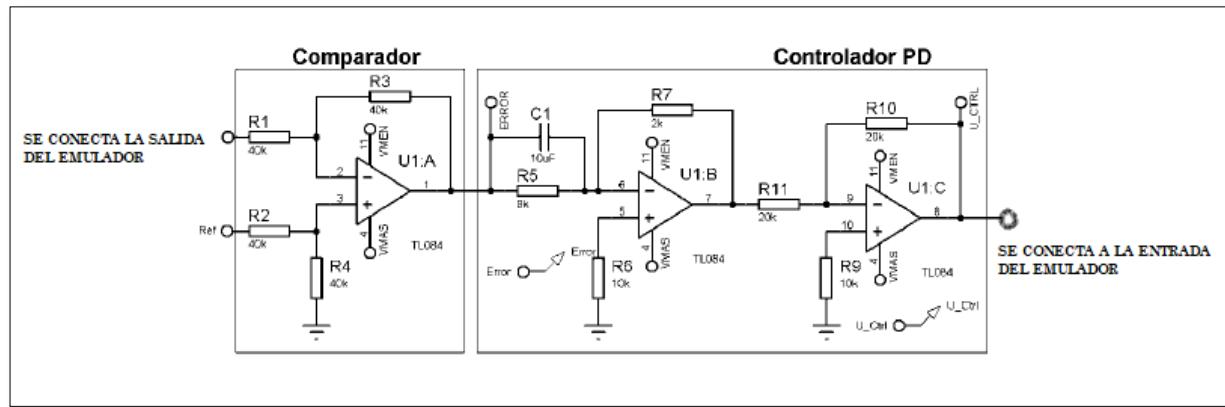


Figura 5-10. Esquema para el lazo de realimentación

Cuando ya se ha cerrado el lazo de control, se ejecuta la aplicación. Como paso final se compara la respuesta del emulador a lazo abierto con la respuesta a lazo cerrado. Mencionando las mejoras que se pueden evidenciar cuando se realimenta y se le aplica un controlador a la planta.

6. RESULTADOS

En el momento en que el prototipo cumplió con las características necesarias para su implementación, se procedió a hacer las respectivas pruebas para reafirmar su funcionalidad.

Los resultados se envían por puerto serial desde el Arduino al Matlab. Allí se procesan dentro de la función `disc`, previamente establecida en Matlab. En esta función hay una instrucción que recibe dichos datos y los grafica en tiempo real sin necesidad de una tarjeta de adquisición de datos.

Para hacer las pruebas se tomó en consideración funciones de transferencia con las que los estudiantes estuvieran familiarizados (sistemas mecánicos, sistemas electrónicos y sistemas de tanques). Se debe tener cuenta que es estrictamente necesario poseer el modelado del sistema a emular.

Cabe mencionar que hay que estudiar detenidamente las restricciones propuestas en el parágrafo 5.4. Ya dicho lo anterior, se procedió a probar el dispositivo teniendo como referencia la simulación en Matlab. Cada función de transferencia sometida al emulador fue simulada también. Con el fin, de establecer un patrón que permitiera ratificar que el emulador esté funcionando de manera adecuada. Las repuestas a cada sistema fueron mostradas de la siguiente manera: En la parte superior y de color azul se representa la simulación en Matlab y en la parte inferior de la gráfica y de color verde se representa la respuesta del emulador.

El tiempo de ejecución para cada función de transferencia se seleccionó dependiendo su tiempo de establecimiento, si su respuesta es lenta se le da un tiempo de ejecución mayor y si en su defecto la planta es rápida, el tiempo de ejecución será menor.

6.1 PLANTA DE PRIMER ORDEN.

Para representar un sistema dinámico de primer orden se hizo alusión a un tanque completamente agitado que recibe un caudal v y del recipiente se le extrae el mismo caudal. De dicho sistema se considera lo siguiente: $V = 5 \text{ m}^3$ con $v = 1 \text{ m}^3/\text{min}$, concentración en estado estacionario 1.25 mol/m^3 . Considerar un cambio en la concentración de entrada desde 1.25 mol/m^3 a 1.75 mol/m^3 ²⁸.

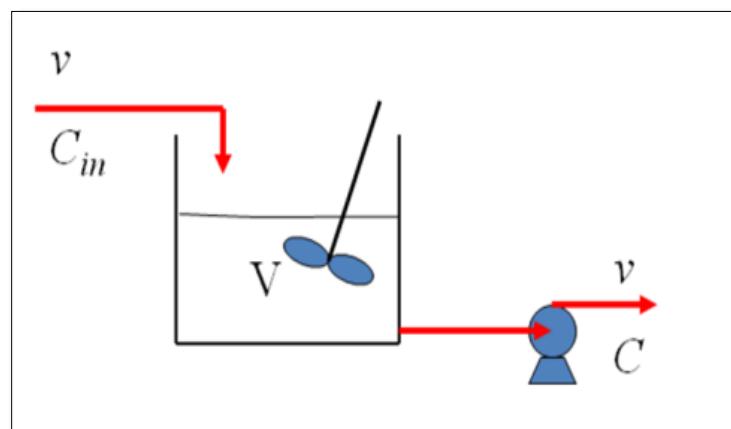


Figura 6- 1 Representación gráfica al sistema de un tanque agitado

La función de transferencia que representa dicho sistema se presenta a continuación:

$$G(S) = \frac{1}{5s+1} \quad (6.1)$$

²⁸ UNIVERSIDAD DE CARTAGENA. Dinámica y control de procesos. p.2.

En la siguiente figura se muestra la respuesta del sistema ante un escalón.

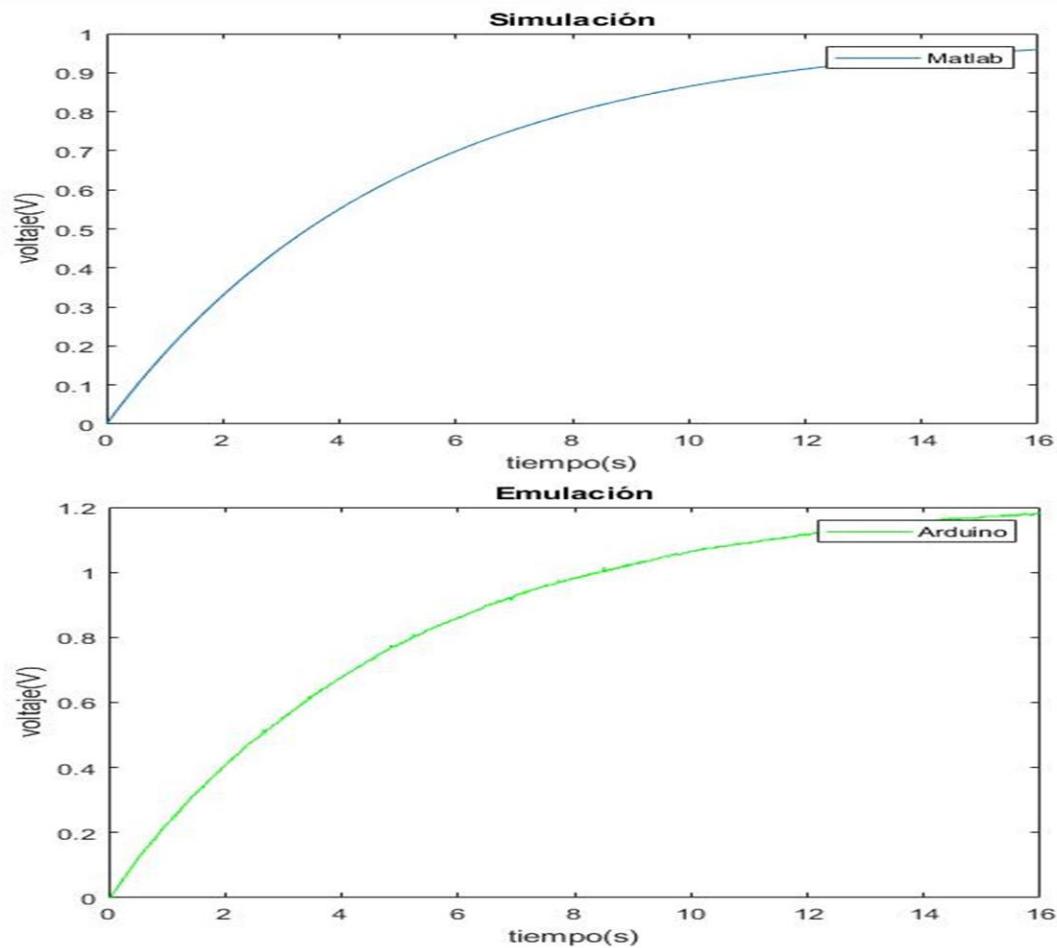


Figura 6-2 *Respuesta a un Sistema de Primer Orden (Tanque Agitado)*

La figura 6-2 representa la respuesta al sistema descrito en la ecuación 6-1 ante un escalón unitario y un tiempo de ejecución de 16 segundos. Se muestra la respuesta simulada en la parte superior de la imagen y la respuesta emulada en la parte inferior de la misma. También, se observa que las dos gráficas siguen fielmente el step ingresado. Con lo anteriormente descrito se puede inferir que, teniendo como referencia la simulación en Matlab, es posible utilizar el dispositivo para la emulación de plantas de primer orden.

6.2 PLANTAS DE SEGUNDO ORDEN

6.2.1 Motor DC sobre sobre plataforma ARM. Los datos de esta planta se obtuvieron de la velocidad del motor DC mediante la conexión serial del encoder del motor a la tarjeta ARM STM32F4-Discovery. Se tomaron 2000 muestras pseudo aleatorias, con un tiempo de muestreo de 10ms, en base al criterio de Nyquist, filtrando aquellas muestras que presentaban ruido²⁹. La función de transferencia obtenida tiene la siguiente expresión matemática.

$$G(s) = \frac{2.45s+177}{s^2+3.163s+23.16} \quad (6.2)$$

²⁹ PALACIOS, Sebastián; CHIRIBOGA, Sebastián y MONTALVO Willian. Controlador de velocidad PID-2DOF-ACO para motor DC sobre plataforma ARM. Universidad Politécnica Salesiana. 2020. p.221.

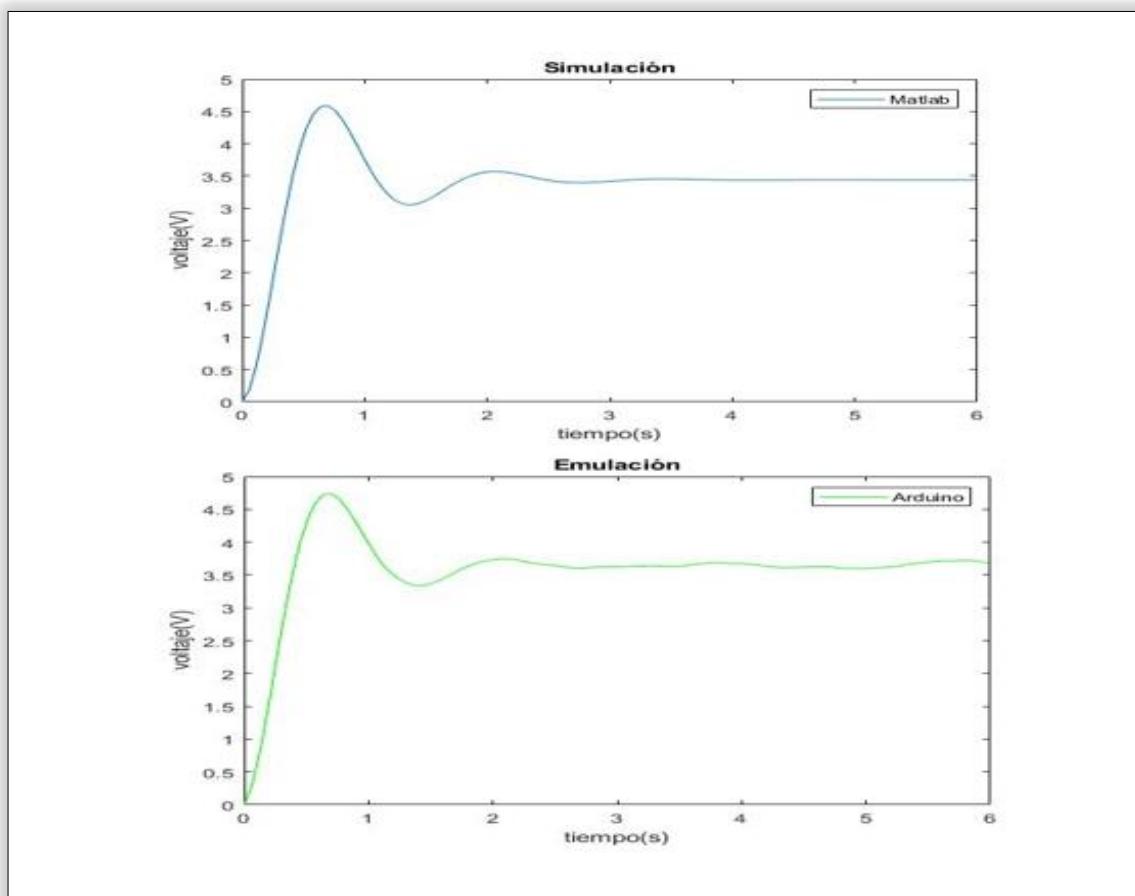


Figura 6-3 Respuesta a un Sistema de Segundo Orden (Motor DC sobre Plataforma ARM)

La figura 6-3 describe el sistema propuesto en la ecuación 6-2 ante un escalón de 0.4 Voltios y 6 segundos de tiempo de ejecución. En ambas gráficas se muestra que la respuesta al sistema es subamortiguada. En la simulación, el sistema tiene un tiempo de establecimiento de 2.7 segundos y se da cuando la amplitud es de 3.42V, mientras que el sistema emulado se establece en 3 segundos y se da cuando la amplitud es de 3.7V. Ambas gráficas tienen un sobre pico de 1.1V, la respuesta simulada tiene un error en estado estacionario de 0.58V y el sistema emulado presenta un error en estado estacionario de 0.3V.

Tomando como referencia la simulación, se puede deducir que, aunque los parámetros de salida no sean los mismos es posible emular este sistema.

6.2.2 Modelo Dinámico de un Servomotor DC. Para probar el dispositivo se usó un sistema de segundo orden, donde la función de transferencia ya tiene integrado un control proporcional de posición del servomotor DC en el lazo interno³⁰.

$$G(s) = \frac{4}{s^2 + 4s + 4} \quad (6.3)$$

³⁰ SANCHEZ, Edgar. Diseño de un sistema de control de velocidad de una turbina pelton en la central hidroeléctrica Buenos Aires Niepos usando algoritmo PI digital. Universidad Católica Santo Toribio de Mogrovejo. 2019.p.72.

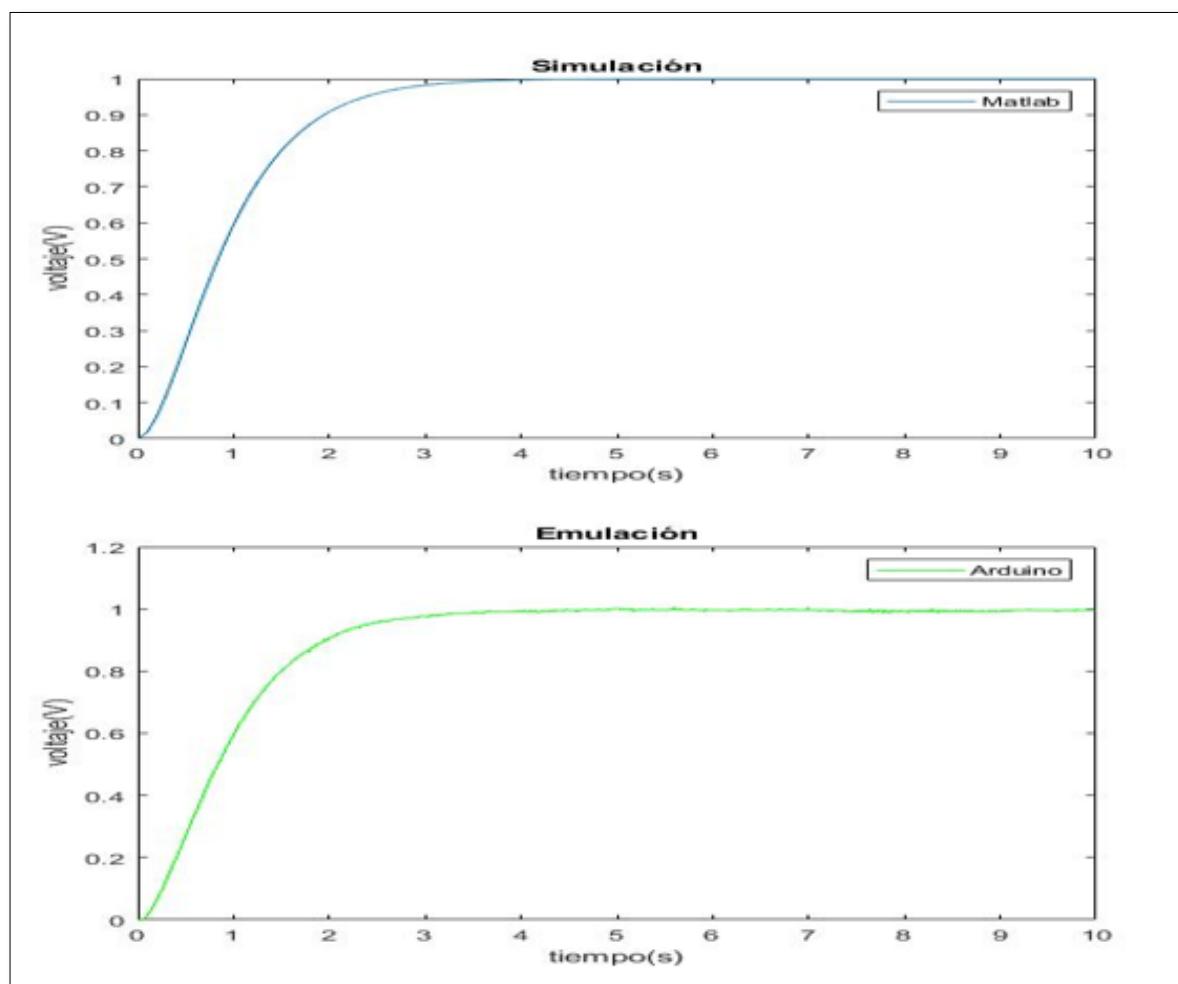


Figura 6- 4 Respuesta a un Sistema de Segundo Orden (Servomotor DC)

El sistema de segundo orden presentado en la figura 6-2 presenta un sistema sobreamortiguado, las dos gráficas siguen la referencia de un step de amplitud 1 y el tiempo de establecimiento es de 4 segundos, tanto para la gráfica simulada como para la gráfica de la emulación, de lo que se puede deducir que el emulador funciona satisfactoriamente para esta planta de segundo orden.

6.3 PLANTAS DE TERCER ORDEN

Para la emular una planta de tercer orden, se ha tomado una planta de velocidad para una transmisión hidrostática que por medio de una fuente de voltaje varía la tensión a la entrada del sistema para conocer la velocidad de salida de la transmisión hidrostática. En la medición de la velocidad de giro del motor hidráulico, se utilizó un detector de proximidad inductivo. Este detector genera un pulso por cada giro del motor para capturar los pulsos emitidos del sensor y poder observar el valor de velocidad en RPM³¹.

La función de transferencia del sistema obtenida por Burgos y Daza³² es:

$$G(s) = \frac{1.2s^2 + 20s + 13.04}{s^3 + 4.972s^2 + 22.68s + 13.01} \quad (6.4)$$

Donde $G(s)$ corresponde a la función de transferencia de la velocidad de salida con respecto a la tensión de entrada.

³¹ BURGOS, Diego; DAZA, Carlos. Control Fuzzy de velocidad para una transmisión hidrostática. Universidad de Antioquia. Colombia. 2008.p.139.

³² Ibíd, p.140.

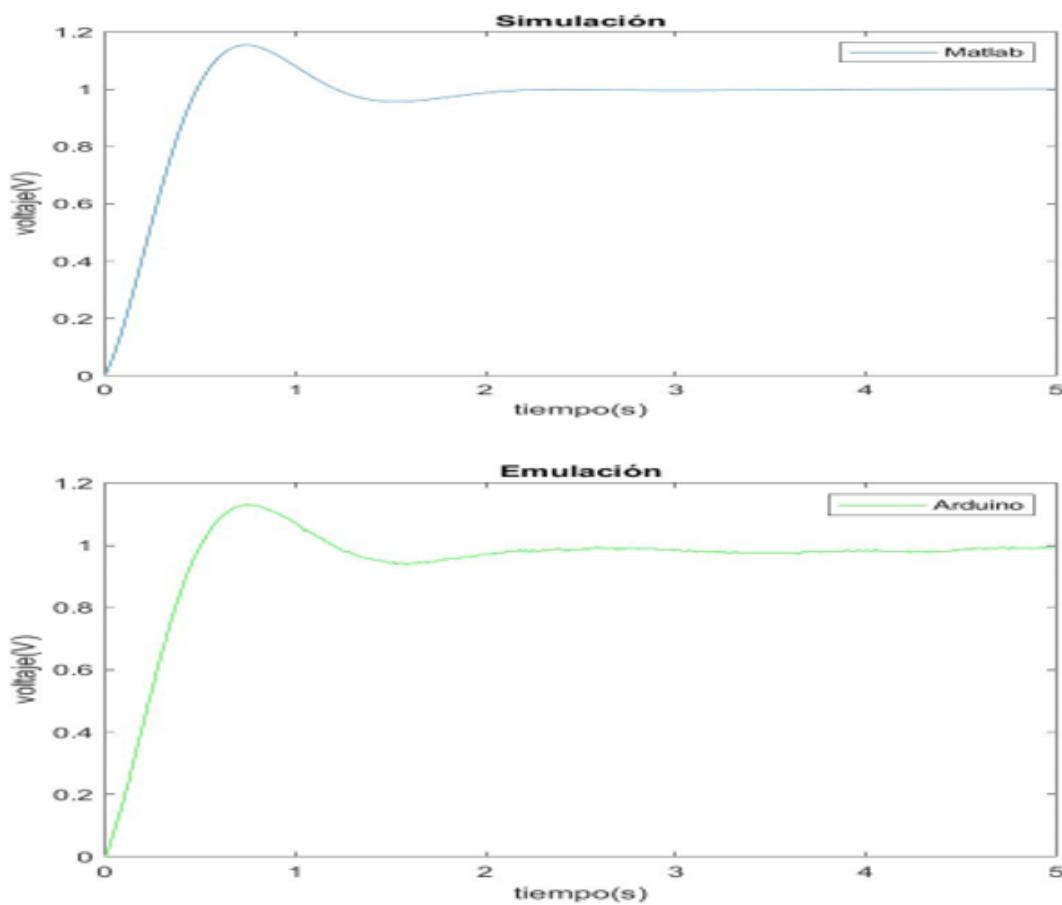


Figura 6-5 Respuesta a un Sistema de Tercer Orden (Planta de velocidad para una transmisión hidrostática)

En la figura 6-5 se presenta la respuesta simulada y emulada al sistema anteriormente descrito ante un escalón unitario y un tiempo de ejecución de 5 segundos. En las dos gráficas se observa una respuesta subamortiguada. La simulación y la emulación se establecen en 1V, el sobre pico máximo que tiene la simulación es de 0.17V, mientras que el sobre impulso máximo de la emulación es de 0.16V. La respuesta simulada se establece a los 1.4 segundos, mientras que la emulación se establece a los 1.3 segundos.

De lo anterior se puede deducir que es posible emular plantas de tercer orden con este dispositivo, teniendo como referencia el software de Matlab, dado a que la respuesta simulada es muy similar a la respuesta entregada por el emulador.

6.4 PLANTAS DE CUARTO ORDEN

Para representar la planta de cuarto orden, se usa una función de transferencia encontrada en un recurso académico perteneciente a la Facultad de Ingeniería de la Universidad Nacional Autónoma de México (UNAM)³³.

$$G(s) = \frac{100}{s^4 + 18s^3 + 87s^2 + 70s + 100} \quad (6.5)$$

³³ DEPARTAMENTO DE CONTROL, DIVISIÓN DE INGENIERÍA ELÉCTRICA, FACULTAD DE INGENIERÍA UNAM. Análisis de error en estado estacionario. México. 2006.

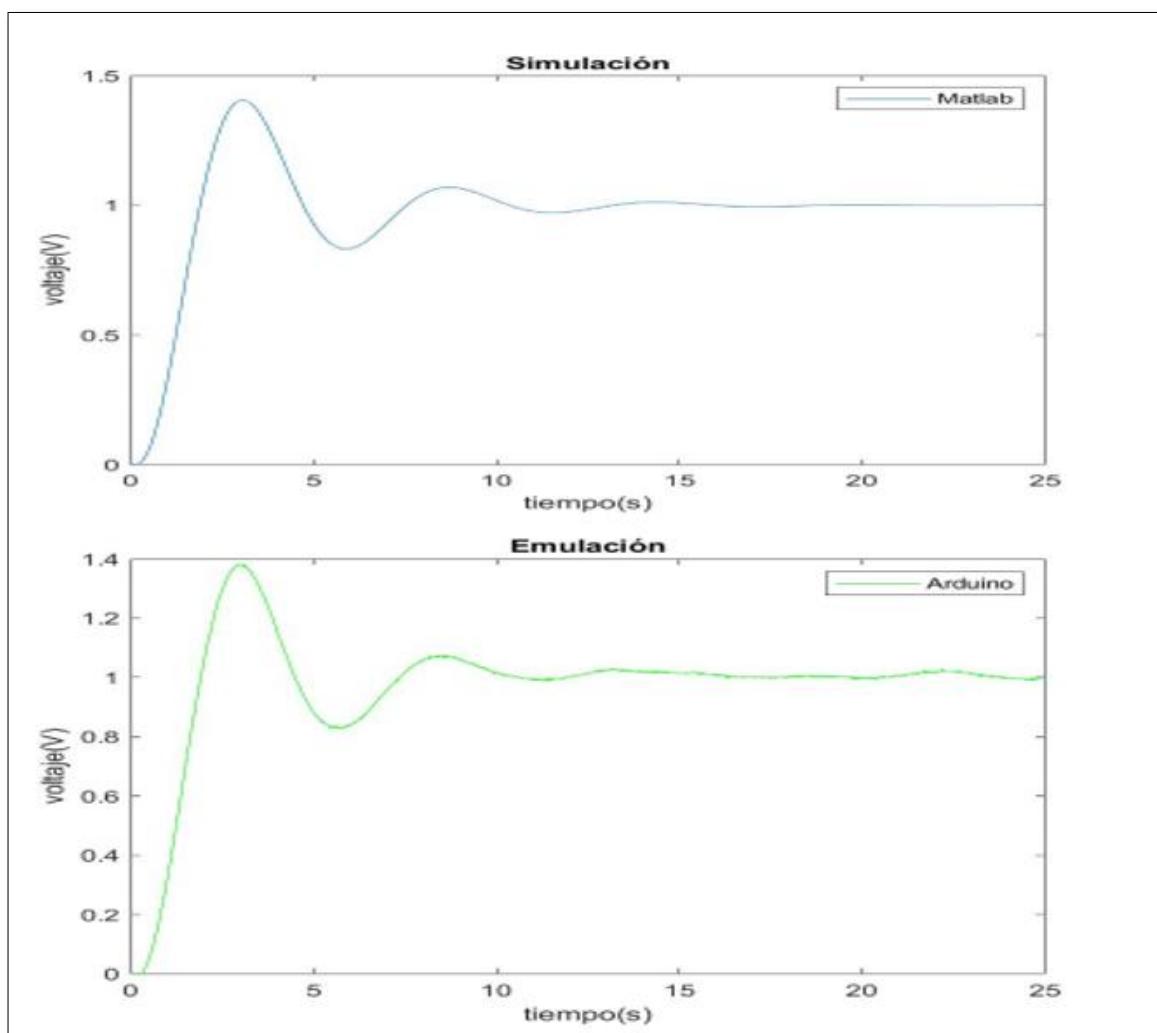


Figura 6- 6 Respuesta a un Sistema de Cuarto Orden

La figura 6-6 representa la respuesta al sistema de cuarto orden descrito en la ecuación 6.5 ante un escalón unitario y un tiempo de ejecución de 25 segundos. Las dos gráficas representadas en la figura muestran que el sistema es subamortiguado. La gráfica simulada tiene un sobre pico de 0.4V, mientras que la emulada presenta un sobre pico de 0.38V. En la simulación el sistema se establece en 12.5 segundos y se da cuando la amplitud es igual a 1V. Paralelamente la respuesta del emulador se establece en 14 segundos cuando su amplitud es 1V. Es evidente que las respuestas no son iguales; pero teniendo en cuenta que los sistemas simulados son ideales y que el emulador cuenta con dispositivos reales expuestos a diversos parámetros y diferente método de discretización, se considera posible emular dicha función de cuarto orden.

7. ANÁLISIS DE RESULTADOS

Con los resultados obtenidos se determina que es posible crear un dispositivo portátil capaz de emular funciones de transferencia lineales. La versatilidad que presenta el Arduino y su facilidad de manejo con respecto al Convertidor Digital Analógico MCP4725, hicieron que fueran los dispositivos ideales para el hardware desarrollado. La facilidad que tiene la Interfaz gráfica de Usuario en Matlab, el manejo simple del entorno de desarrollo proporcionado por Arduino y la buena disponibilidad de dichos softwares para la comunicación entre dispositivos, hicieron de estos las plataformas ideales para la creación del software necesario para el prototipo.

Es importante reiterar que el usuario debe tener conocimientos básicos en electrónica y control para el mejor aprovechamiento del dispositivo. Para ello se diseñó una práctica de laboratorio dirigida a los estudiantes de Ingeniería Electrónica de la Universidad Surcolombiana, con el fin de instruirlos en la construcción e implementación del dispositivo.

Para que fuera fácil el uso del dispositivo se creó una app con la herramienta App Designer de Matlab, que permitió incorporar toda la programación hecha en este software, lo que hace que el usuario con un ordenador y Matlab 2019a o superior, solo deba conectar el dispositivo al ordenador y escribir de manera ordenada los coeficientes de la función de transferencia que desea emular, para disfrutar de este novedoso dispositivo.

En el proceso de investigación y elaboración del presente proyecto, se encontró un trabajo de Velo³⁴ que tiene gran afinidad con lo propuesto en este. Dicho trabajo propone simular el comportamiento de plantas industriales mediante la emulación de su función de transferencia. Para ello, se tiene una pequeña base de datos con algunas funciones de transferencia predeterminadas de las que el usuario podrá seleccionar una en particular. Dicha función de transferencia está previamente discretizada y su respuesta es enviada al dispositivo DAC para su posterior lectura.

Cabe mencionar que en el proceso de diseño se pretendió utilizar algunas herramientas del trabajo en mención, pero no fue posible dado que los códigos descritos por el autor no funcionaron. De esta manera se hizo necesario iniciar de cero en este arduo proceso, que una vez culminado ha permitido evidenciar una diferenciación: la presente propuesta logró crear una aplicación mediante App Designer de Matlab, donde el usuario tiene la posibilidad de interactuar y establecer no solo unas pocas plantas a emular; sino que ofrezca la opción de inscribir muchas de estas, debido a que el software propuesto genera una ecuación en diferencia para cada planta inscrita por el usuario.

Es importante destacar que, de las plantas propuestas en el proyecto, la del tanque agitado y el servomotor DC presentadas en los párrafos 6.2.1 y 6.2.2, respectivamente, presentan un porcentaje de similitud muy alto. Esto se puede evidenciar ya que las características dinámicas presentadas en la simulación y en la emulación de dichos sistemas son prácticamente idénticas.

Es importante que el lector comprenda que el emulador es un dispositivo al cual se le ingresan variables reales como el voltaje y que su hardware está compuesto por dispositivos reales que están expuestos a interferencias del medio. Así que no es posible encontrar una similitud exacta con la simulación que es ideal. Es el caso de la función de transferencia que representa al motor DC sobre plataforma ARM en el apartado 6.2 y la planta de cuarto orden en el apartado 6.4. En la respuesta obtenida a través de estas plantas queda en evidencia que las características de la gráfica

³⁴ VELO, Esteban. Desarrollo y testeо de un emulador de plantas industriales basadas en Arduino. Universidad de Coruña. 2014.

simulada y emulada no son las mismas; pero sus respuestas transitorias tienen gran parecido y es lo que se busca para determinar si se puede o no emular.

La práctica de laboratorio se propone con el fin de que los estudiantes de Ingeniería Electrónica puedan crear su propio emulador y que, con él, tengan la oportunidad de que manipular diferentes procesos, potencializando y desarrollando aún más sus conocimientos en cuanto a análisis de procesos y control. En primera medida se les entrega el software de la aplicación diseñada en App Designer y el software requerido para la emulación en la tarjeta de desarrollo. En segunda medida, se les da las pautas necesarias para la implementación del hardware. Como tercera medida se ingresa una función de transferencia al emulador para corroborar su funcionalidad. Cuando el dispositivo esté funcionando, se compara la respuesta emulada del sistema con la respuesta simulada en Matlab. Como paso final se cierra el lazo de control de la planta (emulador) y se comparan las respuestas a lazo abierto y a lazo cerrado.

BIBLIOGRAFIA

AYALA, Tita Paola, Análisis del Método de Disparo Lineal y los Métodos de Diferencias Finitas para problemas de valores en la frontera. En: Universidad Nacional José Faustino Sánchez Carrión, Perú.2021. p.23.

BURGOS, Diego; DAZA, Carlos. Control Fuzzy de velocidad para una transmisión hidrostática. Universidad de Antioquia. Colombia. 2008.p.139-140

CATRO, Brayan. Desarrollo de un sistema portátil emulador de dinámicas de procesos de primer y segundo orden integrado con sistemas de control clásico. Revista Ingeniería e Innovación. Universidad de Pamplona, Norte de Santander, Colombia. (2021).

CHAPRA Y CANALE citado por ROVETTO, Carlos. Métodos Numéricos para Ingenieros. Universidad Tecnológica de Panamá, (junio de 2021); p.91.

CHAVEZ, Washintong. Desarrollo de aplicaciones prácticas de sistemas dinámicos en Matlab/ Simulink y Octave para la asignatura de control y automatismo. Guayaquil. Universidad Católica de Santiago de Guayaquil, 2020. p.15.

DEPARTAMENTO DE CONTROL, DIVISIÓN DE INGENIERÍA ELÉCTRICA, FACULTAD DE INGENIERÍA (UNAM). Análisis de error en estado estacionario. México.2006.

FERNÁNDEZ, Javier. Ecuaciones y Sistemas en Diferencias. Aplicaciones De La Dinámica De Poblaciones. Sevilla. Universidad de Sevilla. p.8.

HERRAMIENTAS TECNOLÓGICAS PROFESIONALES.

HUGHERS, GR citado por IZQUIERO, Luis, et al. Modelado de sistemas complejos mediante simulación basada en agentes y mediante dinámica de sistemas. En: Empiria.Vol.,16 (jul-dic,2008); 86.p.

HUIRCAN, Juan Ignacio. Conversores Análogo-Digitales y Digital-Análogo: Conceptos Básicos. En: Revista Academia. (2007) 3p.

MICROSHIP TECHNOLOGY INC. 12-Bit Digital-to-Analog Converter with EEPROM Memory in SOT-23-6.2007.

JEN-HSING, Li. Control System Laboratory with Arduino. En: IEEE, (2018); p.2

LEDESMA, Norma; CARDOZO, Berenice y TORRES Oliva. Modelo de aprendizaje para arduino uno básico. En: Revista de Cómputo Aplicado.Vol.,3 (abr-jun,2019); 16p.

MARTIN, Daprotis. Función transferencia y respuesta impulsiva. Bahía Blanca: 2011. p.2.

MELINA S,2017 citado por PARRA Irvin y HOYOS Víctor. Modelamiento matemático en el proceso de cocción del pan,2018. p.24.

MICROSHIP TECHNOLOGY INC. 12-Bit Digital-to-Analog Converter with EEPROM Memory

MONATERIO, Félix y Gutiérrez Álvaro. Introducción a los Sistemas Electrónicos de Control, 2021. p.2.

MORENO, Antonio. BUS I2C. Universidad de Córdoba. España. (2004). p.183.

NAVARRETE MOLANO, Jenny. Introducción a las ecuaciones en diferencia. Bogotá. 2003.8p. Trabajo de grado (Matemático). Fundación Universitaria Konrad Lorenz. Facultad de Matemáticas.

KATSUHIKO, Ogata. Dinámica de sistemas. Naucalpan de Juárez: Prentice-Hall Hispanoamericana S.A, 1987. p.1.

KATSUHIKO, Ogata. Ingeniería de control moderna. 5° Ed. Madrid: Pearson Educación S.A, 2010. p.3.

KATSUHIKO, Ogata, Sistemas de Control en Tiempo Discreto, segunda edición: México. Prentice Hall Hispanoamérica S.A. 1996. P. 90.

PALACIOS, Sebastián; CHIRIBOGA, Sebastián y MONTALVO William. Controlador de velocidad PID-2DOF-ACO para motor DC sobre plataforma ARM. Universidad Politécnica Salesiana. 2020. p.221.

PARRA, Irvin y HOYOS, Víctor, OB.CIT., p.24.

PÉREZ, Mario Alberto; PEREZ HIDALGO, Analía y PÉREZ BERNGUER, Elisa. Introducción a los sistemas de control y modelado matemático para sistemas lineales invariantes en el tiempo Universidad Nacional de San Juan. Facultad de Ingeniería. Departamento de Electrónica y Automática. .2008. p.2.

ROLDAN, Ángel. Emulador de GAMEBOY para dispositivos móviles. Universidad de Castilla-La Mancha. Castilla. 2017. p.14.

SANCHEZ, Edgar. Diseño de un sistema de control de velocidad de una turbina pelton en la central hidroeléctrica Buenos Aires Niepos usando algoritmo PI digital. Universidad Católica Santo Toribio de Mogrovejo. 2019.p.72.

UNIVERSIDAD DE CARTAGENA. Dinámica y control de procesos. p.2.

UNIVERSIDAD PEDAGÓGICA Y TECNOLÓGICA DE COLOMBIA. Modelamiento de Sistemas LTI. 2022.

UNIVERSIDAD POLITÉCNICA DE CATALUNYA (España), Métodos Numéricos para Ecuaciones Diferenciales Ordinarias, p.9.

VELO, Esteban. Desarrollo y testeo de un emulador de plantas industriales basadas en Arduino. Universidad de Coruña. 2014

ANEXOS

Se ha propuesto en este apartado mostrar el código en Arduino con el que funciona el software, el diagrama electrónico y las pistas del diagrama electrónico.

Anexo 1 Algoritmo en Arduino

```
#include <Wire.h>
#include <Adafruit_MCP4725.h>

Adafruit_MCP4725 dac;

float A[5];//={1,1,0,0,0};
float B[5];//={1,0,0,0,0};//[b4 b3 b2 b1 b0] se debe ingresar en ese orden
float U[5];
float Y[5];
float BT[5];
float AT[5];
float P[5][5]={{1,0,0,0,0},
               {1,-1,0,0,0},
               {1,-2,1,0,0},
               {1,-3,3,-1,0},
               {1,-4,6,-4,1}};
float DU[5];
float DY[5];
float T[5];
float t=0.01;
int n =2 ;
float in; float out; float den;
bool e=true;
float rx[12];
float ele;
int cont=0;
float muestras;
float te;
unsigned long ta=0; unsigned long tp=0; unsigned long dt=0;
uint32_t vo;

void setup() {
  Serial.begin(9600);
  dac.begin(0x60);
}

void loop() {
  if (e==true){
    leer(rx,12);
    n=rx[10];//se extrae orden del sistema
    te=rx[11];//se extrae tiempo de ejecucion
    ele=n+1;//cantidad de elementos o coef
    llenar(B,rx,0,ele);//llena coef numerador [0-4]
    llenar(A,rx,5,5+ele);//llena coef denominador [5-9]
    llenar(T,t,n);//calcula de vector T
    //Serial.println("BT= ");
    mult(B,T,BT,0,n);//multiplica punto a punto B.T
    //Serial.println("AT= ");
    mult(A,T,AT,1,n);//multiplica punto a punto A.T
    den=punto(A,T,0,n);//producto escalar A.*T
    Serial.flush();//limpia el serial
    e=false;//variable del menu a False
  }

  ta=millis();//tiempo transcurrido en milisegundos
  dt=ta-tp;//diferencia de tiempo: (tiempo actual - tiempo pasado)
  muestras=te/t;//calcula de muestra = te/0.01
  //Serial.print("muestras= ");Serial.println(muestras);

  if (cont<=muestras){//condicional para limitar la ejecucion del programa a
    la cantidad de muestras
  }
}
```

```

if (dt>=10){//condicional para que cada ejecucion se realize en el tiempo de
muestreo
    tp=ta;//actualizacion de la variable de tiempo pasado
    //Serial.print(cont);Serial.print(" ");Serial.println(dt);
    U[0]=(analogRead(A0)*5.0/1024);
    //U[0]=0.7;//lectura de la entrada
    dx(P,U,DU,0,n);//se obtiene DU que almacena las derivadas de la entrada
U para ello se emplea la matriz de pascal precargada
    in=punto(BT,DU,0,n);//se calcula in como producto punto escalar
    dx(P,Y,DY,1,n);//calcula P*Y(2:end) DY = derivadas de la salida
    out=punto(AT,DY,1,n);//calculo de out
    Y[0]=(in-out)/den;//calculo de la salida
    vo=(4096*Y[0])/5.0; //formato de salida para el dac de 12 bits
    dac.setVoltage(vo,false);
    delay(1);
    float vi = analogRead(A1)*5.0/1024;
    Serial.println(vi,5);
    //Serial.println(Y[0],5);//visualizamos la salida calculada
    //DESPLAZAMIENTOS
    desplaza(U,n);desplaza(Y,n); //desplazamos los vectores U,Y
    Y[0]=0;//reseteamos posicion 0 del vector Y
    cont++;//aumentamos contador de muestras
}
}
delay(10);//pausa para asegurar tiempo de muestreo
}

//llenar T
void llenaT(float *V,float ts,int n){
    float r=0;
    for (int i=0;i<n+1;i++){
        V[i]=pow(ts,(n-r));
        r++;
        //Serial.println(V[i],5);
    }
}
//multiplica matriz p con u y y
void dx (float M[5][5],float *X,float *DX,int ini,int fin){
    for (int i=ini;i<fin+1;i++){
        float aux=0;
        for (int j=ini;j<fin+1;j++){
            aux+=M[i][j]*X[j];
        }
        DX[i]=aux;
        //Serial.println(DX[i],5);
    }
}

void desplaza(float *V,int cant){
    float ax=V[0];
    for(int i=cant;i>=0;i--){
        V[i]=V[i-1];
    }
    V[0]=ax;
}
//multiplicar punto a punto
void mult (float *V1,float *V2,float *R,int ini,int fin){
    for (int i=ini;i<fin+1;i++){
        R[i]=V1[i]*V2[i];
        //Serial.println(R[i],5);
    }
}

//producto punto resultado un escalar

float punto (float *V1,float *V2, int ini,int fin){
    float esc=0;
    for (int i=ini;i<fin+1;i++){
        esc+=V1[i]*V2[i];
        /*Serial.print(V1[i]);Serial.print("*");Serial.print(V2[i],5);
        Serial.print("=");
        Serial.println(V1[i]*V2[i]);
        */
    }
}

```

```

    return esc;
}

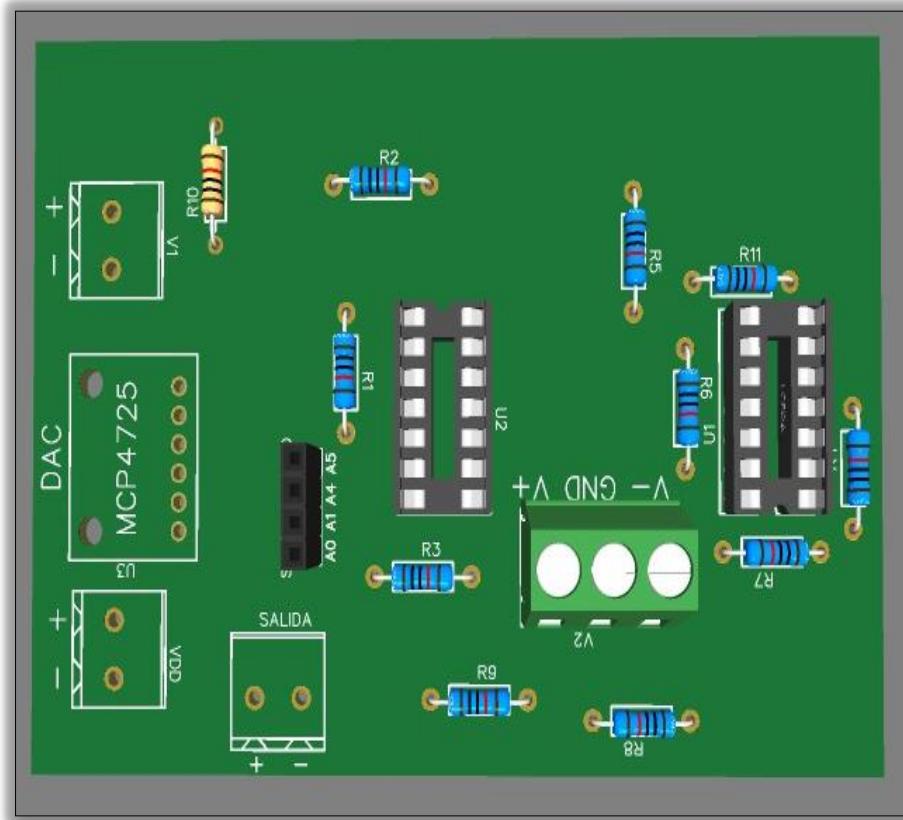
void printer(float *v, int fin, int d) {
    for (int i=0;i<fin+1;i++){
        Serial.println(v[i],d);
    }
}

void leer(float *D, int len){
    for(int i=0;i<len;i++){
        while(!Serial.available()){}
        delay(100);
        while(Serial.available()>0){
            D[i]=Serial.parseFloat();
            Serial.println(D[i]);
        }
    }
}

void llenar(float *arro, float *arri, int init, int fin){
    if (init==0){
        for (int i=init;i<fin;i++){
            arro[i]=arri[i];
            //Serial.print("llenando: ");Serial.println(arri[i]);
        }
    }
    if (init>0){
        for (int i=init;i<fin;i++){
            arro[i-init]=arri[i];
            //Serial.print("llenando: "); Serial.println(arri[i]);
        }
    }
}

```

Anexo 2 Diagrama Electrónico



Anexo 3 Pistas para el Diagrama Impreso

