

**DISEÑO E IMPLEMENTACIÓN UN SISTEMA DIDÁCTICO PARA EL
LABORATORIO DE CONTROL, QUE INTEGRE UNA INTERFAZ GRÁFICA Y
HARDWARE, PARA CONTROLAR EL GRADO DE INCLINACIÓN DE UN
BALANCÍN MOTOR HÉLICE.**

**JUAN CARLOS GUZMÁN
JUAN CARLOS GUZMAN PEREZ**

**INSTITUTO TOLIMENSE DE FORMACIÓN TÉCNICA PROFESIONAL
FACULTAD DE INGENIERÍA Y CIENCIAS AGROINDUSTRIALES
INGENIERÍA ELECTRÓNICA
ESPINAL - TOLIMA
2015**

**DISEÑO E IMPLEMENTACIÓN UN SISTEMA DIDÁCTICO PARA EL
BALANCÍN MOTOR HÉLICE.**

**JUAN CARLOS GUZMÁN
JUAN CARLOS GUZMÁN PÉREZ**

Proyecto de grado para optar por el título de Ingeniero Electrónico

**Director del proyecto
Msc en Control Ing. Oscar Bernate**

**INSTITUTO TOLIMENSE DE FORMACIÓN TÉCNICA PROFESIONAL
FACULTAD DE INGENIERÍA Y CIENCIAS AGROINDUSTRIALES
INGENIERÍA ELECTRÓNICA
ESPINAL - TOLIMA
2015**

Nota de Aceptación.

Presidente del Jurado

Ing. Libardo Cartagena _____
Jurado

Ing. Diego Parra Laguna. _____
Jurado

Espinal-Tolima (9 de noviembre, 2015)

Todo el esfuerzo recopilado en este informe se ve reflejado en el apoyo mutuo que recibimos de todos los miembros de nuestras familias, que día a día se preocuparon por brindarnos toda la colaboración asumiendo por nosotros roles propios y que debíamos dejar por dedicarnos a la redacción y elaboración de este proyecto.

AGRADECIMIENTOS

Queremos dar gracias en primer lugar a Dios por darnos la sabiduría y el entendimiento para poder culminar nuestros estudios de Ingeniería Electrónica.

A nuestro director de proyecto el Ing. Oscar Bernate por su gran apoyo y paciencia, los cuales se reflejan en la entrega de este proyecto.

A nuestros docentes en cada una de las áreas los cuales dejaron en nosotros sus conocimientos para llegar a ser personas íntegras profesionalmente y con muchos valores.

A nuestros compañeros y a todas las personas que de una u otra manera nos ayudaron durante todo el ciclo.

CONTENIDO

PAG.

| | |
|---|----|
| GLOSARIO | 14 |
| RESUMEN | 16 |
| INTRODUCCIÓN | 17 |
| 1. OBJETIVOS | 18 |
| 1.1 OBJETIVO GENERAL | 18 |
| 1.2 OBJETIVOS ESPECÍFICOS | 18 |
| 2. PLANTEAMIENTO DEL PROBLEMA | 19 |
| 2.1 DEFINICIÓN DEL PROBLEMA | 19 |
| 2.2 JUSTIFICACIÓN | 20 |
| 3. MARCO TEÓRICO | 21 |
| 3.1 ANTECEDENTES | 21 |
| 3.2 ESTADO DEL ARTE DE LOS SISTEMAS BALANCÍN MOTOR HÉLICE | 21 |
| 3.3 SISTEMA PROPUESTO. | 23 |
| 3.4 ELEMENTOS QUE CONFORMAN EL SISTEMA PROPUESTO | 24 |
| 3.4.1 Sensor. | 25 |
| 3.4.1.1 Potenciómetro Lineal. | 25 |
| 3.4.2 Actuador. | 26 |
| 3.4.2.1 Motor DC. | 26 |
| 3.4.3 Etapa De Potencia. | 27 |
| 3.4.3.1 Transistor TIP41C. | 29 |
| 3.4.4 Tarjeta De Adquisición. | 30 |
| 3.4.4.1. Tarjeta de Adquisición PTAD-S01. | 30 |
| 3.4.4.2 Placa Arduino Mega 2560. | 31 |
| 3.4.5 Interfaz Gráfica. | 33 |
| 3.4.5.1 Sistemas HMI / SCADA. | 33 |
| 3.4.5.2 LABVIEW. | 34 |
| 3.5. HERRAMIENTAS DE SOFTWARE | 36 |
| 3.5.1 MATLAB. | 36 |
| 3.5.2 SIMULINK. | 37 |
| 3.5.2.1 Capacidades del Programa Simulink. | 37 |
| 3.6 SISTEMAS | 38 |
| 3.6.1 Sistemas Dinámicos. | 38 |
| 3.6.2 Sistemas Lineales E Invariantes En El Tiempo (Lti). | 39 |

| | | |
|----------|--|----|
| 3.6.3 | Sistemas Lineales. | 39 |
| 3.6.4 | Respuesta Paso Para Sistemas Lineales. | 41 |
| 3.6.5 | Respuesta Paso A Sistemas De Primer Orden. | 41 |
| 3.6.5.1 | Constantes de Tiempo para Sistemas de Primer Orden | 42 |
| 3.6.6 | Respuesta Paso A Sistemas De Segundo Orden. | 43 |
| 3.6.7 | Especificaciones en el dominio temporal de los sistemas de segundo orden. | 45 |
| 3.7 | MODELOS | 45 |
| 3.7.1 | Clases De Modelos De Sistemas. | 46 |
| 3.7.2 | Proceso De Modelización. | 46 |
| 3.7.3 | Métodos Para Obtener Un Modelo. | 46 |
| 3.8 | CONSTRUCCIÓN DE MODELOS | 48 |
| 3.8.1 | Señales De Prueba Típicas. | 48 |
| 3.8.2 | Estabilidad Absoluta, Estabilidad Relativa Y Error En Estado Estacionario. | 49 |
| 3.9 | SEÑALES EN TIEMPO DISCRETO | 50 |
| 3.9.1 | Tiempo De Muestreo. | 50 |
| 3.9.2 | Teorema Del Muestreo. | 50 |
| 3.9.2.1 | Método Alternativo Tiempo de Muestreo Matlab. | 50 |
| 3.9.3 | Cuantificación De La Señal. | 52 |
| 3.9.3 | Transformada Z. | 53 |
| 3.9.4 | Comportamiento De Los Sistemas En Tiempo Continúo Y Discreto. | 53 |
| 3.10 | IDENTIFICACION DE SISTEMAS DINAMICOS | 54 |
| 3.10.1 | Modelos No Paramétricos Para Sistemas Lineales Invariantes En El Tiempo. | 54 |
| 3.10.2 | IDENT (Toolbox de Matlab). | 55 |
| 3.11 | SISTEMAS DE CONTROL | 55 |
| 3.11.1 | Clases De Sistemas De Control. | 56 |
| 3.11.2 | CONTROL ANÁLOGO | 57 |
| 3.11.3 | Sistemas De Control Con Controlador Digital. | 58 |
| 3.11.4 | Comparativa Entre Control Análogo Frente Al Discreto. | 59 |
| 3.12 | TIPOS DE CONTROLADORES | 59 |
| 3.12.1 | Controladores On-Off. | 59 |
| 3.12.2 | El Controlador PID. | 60 |
| 3.12.2.1 | Controlador PID discreto. | 62 |
| 3.13 | TÉCNICAS DE SINTONIZACIÓN PID | 64 |
| 3.13.1 | Métodos De Sintonización De Controladores Mediante Ziegler-Nichols (ZN). | 64 |
| 3.13.1.1 | Primer método. | 64 |
| 3.13.1.2 | Segundo método. | 65 |

| | |
|---|----|
| 3.14. FUNCIÓN DE TRASFERENCIA DEL CONTROLADOR DISCRETO | 67 |
| 3.14.1 Ecuaciones En Diferencia Algoritmo Del Controlador PID. | 67 |
| 3.14.1.1 Acción proporcional. | 68 |
| 3.14.1.2 Acción integral. | 68 |
| 3.14.1.3 Acción derivativa. | 68 |
| 3.14.1.4 Acción de control discreto. | 69 |
| 4. DESARROLLO DEL PROYECTO | 70 |
| 4.1 ACONDICIONAMIENTO DE SEÑAL | 70 |
| 4.1.1 ESCALIZACIÓN Y ACONDICIONAMIENTO DE SEÑAL | 71 |
| 4.1.2 Conversión De Voltaje A Grados. | 72 |
| 4.2 DISEÑO DEL INSTRUMENTO VIRTUAL PARA EL ACONDICIONAMIENTO DE SEÑAL Y REGISTRO DE DATOS | 73 |
| 4.3 IDENTIFICACIÓN DEL MODELO | 74 |
| 4.4 PROCESAMIENTO DE DATOS Y VALIDACIÓN DEL MODELO | 75 |
| 4.5 FUNCIÓN DE TRASFERENCIA DEL PROCESO | 76 |
| 4.6 DISEÑO DEL CONTROLADOR PID | 78 |
| 4.6.1 Sintonización Del Controlador. | 79 |
| 4.6.2 Conversión Del Controlador Continúo A Discreto. | 79 |
| 4.6.3 Simulación En Lazo Cerrado Del Modelo Obtenido. | 79 |
| 4.7 ECUACIÓN EN DIFERENCIA DE LA ACCIÓN DE CONTROL DE POSICIÓN | 81 |
| 5. DISEÑO DE LA INTERFAZ GRÁFICA PARA EL MONITOREO Y CONTROL DEL SISTEMA | 82 |
| 5.1 BLOQUE DE FUNCIONES | 82 |
| 5.2 PANEL FRONTAL | 83 |
| 6. DISEÑO DE LA INTERFAZ Y ALGORITMO DE CONTROL (SIMULINK-ARDUINO) | 85 |
| 7. RESULTADOS | 94 |
| 8. MATERIALES Y MÉTODOS | 96 |
| 8.1 MATERIALES | 96 |
| 8.2 MÉTODOS | 96 |
| 7.2.1. Método científico. | 96 |
| 8.2.2. Análisis. | 96 |
| 9. CRONOGRAMA | 97 |
| 10. CONCLUSIONES | 98 |

| | |
|--------------------------------|-----|
| 11. RECOMENDACIONES | 99 |
| 12. REFERENCIAS BIBLIOGRÁFICAS | 100 |
| 13. ANEXOS | 103 |

LISTA DE FIGURAS

PAG.

| | |
|--|----|
| Figura 1. Prototipos de un Sistema Balancín Motor Hélice..... | 22 |
| Figura 2. Modelos de Quanser con Dos y Tres Grados de Libertad | 22 |
| Figura 3. Tarjeta VTOL (Despegue y Aterrizaje Vertical) QNET..... | 23 |
| Figura 4. Prototipo del Balancín Motor Hélice..... | 24 |
| Figura 5. Representación Gráfica de los Elementos que Hacen Parte del Sistema | 24 |
| Figura 6. Representación Física y Simbólica del Potenciómetro Lineal..... | 25 |
| Figura 7. Motor DC utilizado en aeromodelismo Modelo N60-2535..... | 27 |
| Figura 8. Transistor de potencia TIP 41C | 29 |
| Figura 9. Disposición de entadas y salidas de la tarjeta de adquisición PTAD-S01. | 30 |
| Figura 10. Disposición de entadas y salidas de la placa Arduino Mega 2560. | 32 |
| Figura 11. Características de la placa Arduino mega 2560 | 33 |
| Figura 12. Sistema SCADA, Panel de Visualización..... | 34 |
| Figura 13. Ejemplo de un Panel Frontal y Bloque de Funciones de LabVIEW. | 35 |
| Figura 14. Entorno de Trabajo Matlab | 36 |
| Figura 15. Entorno de Trabajo Simulink..... | 37 |
| Figura 16. Sistema Dinámico | 39 |
| Figura 17. Ejemplo de Homogeneidad..... | 40 |
| Figura 18. Ejemplo de Aditividad | 40 |
| Figura 19. Invariabilidad en el Tiempo | 41 |
| Figura 20. a) Diagrama de Bloque de un Sistema de Primer Orden; b) Diagrama Simplificado | 42 |
| Figura 21. Respuesta Paso de un Sistema de Primer Orden | 42 |
| Figura 22. Diagrama de bloque de un sistema de segundo orden..... | 43 |
| Figura 23. Respuesta paso de un sistema de segundo orden con zeta(ζ) variable. | 44 |
| Figura 24. Ubicación de polos en un sistema de segundo orden en función de zeta (ζ)..... | 45 |
| Figura 25. Tipos de Modelos..... | 47 |
| Figura 26. Rise Time Orquilla de Valores de la Señal. | 51 |
| Figura 27. Representación del Rise Time | 52 |
| Figura 28. Cuantificación de una Señal Análoga. | 52 |
| Figura 29. Comportamiento de una Señal en el Tiempo..... | 53 |
| Figura 30. Entorno del Toolbox Ident de Matlab | 55 |
| Figura 31. Sistema de Control en Lazo Abierto. | 56 |
| Figura 32. Diagrama de Bloques de un Proceso de Control en Lazo Cerrado. | 56 |
| Figura 33. Diagrama de bloques de un sistema de control industrial..... | 57 |

| | |
|---|----|
| Figura 34. Diagrama de Bloques de un Sistema de Control Digital | 58 |
| Figura 35. Control On-Off | 60 |
| Figura 36. Representación Gráfica de las Acciones De Control | 62 |
| Figura 37. Lazo de Control de la Forma de Posición Simulink | 63 |
| Figura 38. Lazo de Control de la Forma de Velocidad Simulink | 64 |
| Figura 39. Respuesta en Forma de S. | 65 |
| Figura 40. Oscilación Sostenida. | 66 |
| Figura 41. Ganancia Crítica Kcr y el Periodo Crítico Pcr, Toolbox Siso de Matlab | 66 |
| Figura 42. Representación Gráfica del Sistema Implementado..... | 70 |
| Figura 43. Tarjeta de Adquisición PTAD-S01. | 71 |
| Figura 44. Sistema Mecánico de Visualización de Posición. | 71 |
| Figura 45. Panel Frontal y Bloque de Funciones para la Adquisición y Registro de Datos | 73 |
| Figura 46. SubVIs de la DAQ P-TDA S01 compatible con LabVIEW 2011..... ¡Error! Marcador no definido. | |
| Figura 47. Toolbox Import de Matlab | 75 |
| Figura 48. Estimación y validación del modelo con IDENT de Matlab..... | 76 |
| Figura 49. Toolbox SISO de Matlab..... | 78 |
| Figura 50. Lazo de Control y Respuesta del Sistema Simulink. | 80 |
| Figura 51. Lazo de Control y Respuesta del Sistema Simulink. | 80 |
| Figura 52. Acción de Control de Posición y Velocidad..... | 81 |
| Figura 53. Bloque de Funciones, Algoritmo de Control. LabVIEW 2011..... | 82 |
| Figura 54. Panel frontal de supervisión y control del proceso. LabVIEW 2011..... | 84 |
| Figura 55. Acondicionamiento de señal, bloques Simulink | 85 |
| Figura 56. Diagrama de bloques para el registro de datos | 86 |
| Figura 57. Datos importados al Workspace a través de bloque “Scope” de Simulink. | 86 |
| Figura 58. Ventana del Iden. Modelo estimado..... | 87 |
| Figura 59. Ventana del command Windows con la Función de transferencia del modelo | 87 |
| Figura 60. Respuesta modelo visualización de la señal de entrada y salida. | 88 |
| Figura 61. Características de la respuesta temporal del modelo | 89 |
| Figura 62. Ventana de la herramienta SISO tool de Matlab..... | 90 |
| Figura 63. Diagrama de bloques del lazo de control en Simulink | 91 |
| Figura 64. Visualización de la acción de control y la respuesta del sistema..... | 92 |
| Figura 65. Interfaz gráfica de monitoreo y control Simulink | 93 |
| Figura 66. Respuesta del sistema real..... | 94 |
| Figura 67. Comportamiento de la Acción de control y la respuesta del sistema real | 95 |

LISTA DE TABLAS

| | PAG. |
|--|------|
| Tabla 1. Características del Elemento Captador..... | 25 |
| Tabla 2. Características motor DC modelo n60-2535..... | 27 |
| Tabla 3. Características CMOS vs BIPOLAR | 28 |
| Tabla 4. Características del TIP 41 | 29 |
| Tabla 5. Funciones básicas de la tarjeta PTAD-S01..... | 31 |
| Tabla 6. Métodos de Identificación de Sistemas..... | 48 |
| Tabla 7. Característica de las acciones de control PID..... | 62 |
| Tabla 8. Valores de Sintonización ZN Método 1 | 65 |
| Tabla 9. Valores de Sintonización ZN método 2..... | 67 |
| Tabla 10. Lecturas del Sensor Tomadas Cada 5 Grados de Inclinación | 72 |
| Tabla 11. Reglas de sintonización segundo método ZN | 90 |

LISTA DE GRAFICAS

PAG.

| | |
|---|----|
| Gráfica 1. Representación Gráfica de los Datos y su Respectiva Ecuación de la Recta. | 72 |
| Gráfica 2. Gráficos en Excel Respuesta del Sistema a una Entrada Escalón | 75 |
| Gráfica 3. Respuesta del Sistema a una Entrada Escalón Unitario | 77 |
| Gráfica 4. Discretización del señal, para un $T_s=0.0257$ | 77 |

GLOSARIO

CONVERTIDOR ANALÓGICO-DIGITAL (A/D): Conocido también como codificador, convierte una señal analógica en una señal digital, usualmente una señal codificada numéricamente.

COVARIANZA: Medida del grado en que dos variables aleatorias se mueven en la misma dirección o en direcciones opuestas la una respecto a la otra.

DINÁMICA: Es la rama de la física que describe la evolución en el tiempo de un sistema físico en relación con las causas que provocan los cambios de estado físico y/o estado de movimiento.

ESTOCÁSTICO: Relativo a una variable aleatoria; algo que sigue una determinada distribución de probabilidad, usualmente con varianza finita.

FUNCIONES RACIONALES LINEALES: es una función racional cuyo numerador es un número o un polinomio de grado 1 y que tiene por denominador un polinomio de grado 1.

HEURÍSTICAS: Se define como un conjunto de reglas metodológicas no necesariamente forzosas, positivas y negativas, que sugieren o establecen cómo proceder y qué problemas evitar a la hora de generar soluciones y elaborar hipótesis.

HISTÉRESIS: se define como el momento entre dos estados transicionales que produce un retraso entre el efecto y la causa que lo produjo ese momento también se conoce como momento de basculante o de transición, el cual se da entre dos umbrales de cambio (alto y bajo el espacio entre ellos se llama zona prohibida o inactiva).

INVARIANTES: Un sistema es *invariante* con el tiempo si su comportamiento y sus características son fijas. Esto significa que los parámetros del sistema no van cambiando a través del tiempo y que por lo tanto, una misma entrada nos dará el mismo resultado en cualquier momento (ya sea ahora o después).

ITERATIVO: significa el acto de repetir un proceso con el objetivo de alcanzar una meta deseada, objetivo o resultado. Cada repetición del proceso también se le denomina una "iteración", y los resultados de una iteración se utilizan como punto de partida para la siguiente iteración.

LINEALIZAR: se refiere al proceso de encontrar la aproximación lineal a una función en un punto dado. En el estudio de los sistemas dinámicos, la linealización es un método para estudiar la estabilidad local de un punto de equilibrio de un sistema de ecuaciones diferenciales no lineales.

MUESTREADOR Y RETENEDOR (SAMPLE-AND-HOLD, S/H): Es un circuito que recibe una señal analógica y la mantiene en un valor constante durante un tiempo específico.

MUESTREO O DESRATIZACIÓN: transforma las señales de tiempo continuo en datos de tiempo discreto.

PARAMÉTRICO: El término "paramétrico" proviene de las matemáticas y se refiere al uso de parámetros o variables que permiten manipular o alterar el resultado final de una ecuación o sistema.

PERTURBACIÓN: Alteración o trastorno que se produce en el orden o en las características permanentes que conforman una cosa o en el desarrollo normal de un proceso.

PLANTA O PROCESO: Una planta es cualquier objeto físico a ser controlado.

TRANSDUCTOR: Dispositivo que convierte una señal de entrada en una señal de salida de naturaleza diferente a la de la entrada.

RESUMEN

Este documento está basado en el diseño de un sistema didáctico que a través de una interfaz gráfica desarrollada en la plataforma de Labview / Simulink, se monitorea y controla el ángulo de inclinación de la planta BMH (Balancín, Motor, Hélice). Se utilizan los bloques de funciones de la tarjeta de adquisición PTAD_S01 y la placa ARDUINO compatibles con LabVIEW y Matlab para la implantación del algoritmo de control.

Los circuitos eléctricos y electrónicos que se encuentran en este documento, están basados en los conocimientos que se adquirieron en el transcurso de la carrera de Ingeniería Electrónica que corresponden al control de variables. Se usan sensores y actuadores comerciales junto con la tarjeta de adquisición de datos (PTAD-S01-ARDUINO MEGA 2560).

Se utiliza un sensor de tipo resistivo para que proporcione una señal de voltaje baja que es leída por la tarjeta, la cual procesa y entrega otra señal, esta es enviada a través de una salida análoga de la tarjeta y acoplada a una etapa de potencia que hace girar un motor hélice.

Este tipo de control se implementa por primera vez en la institución utilizando este software. Este estudio pretende ser una base teórico práctica para posteriores trabajos de grado que profundicen y amplíen el conocimiento en sistemas de control.

PALABRAS CLAVES: Función de transferencia, respuesta temporal, sistemas dinámicos, estabilidad.

INTRODUCCIÓN

En la actualidad todas las industrias desean ser competitivas en el mercado y buscan reducir sus costos de fabricación, mientras que simultáneamente mejoran su calidad, la fiabilidad, la modernidad y la durabilidad de sus productos; para lograr estos objetivos, se requieren métodos apropiados de administración de producción y optimización, lo cual es imposible sin la aplicación del control automático y la robótica en las tecnologías del proceso. La implementación de las tecnologías se basa en el conocimiento del control automático de procesos continuos, así como de algunos de sus parámetros fundamentales, tales como: cambios de temperatura, contenidos de materiales líquidos en tanques, presión en sistemas, intensidad de luz, velocidad de rotación, etc. Por lo tanto es necesario enseñar a los estudiantes las bases de control y automatización, reglas del funcionamiento de sistemas de sensores, reguladores y sistemas operativos.

El Sistema didáctico de control propuesto permite familiarizar a los estudiantes con los conceptos básicos de la instrumentación y del control de variables. Está diseñado para controlar el Angulo de inclinación de un balancín motor hélice, aunque puede ser utilizado para el control de diferentes tipos de procesos (presión, caudal, nivel, temperatura, etc) que incluyan un sensor y un actuador análogo.

El sistema didáctico permite hacer la adquisición de los datos a través de una tarjeta ADQ (sistema embebido), para posteriormente hacer la identificación del sistema con la ayuda del Toolbox ident de Matlab.

Se podrán hacer ajustes de las ganancias de la acción del control PID (proporcional, integral y derivativo), el tiempo de muestreo durante el proceso y comparar el error en estado estacionario a través de un instrumento virtual elaborado en Labview/Simulink.

1. OBJETIVOS

1.1 OBJETIVO GENERAL

Diseñar e implementar un sistema didáctico para el laboratorio de control, que integre una interfaz gráfica y hardware, para controlar el grado de inclinación de un balancín motor hélice.

1.2 OBJETIVOS ESPECÍFICOS

- Diseñar la interfaz gráfica en LABVIEW y SIMULINK para el procesamiento de los datos y la ley de control (PID).
- Establecer la comunicación entre la PC y la Planta, Implementando la tarjeta de adquisición DAQ PTDA – S01 y la placa ARDUINO MEGA 2560.
- Diseñar y construir los circuitos necesarios para el sensor y la etapa de potencia.
- Construir y entregar prototipo (Balancín Motor hélice).
- Elaborar guías de laboratorio para el control de la planta.

2. PLANTEAMIENTO DEL PROBLEMA

2.1 DEFINICIÓN DEL PROBLEMA

El Control automático es una de las ramas de la ingeniería electrónica que se ve con mayor demanda en el área de la industria, debido a que los procesos de producción demandan aprovechamiento del tiempo y costos mínimos para llevar a cabo elaboración de sus productos.

Según estudios recientes hechos por el ICDR (Instituto Colombiano de Desarrollo Rural) la mayor demanda laboral que se da en la región del Sur y oriente del Tolima se encuentra en el área de la agroindustria, lo cual lleva a la institución a desarrollar en el profesional egresado conocimientos teórico-prácticos de competencias en dicha área.

Obedeciendo a esto el ITFIP, implanta en el perfil profesional del Ingeniero Electrónico competencias para el mejoramiento de sistemas y equipos electrónicos, especialmente en el área de control y automatización de procesos agroindustriales, pero con algunas limitaciones relacionadas con el control digital ya que a raíz de un sondeo realizado a los estudiantes de ingeniería electrónica de 9º y 10º de esta Institución, concluyó que los estudiantes tienen limitaciones en el control de variables debido a que no se tienen las herramientas necesarias para desarrollar este tipo de conocimientos y que por motivos económicos, la institución solo está centrada en modelos teóricos quedando rezagados en lo práctico ya que no cuenta laboratorios en este tipo. Partiendo de este problema nos vemos en la obligación de realizar el diseño e implementación de prototipos, que ayuden a los futuros ingenieros de nuestra institución a adquirir un mejor manejo y destreza en el control de sistemas dinámicos para la aplicación de los distintos métodos para realizar la estrategia de control adecuada.

2.2 JUSTIFICACIÓN

El diseño y la implementación de herramientas didácticas para los laboratorios de control de la institución, son fundamentales para llevar a cabo las prácticas necesarias para analizar y entender los diversos temas relacionados en el control de variables.

Cabe resaltar que las instituciones educativas cuentan con recursos limitados para la incorporación de ayudas tecnológicas que satisfagan estas necesidades.

Así mismo, en el mercado existen soluciones completas para el aprendizaje en el área de la ingeniería de control.

Con la implementación del sistema didáctico propuesto, los estudiantes de ingeniería electrónica tendrían la facilidad de interactuar con el módulo didáctico y realizar el control de variables sin la necesidad de llevar a cabo el montaje físico el cual demanda tiempo y dinero, además el estudiante estará en la capacidad de analizar el comportamiento del sistema, buscar la estrategia de control más eficaz, hacer los ajustes necesarios y cumplir con los requerimientos de diseño, esto le dará la confianza necesaria para que a futuro aplique estas técnicas y las implemente en el campo laboral.

Con estos prototipos la institución reduce costos de inversión y ayudan al estudiante de ingeniería electrónica a adquirir destreza y confianza para obtener acreditación de alta calidad.

3. MARCO TEÓRICO

3.1 ANTECEDENTES

El proyecto actual tiene como antecedente el funcionamiento de un helicóptero el cual consigue su equilibrio gracias al giro de su rotor principal, impulsando el aire desde la parte superior a la inferior de su rotor, y generando un potente chorro de aire debido al “Principio o Teorema de Bernouilli” (aplicable también a los fluidos), esto causa que la masa superior de aire, al aumentar su velocidad, disminuya su presión, creando así una succión que equilibra la aeronave. El perfil de las palas está diseñado de tal forma que el aire circula a mayor velocidad por su parte superior que por la inferior, a mayor velocidad hay menor presión (sustentación), y a menor velocidad habrá mayor presión. Una vez en el aire, tiende a girar sobre sí mismo pero en sentido contrario al giro de su rotor principal, por ello, este giro ha de ser sincronizado con el giro de su rotor secundario generando lo que se denomina “efecto anti-par”. En la mayoría de los helicópteros consiste en una doble pala situada en la cola (rotor de cola), en un plano vertical y que empuja en el mismo sentido que el giro del rotor principal, en esta práctica este efecto no será notable ya que el sistema esta fijo a una bancada que restringe la tendencia del sistema a girar respecto al eje vertical¹.

El control de posición de un balancín con una hélice y motor se ha vuelto muy popular en el campo de la electrónica de control. La idea de un balancín móvil controlado ha surgido en años recientes y ha llamado la atención de los investigadores de sistemas de control alrededor del mundo, éste se podría asemejar a un helicóptero en el cual el motor o la hélice son mecánicamente inestables, por lo que es necesario explorar las diversas formas de implementar un sistemas de control para mantener el equilibrio.

3.2 ESTADO DEL ARTE DE LOS SISTEMAS BALANCÍN MOTOR HÉLICE

Es comúnmente ver en redes sociales o educativas fabricaciones de prototipos para el control de posición de un balancín motor hélice, como proyectos de grado y/o como pasatiempos, pero muy poco enfocados a la parte didáctica la cual integre la implementación del lenguajes gráficos, sistemas embebidos y software especializado para su control y desarrollo. En la figura 1, se muestran algunos ejemplos de estos prototipos.

¹ Viltres La Rosa, Vladimir (2012).

Figura 1. Prototipos de un Sistema Balancín Motor Hélice.



Fuente: <http://blog.deinventos.com/construyendo-un-cuadricoptero-i/>

Buscando información en la red nos encontramos con una empresa canadiense llamada QUANSER dedicada al área educativa. Quanser fue fundada en 1989 en respuesta a una necesidad entre la ingeniería de educadores e investigadores de la máxima precisión de los equipos de sistemas de control robusto. Quanser contó con una plataforma modular altamente innovadora para la enseñanza y la investigación, permitiendo el control en tiempo real sobre las plataformas de PC comunes. Este producto es ahora el estándar para plataformas de experimentación de servomotor en todo el mundo².

Figura 2. Modelos de Quanser con Dos y Tres Grados de Libertad.



Fuente: www.quanser.com/products/2dof_helicopter

El sistema Helicóptero 2 DOF y El sistema del helicóptero 3 DOF de Quanser como se muestra en la figura 2, son modelo de helicóptero simplificados, ideal para introducirse en los conceptos y teorías pertinentes a las aplicaciones del mundo real de la dinámica y de control de vuelo en los helicópteros de rotor en tándem, o cualquier dispositivo con una dinámica similar de control avanzado

²²Fuente: <http://blog.deinventos.com/construyendo-un-cuadricoptero-i/>

llamados comúnmente Drones. Estos modelos en el mercado tienen un costo elevado³.

La Tarjeta VTOL (para Despegue y Aterrizaje Vertical) Quanser QNET para NI ELVIS de LabVIEW, es un helicóptero de 1 grado de libertad (1 DOF) que brinda el acceso a conceptos introductorios de control de movimiento, sistemas dinámicos, aeroespaciales, cinemática y control. El sistema consiste de un ventilador con velocidad variable y una cubierta de seguridad montada al final de un brazo dinámico. El ventilador y el ensamble del brazo giran sobre un eje sujetado a una base como lo muestra la figura 3. Tomado de : <http://sine.ni.com/nips/cds/view/p/lang/en/nid/210347>⁴

Figura 3. Tarjeta VTOL (Despegue y Aterrizaje Vertical) QNET.



Fuente: <http://www.ni.com/white-paper/52079/en/>⁵

3.3 SISTEMA PROPUESTO.

La planta como se observa en la figura 4, está formada por un tubo rectangular en aluminio como balancín o barra móvil cuya longitud es de 30 cm y un bastidor cuya altura es de 25 cm, el balancín va unido a la bancada por medio del eje cursor del potenciómetro el cual permite el movimiento de giro. En uno de los extremos del balancín hay un soporte donde va colocado el motor y la hélice.

Acoplado al eje de giro mediante unos pequeños tornillos está fijado el cursor de un potenciómetro que actúa como sensor de posición. Es aquí donde comienza todo el flujo de datos, el sensor (potenciómetro), es alimentado con la fuente de 5

³ http://www.quanser.com/products/2dof_helicopter

⁴ <http://sine.ni.com/nips/cds/view/p/lang/en/nid/210347>

⁵ <http://www.ni.com/white-paper/52079/en/>

Volts que proporciona la tarjeta de adquisición, de tal forma que a cada valor de tensión entre cursor y el común le corresponderá una posición determinada del balancín.

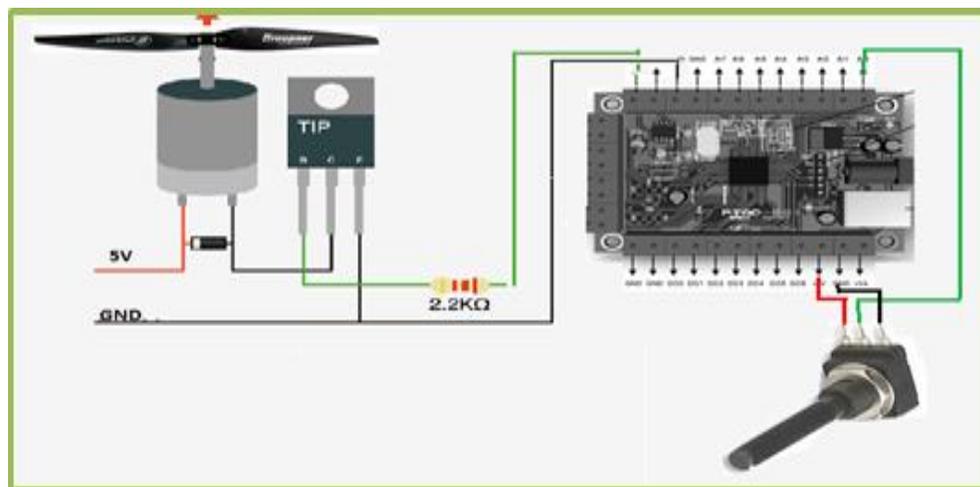
Este valor de tensión será registrada por la tarjeta DAQ mediante una de sus entradas analógicas una vez hecho el procesamiento y aplicada la acción del controlador se entrega a través de una salida análoga o PWM de la misma tarjeta ADQ la cual es acoplada a la etapa de potencia por medio de transistor TIP 41, ver figura 5.¹

Figura 4. Prototipo del Balancín Motor Hélice



Fuente: Elaboración propia.

Figura 5. Representación Gráfica de los Elementos que Hacen Parte del Sistema



Fuente: Elaboración propia.

3.4 ELEMENTOS QUE CONFORMAN EL SISTEMA PROPUESTO.

3.4.1 Sensor.

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH, etc. Una magnitud eléctrica puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad), una tensión eléctrica (como en un termopar), una corriente eléctrica (como en un fototransistor), etc⁶.

3.4.1.1 Potenciómetro Lineal.

Para registrar la posición de la barra, como ya se había comentado anteriormente, se eligió un potenciómetro de cursor giratorio ya que el movimiento del cursor se asemeja al movimiento que se quiere parametrizar, tabla 1.

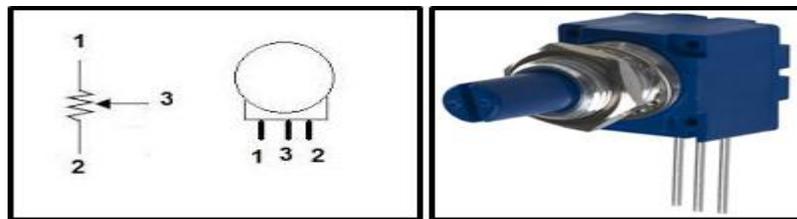
Tabla 1. Características del Elemento Captador.

| Transductor | Magnitud | Señal |
|---------------|---------------------------|-----------|
| Potenciómetro | Posición lineal y angular | Analógica |

Fuente: Elaboración propia

Se ha seleccionado un resistor variable de manera que la variación de resistencia en función de la posición del cursor sea lineal, de esta manera tendremos una ley de variación matemáticamente sencilla.

Figura 6. Representación Física y Simbólica del Potenciómetro Lineal.



⁷ Fuente: <http://preciod.com/ar/potenciometro-de-cermet-100k-lineal-2-watts-bourns-91a-104-akODS.precio>

⁶ <https://es.wikipedia.org/wiki/Sensor>

⁷ <http://preciod.com/ar/potenciometro-de-cermet-100k-lineal-2-watts-bourns-91a-104-akODS.precio>

POTENCIÓMETRO DE PRECISIÓN DE 1 VUELTAS MARCA CERMET MODELO CM47179 RESISTENCIAS 50K OHMS, LINEAL, PRECISIÓN, 2W, 5% DE TOLERANCIA, EJE METÁLICO.

Para cada posición del cursor habrá un valor diferente de resistencia (R) y por ende la tensión (V), este último parámetro será el registrado mediante la lectura de las entradas analógicas de la tarjeta de adquisición.

El potenciómetro seleccionado tiene un valor nominal de 50kΩ, si dividimos la tensión de alimentación empleada para energizar este sensor entre el valor nominal de resistencia, obtendremos la variación de la tensión leída por cada ohm que se incremente debido al movimiento del cursor, en otras palabras, la resolución, que indicará cuanto debe variar la posición de la barra para notar algún cambio en la lectura del valor indicado por el sensor, si hacemos el cálculo tenemos:

$$Res = \frac{V_{cc}}{R} = \frac{5v}{50k} = 0.1V/k\Omega$$

Para el rango de tensiones en el que se trabaja esta resolución es buena ya que ante mínimas variaciones de posición tendremos un cambio apreciable en la lectura del sensor.

3.4.2 Actuador.

Un actuador es un dispositivo capaz de transformar la energía ya sea hidráulica, neumática o eléctrica de un proceso y generar un efecto dentro del mismo proceso. El actuador recibe la orden de un regulador o controlador y activa el elemento a controlar. Como ejemplo tenemos varios tipos de actuadores:

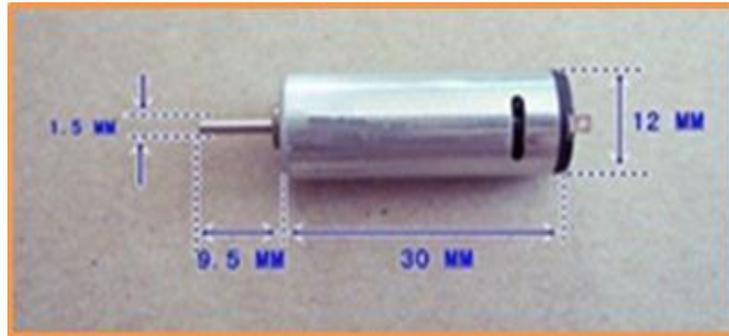
- Electrónicos
- Hidráulicos
- Neumáticos
- Eléctricos

Los actuadores hidráulicos, neumáticos y eléctricos son usados para manejar aparatos mecatrónicos.

3.4.2.1 Motor DC.

El motor de corriente continua (denominado también motor de corriente directa, motor CC o motor DC es una máquina que convierte la energía eléctrica en mecánica, provocando un movimiento rotatorio, gracias a la acción que se genera del campo magnético. Para este proyecto se optó por un motor DC genérico implementado comúnmente en aeromodelismo, ver figura 7.

Figura 7. Motor DC utilizado en aeromodelismo Modelo N60-2535.



Fuente: <http://www.aliexpress.com/popular/brushed-dc-motor.html>

Los motores de corriente continua (CC) también se utilizan en la construcción de servomotores y motores paso a paso. Además existen motores de CD sin escobillas llamados brushless utilizados en el aeromodelismo por su bajo torque y su gran velocidad.

Tabla 2. Características motor DC modelo n60-2535

| | |
|------------------------------|---|
| Modelo P/N | N60-2535 |
| Voltaje | 3V |
| Corriente | 0.3A (locked rotor current: 2.8A) |
| Diámetro | 12mm |
| Longitud | 39.5mm |
| Peso | 15g |
| Velocidad de rotación | 3V 0.3A 18500 RPM, 3.6V 0.31A 22000RPM y 4.5V 0.35A 24000 RPM |

Fuente: <http://www.uv.es/marinjl/lectro/transistores.html>⁸

Es posible controlar la velocidad y el par de estos motores utilizando técnicas de control de motores CD. En la tabla 2, podemos apreciar las características del motor implementado.

3.4.3 Etapa De Potencia.

El funcionamiento y utilización de los transistores de potencia es idéntico al de los transistores normales, teniendo como características especiales las altas

⁸ <http://www.uv.es/marinjl/electro/transistores.html>

tensiones e intensidades que tienen que soportar y, por tanto, las altas potencias a disipar, ver tabla 3.

Existen tres tipos de transistores de potencia:

- bipolar.
- unipolar o FET (Transistor de Efecto de Campo).
- IGBT.

Tabla 3. Características CMOS vs BIPOLAR.

| Parámetros | MOS | Bipolar |
|---------------------------------|--------------------|--------------------|
| Impedancia de entrada | Alta (1010 ohmios) | Media (104 ohmios) |
| Ganancia en corriente | Alta (107) | Media (10-100) |
| Resistencia ON (saturación) | Media / alta | Baja |
| Resistencia OFF (corte) | Alta | Alta |
| Voltaje aplicable | Alto (1000 V) | Alto (1200 V) |
| Máxima temperatura de operación | Alta (200°C) | Media (150°C) |
| Frecuencia de trabajo | Alta (100-500 Khz) | Baja (10-80 Khz) |
| Costo | Alto | Medio |

Fuente: <http://www.uv.es/marinjl/electro/transistores.html>

El IGBT ofrece a los usuarios las ventajas de entrada CMOS, más la capacidad de carga en corriente de los transistores Bipolares:

- Trabaja con tensión.
- Tiempos de conmutación bajos.
- Disipación mucho mayor (como los bipolares).

Nos interesa que el transistor se parezca, lo más posible, a un elemento ideal:

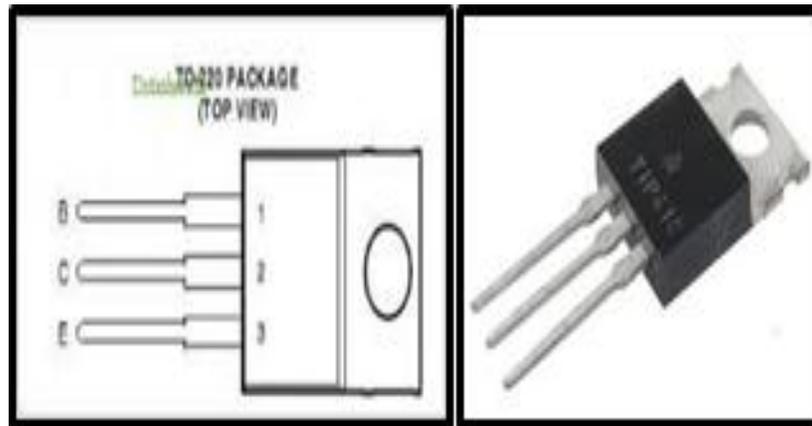
- Pequeñas fugas.
- Alta potencia.
- Bajos tiempos de respuesta (ton, toff), para conseguir una alta frecuencia de funcionamiento.
- Alta concentración de intensidad por unidad de superficie del semiconductor.
- Que el efecto avalancha se produzca a un valor elevado (VCE máxima elevada).
- Que no se produzcan puntos calientes (grandes di/dt).

Una limitación importante de todos los dispositivos de potencia y concretamente de los transistores bipolares, es que el paso de bloqueo a conducción y viceversa no se hace instantáneamente, sino que siempre hay un retardo (ton, toff). Las causas fundamentales de estos retardos son las capacidades asociadas a las uniones colector - base y base - emisor y los tiempos de difusión y recombinación de los portadores⁹.

3.4.3.1 Transistor TIP41C.

Es un transistor bipolar NPN de potencia para frecuencias bajas fabricado de silicio, con excelentes características para ser utilizado en conmutación, lo podemos encontrar en diversas versiones cuyo sufijo indica la tensión máxima entre colector emisor, su par complementario es el transistor PNP TIP42C.

Figura 8. Transistor de potencia TIP 41C



Fuente: Electrónica sistemas digitales.

En la figura 8, se evidencia el tipo de encapsulado y la disposición de los pines para su respectiva configuración. En la tabla 4, se dan las características del transistor

Tabla 4. **Características del TIP 41** ¹⁰

⁹ <http://www.uv.es/marinjl/electro/transistores.html>

¹⁰ <http://spanish.alibaba.com/product-gs/transistor-tip41-363414024.html>

| | |
|---|--|
| Datasheets | TIP41/AB/C |
| Product Photos | TO-220-3 Pkg |
| Product Change Notification | Lead Frame Dimensions Change 29/Nov/2007 |
| Catalog Drawings | Transistor TO-220(AB) |
| Standard Package | 200 |
| Category | Discrete Semiconductor Products |
| Family | Transistors (BJT) - Single |
| Series | - |
| Transistor Type | NPN |
| Current - Collector (Ic) (Max) | 6A |
| Voltage - Collector Emitter Breakdown (Max) | 100V |
| Vce Saturation (Max) @ Ib, Ic | 1.5V @ 600mA, 6A |
| Current - Collector Cutoff (Max) | 700µA |
| DC Current Gain (hFE) (Min) @ Ic, Vce | 15 @ 3A, 4V |
| Power - Max | 2W |
| Frequency - Transition | 3MHz |
| Mounting Type | Through Hole |

Fuente: <http://spanish.alibaba.com/product-gs/transistor-tip41-363414024.html>

3.4.4 Tarjeta De Adquisición.

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

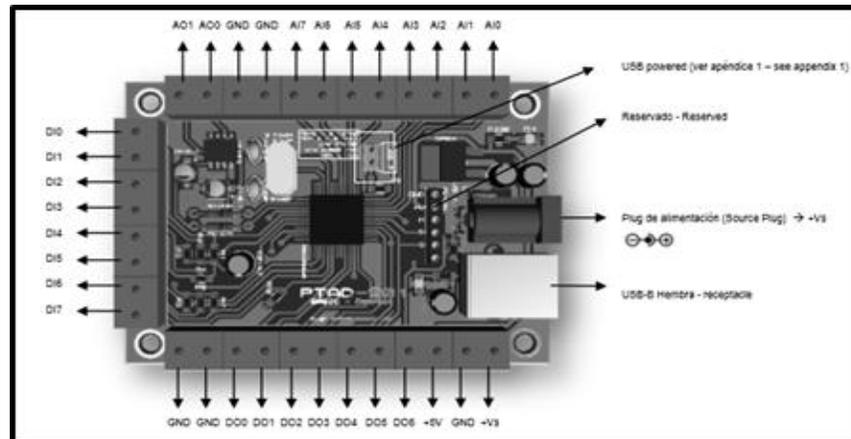
Estos dispositivos en formato tarjeta de circuito impreso se conectan al ordenador a través de las ranuras de expansión o cable de comunicación serial (USB)¹¹.

3.4.4.1. Tarjeta de Adquisición PTAD-S01.

La tarjeta PTAD-S01 Basic figura 8, es un sistema USB de adquisición de datos y control para PC orientado hacia la robótica y el control de procesos, el cual permite digitalizar señales en un rango de 0 a 5 voltios. También cuenta con canales digitales de entrada y salida TTL que le permite controlar eventos externos como por ejemplo actuadores, control de relees y otros sistemas digitales.

Figura 9. Disposición de entadas y salidas de la tarjeta de adquisición PTAD-S01.

¹¹ https://es.wikipedia.org/wiki/Adquisici%C3%B3n_de_datos



Fuente: Manual Técnico P-TAD SO1 Basic USB DAQ Multifunción. Web Site:
<http://181.143.31.26/pub/internet/nuevo/ptads01.pdf>

El sistema PTAD-S01 tiene integrado un procesador propio que administra todas sus funciones en forma independiente al PC, ventaja que permite realizar adquisiciones con el sistema sin cargar el equipo de cómputo. Su fácil funcionamiento y sus librerías para su puesta en operación en LabVIEW, MATLAB y Simulink la convierten en una poderosa herramienta para los fines didácticos e investigativos. En la tabla 5, se muestra las funciones básicas de la tarjeta de adquisición.

Tabla 5. Funciones básicas de la tarjeta PTAD-S01

| | |
|--|------------------------------------|
| Rango de voltaje de operación – Operating Voltage Range (V) | 9 – 12 |
| Corriente de alimentación Min – Min Operating Current Range (mA) | 200 |
| Voltaje de salida (E/S) – Output Voltage Range I/O (V) | 0 - 5 |
| Corriente de salida (E/S) – Output Current Range I/O (mA) | 25 mA |
| USB (Velocidad, Cumplimiento, CDC) - (speed, compliance, CDC) | Full Speed 12Mb/s, USB 2.0, Serial |
| Temperatura de operación - Temperature Range (C) | -40 to 85 |
| Rendimiento del CPU - CPU Core Performance | 48 MHz (12 MIPS) |
| Canales análogos de entrada – Analog inputs channels | 8 (AI0-AI7) |
| Canales análogos de salida – Analog outputs channels | 2 (AI0, AI1) |
| Canales digitales de entrada – Digital inputs channels | 8 (DI0-DI7) |
| Canales digitales de salida - Digital outputs channels | 7 (DO0-DO6) |
| Resolución de entrada - Input Resolution (bits) | 10 |
| Máxima velocidad de muestreo - Max Sampling Rate (kS/s) | 25 |
| Tipo de medida | Voltaje (V) |
| Software objetivo (Windows) | LabVIEW®, MATLAB® & Simulink® |
| Sistema operativo compatible – Compatible operative system | Windows, Linux, MAC |

Fuente: Manual Técnico P-TAD SO1 Basic USB DAQ Multifunción. Web Site:
<http://181.143.31.26/pub/internet/nuevo/ptads01.pdf>

3.4.4.2 Placa Arduino Mega 2560.

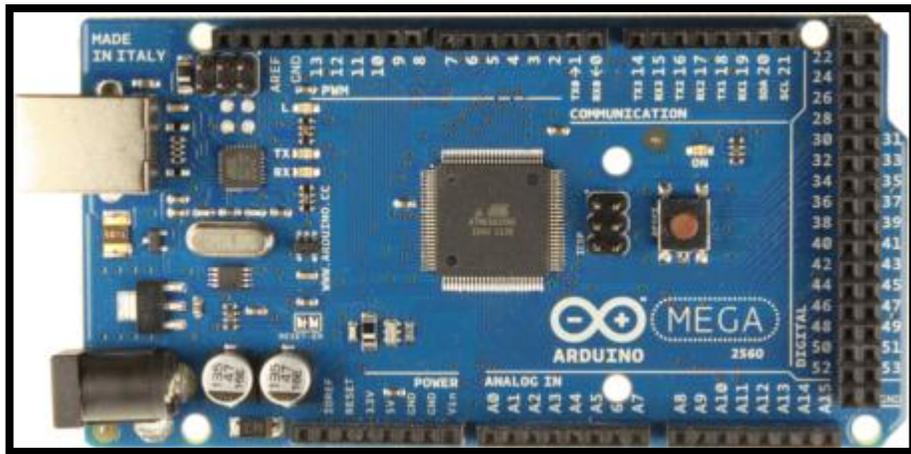
Arduino es una marca de micros controladores mundialmente conocida por los amantes de la electrónica, la programación y la robótica. Es un proyecto Open

Source que pone a disposición de sus usuarios una amplia gama de dispositivos basados en el micro controlador AtMega.

El Arduino Mega 2560 es considerado probablemente el micro controlador más capaz de la familia Arduino. Posee 54 pines digitales que funcionan como entrada/salida; 16 entradas análogas, un cristal oscilador de 16 MHz, una conexión USB, un botón de reset y una entrada para la alimentación de la placa.

La comunicación entre la computadora y Arduino se produce a través del puerto serie, sin embargo posee un convertidor USB-serie, por lo que sólo se necesita conectar el dispositivo a la computadora utilizando un cable USB como el que utilizan las impresoras.

Figura 10. Disposición de entadas y salidas de la placa Arduino Mega 2560.



Fuente: <https://www.arduino.cc/en/Main/arduinoBoardMega>¹²

Arduino es compatible con LabVIEW a través del toolkit LIFA que se puede descargar directamente de la página de NATIONAL INSTRUMENT y que permite a los usuarios de Arduino adquirir datos desde el microcontrolador y procesarlos en el entorno de programación grafica de LabVIEW.

La placa Arduino permite compatibilidad con Simulink a través de bloques funcionales que se pueden descargar de la página de Matwork esto le permite

¹² <https://www.arduino.cc/en/Main/arduinoBoardMega>

monitorizar y ajustar los algoritmos que se ejecutan en tiempo real sin afectar el funcionamiento de proceso.

Figura 11. Características de la placa Arduino mega 2560

| Característica | Descripción |
|----------------------------------|------------------|
| Microcontrolador | ATmega 2560 |
| Voltaje de Operación | 5V |
| Tensión de Entrada (Recomendada) | 7-12V |
| Tensión de Entrada (límite) | 6-20V |
| Pines Digitales de E/S | 54 (15 para PWM) |
| Pines de Entrada Analógicos | 16 |
| Corriente DC por pin E/S | 40mA |
| Corriente DC para pin 3.3V | 50mA |
| Memoria Flash | 256 KB |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Frecuencia de Reloj | 16 MHz |

3.4.5 Interfaz Gráfica.

3.4.5.1 Sistemas HMI / SCADA.

El funcionamiento de muchas operaciones industriales depende de procesos remotos. Los Sistemas de Telecontrol y Supervisión HMI / SCADA permiten al cliente rápidamente conocer e identificar cualquier problema potencial, así como administrar y manejar maquinaria y procesos remotos. El cliente puede conocer en todo momento el estado de una instalación centralizando toda la información de los emplazamientos remotos en uno o varios Puestos de Control. Los equipos de control situados en las estaciones analizan los parámetros requeridos recogiendo los valores aportados por los diferentes tipos de sensores. Cuando se identifica una situación especial o de alerta estos equipos realizan la actuación pertinente y advienen del mismo al Puesto de Control. Desde el Puesto Central se procesa la información y se genera de forma automática el procedimiento de actuación pertinente. De igual forma, desde el Puesto Central se puede obtener, en tiempo real, cualquier información relativa a las estaciones remotas, ver figura 11.

Figura 12. Sistema SCADA, Panel de Visualización.



Fuente:<http://www.automatik solutions.com/servicios/soluciones-en-software/sistema-de-supervision-scada>

Además de gestionar alarmas y de capturar datos, los Sistemas SCADA permiten generar planes de mantenimiento y eficaces procedimientos de actuación para el personal. Dichos planes facilitan el trabajo de mantenimiento permitiendo automatizar procesos hasta niveles insospechados por el propio cliente¹³.

Los Sistemas SCADA permiten, entre otros la gestión y control remoto de:

- Sistemas y Equipos Industriales: agua, gas, electricidad, etc.
- Instalaciones Petroleras, Refinerías, Oleoductos, etc.
- Instalaciones Remotas de Telecomunicaciones.
- Telecontrol de Procesos Hidráulicos.
- Mediciones Ambientales.
- Sistemas de Supervisión y Control.

3.4.5.2 LABVIEW.

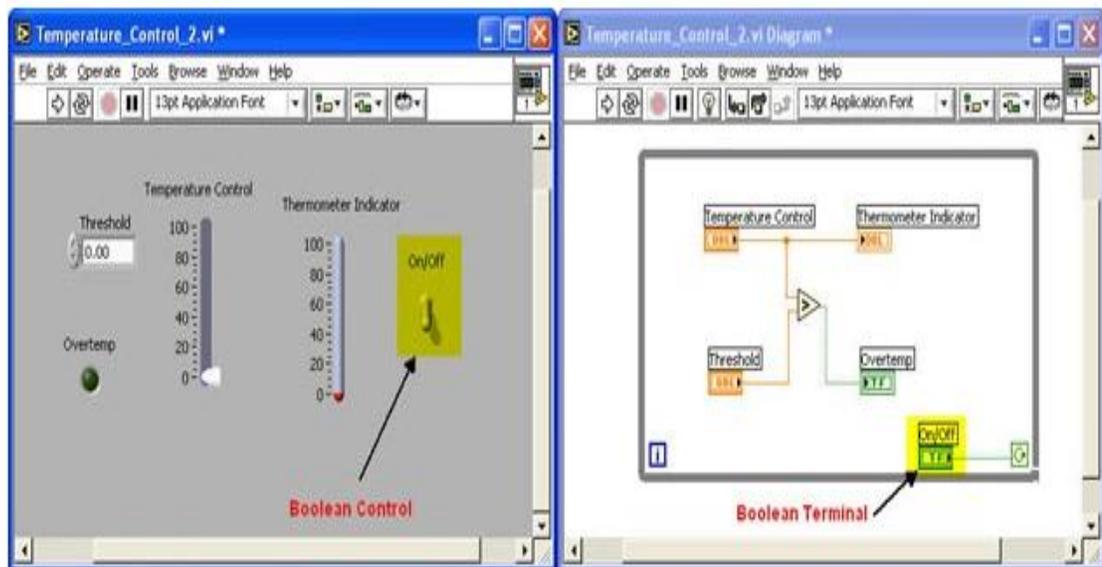
LabVIEW (acrónimo de Laboratory Virtual Instrumentation Engineering Workbench) es una plataforma y entorno de desarrollo para diseñar sistemas, con un lenguaje de programación visual gráfico. Recomendado para sistemas hardware y software de pruebas, control y diseño, simulado o real y embebido,

¹³ Telecomunicaciones Prodetel (2015). Sistemas HMI / SCADA. Web Site: <https://prodetel.net/sections/scada.html>

pues acelera la productividad. El lenguaje que usa se llama lenguaje G, donde la G simboliza que es lenguaje Gráfico.

Los programas desarrollados con LabVIEW se llaman Instrumentos Virtuales, o Vis, y su origen provenía del control de instrumentos, aunque hoy en día se ha expandido ampliamente no sólo al control de todo tipo de electrónica (Instrumentación electrónica) sino también a su programación embebida, comunicaciones, matemáticas, etc. Un lema tradicional de LabVIEW es: “La potencia está en el Software”, que con la aparición de los sistemas multinúcleo se ha hecho aún más potente. Entre sus objetivos están el reducir el tiempo de desarrollo de aplicaciones de todo tipo (no sólo en ámbitos de Pruebas, Control y Diseño) y el permitir la entrada a la informática a profesionales de cualquier otro campo. LabVIEW consigue combinarse con todo tipo de software y hardware, tanto del propio fabricante (tarjetas de adquisición de datos, PAC, Visión, instrumentos y otro Hardware) como de otros fabricantes, figura 12.

Figura 13. Ejemplo de un Panel Frontal y Bloque de Funciones de LabVIEW.



Fuente: <http://attila.sdsu.edu/me205/modules/1labview/03loops/2tutorial.html>

LabVIEW es un lenguaje y a la vez un entorno de programación gráfico en el que se pueden crear aplicaciones de una forma rápida y sencilla. Labview es una herramienta de programación gráfica para pruebas, control y diseño mediante la programación de bloques. El lenguaje que usa se le llama lenguaje G, donde la G simboliza que es lenguaje Grafico. Este programa fue creado por National Instruments (1976) para funcionar sobre máquinas MAC, salió al mercado por primera vez en 1986. Ahora está disponible para las plataformas Windows, UNIX,

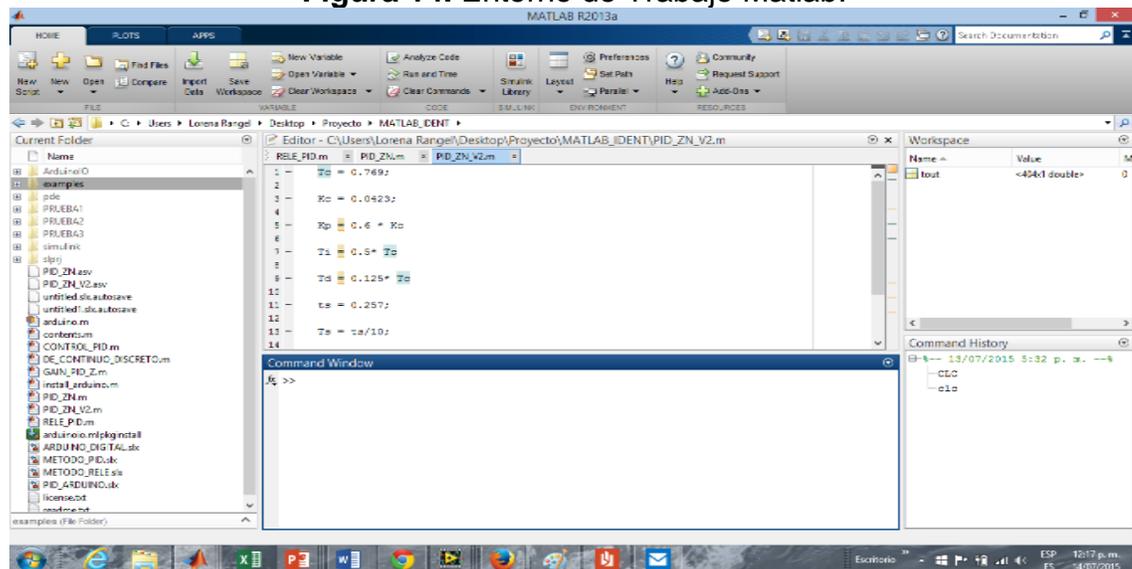
MAC y Linux. La versión actual 2014, publicada en abril de 2014 cuenta con soporte para Windows Eight (Win 8). [6]

3.5. HERRAMIENTAS DE SOFTWARE.

3.5.1 MATLAB.

MATLAB (abreviatura de *MATrix LABORatory*, “laboratorio de matrices”) es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux¹⁴. figura 13

Figura 14. Entorno de Trabajo Matlab.



Fuente: Elaboración propia.

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multi-dominio) y GUIDE (editor de interfaces de usuario – GUI). Además, se pueden ampliar las capacidades de MATLAB con las *cajas de*

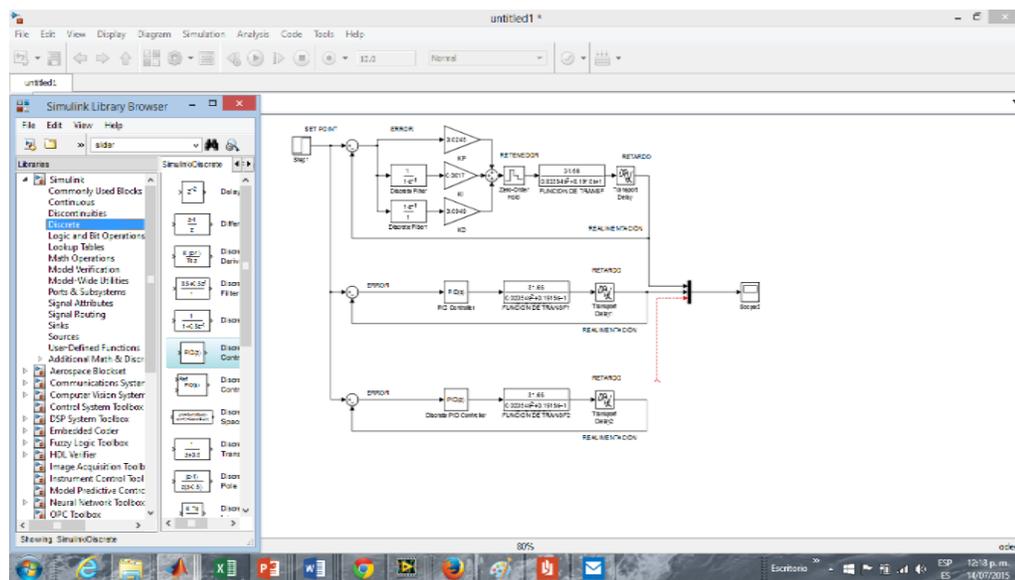
¹⁴ http://www.isa.cie.uva.es/~felipe/docencia/ra12itielec/tema1_trasp.pdf

herramientas (toolboxes); y las de Simulink con los paquetes de bloques (blocksets)¹⁵.

3.5.2 SIMULINK.

Simulink® es un entorno de diagrama de bloques para multi simulación de dominio y diseño basado en modelos. Es compatible con la simulación, generación automática de código, y la prueba continua y verificación de sistemas embebidos. Simulink ofrece un editor gráfico, bibliotecas de bloques personalizables y solucionadores para el modelado y simulación de sistemas dinámicos. Está integrado con MATLAB®, lo que le permite incorporar algoritmos de MATLAB en modelos y resultados de la simulación de exportación a MATLAB para su posterior análisis¹⁶. Figura 14.

Figura 15. Entorno de Trabajo Simulink.



Fuente: Elaboración propia.

3.5.2.1 Capacidades del Programa Simulink.

Construcción del modelo - subsistemas jerárquicos Modelo con los bloques de la biblioteca predefinidos.

¹⁵ http://www.isa.cie.uva.es/~felipe/docencia/ra12itielec/tema1_trasp.pdf

¹⁶ Ghidella, J. (1994).

Simulación del Modelo - Simular el comportamiento dinámico de su sistema y ver los resultados como las corridas de simulación.

Analizar los resultados de simulación - Ver resultados de la simulación y depurar la simulación.

Gestor de Proyectos - Administre fácilmente archivos, componentes, y grandes cantidades de datos para su proyecto.

Conexión a Hardware – Conexión del modelo de hardware para las pruebas en tiempo real y el despliegue de sistemas embebidos¹⁷.

3.6 SISTEMAS.

Un sistema es una combinación de componentes que actúan conjuntamente para alcanzar un objetivo específico. Una componente es una unidad particular en su función en un sistema. De ninguna manera limitado a los sistemas físicos, el concepto de sistema se puede ampliar a fenómenos dinámicos abstractos, tales como los que se encuentran en la economía, el transporte, el crecimiento de la población y la biología.

Un sistema puede formar parte de otro sistema de mayor nivel, en cuyo caso al primero se le denomina subsistema. Los diferentes subsistemas intercambian por lo general información, materia o energía para lograr algún objetivo. Los términos señales de entrada o de salida se utilizan entonces para abstraer ese flujo de información, materia o energía en el concepto matemático de funciones.

El sistema entonces puede interpretarse como un conjunto de subsistemas que logran transformar una señal en otra. Estos dispositivos pueden ser entes físicos, como un circuito electrónico, o virtuales, como algoritmos implementados en software¹⁸.

3.6.1 Sistemas Dinámicos.

Un sistema se llama dinámico si su salida en el presente depende de una entrada en el pasado; si su salida en curso depende solamente de la entrada en curso, el sistema se conoce como estático. La salida de un sistema estático permanece constante si la entrada no cambia y cambia sólo cuando la entrada cambia. En un sistema dinámico la salida cambia con el tiempo cuando no está en su estado de equilibrio. *Tomado de: Dinámica de sistemas Ogata Prentice – Hall*¹⁹

Un sistema dinámico se entiende como un proceso en el cual interactúan variables de diferentes clases. Se tienen entradas $u(t)$, salidas $y(t)$ y entradas de

¹⁷ Ghidella, J. (1994).

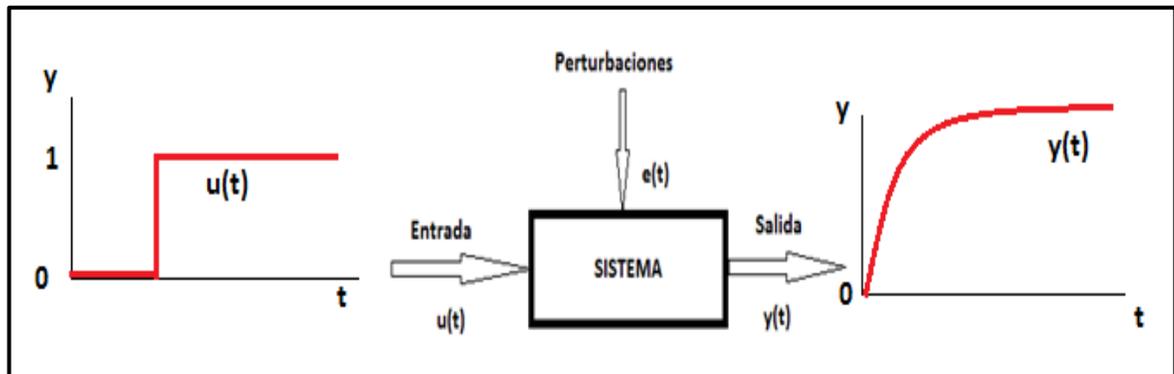
¹⁸ Alvarado Moya, P. (2008).

¹⁹ Ogata *Prentice – Hall* (pag . 2004)

perturbaciones $e(t)$ que son señales externas no deseadas, pero que en todo proceso están presentes.

Un sistema dinámico puede ser definido como un proceso en el cual las entradas $u(t)$ pueden ser controladas por el observador, mientras que las perturbaciones $e(t)$, no lo son. La señal de salida $y(t)$, son variables y suministran información sobre el sistema, figura 15.

Figura 16. Sistema Dinámico.



Fuente: Elaboración propia.

3.6.2 Sistemas Lineales E Invariantes En El Tiempo (Lti).

En procesamiento de señales, un **sistema LTI** (Linear Time-Invariant) o sistema lineal e invariante en el tiempo, es aquel que, como su propio nombre indica, cumple las propiedades de linealidad e invariancia en el tiempo.

Estos sistemas son la clase más importante de los sistemas dinámicos y representan idealmente los procesos encontrados en la vida real²⁰.

3.6.3 Sistemas Lineales.

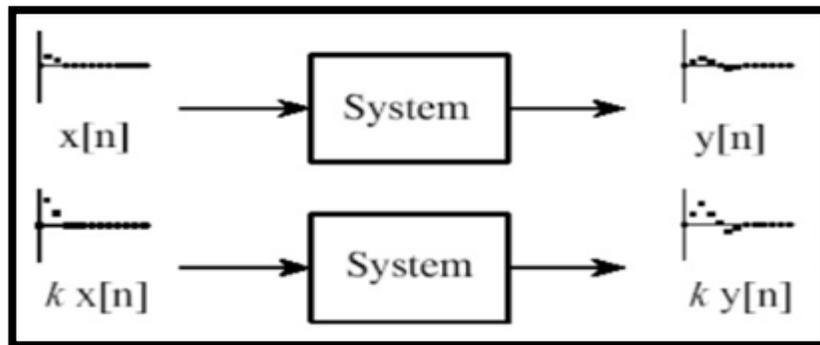
Para que un sistema se considere lineal debe cumplir con tres requerimientos como son:

Homogeneidad: Decimos que un sistema es homogéneo cuando un cambio en la amplitud de la señal de entrada produce una variación proporcional en la señal de salida, figura 16.

**Si, una señal de entrada $x[n]$ produce una salida $y[n]$
Entonces, una señal $Kx[n]$ produce una salida $Ky[n]$**

²⁰ http://www.isa.cie.uva.es/~felipe/docencia/ra12itielec/tema1_trasp.pdf

Figura 17. Ejemplo de Homogeneidad.

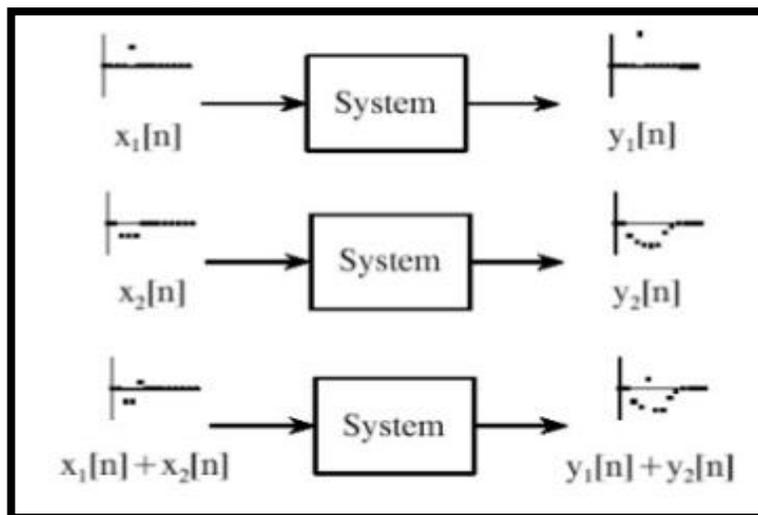


Fuente: Lara, C. (2008).

Aditividad: Un sistema es aditivo cuando la señal de salida es igual a la suma de las salidas generadas por las diferentes señales de entrada, figura 17.

Si $x_1[n]$ produce $y_1[n]$ y $x_2[n]$ produce $y_2[n]$
Entonces $x_1[n] + x_2[n]$ produce $y_1[n] + y_2[n]$

Figura 18. Ejemplo de Aditividad.



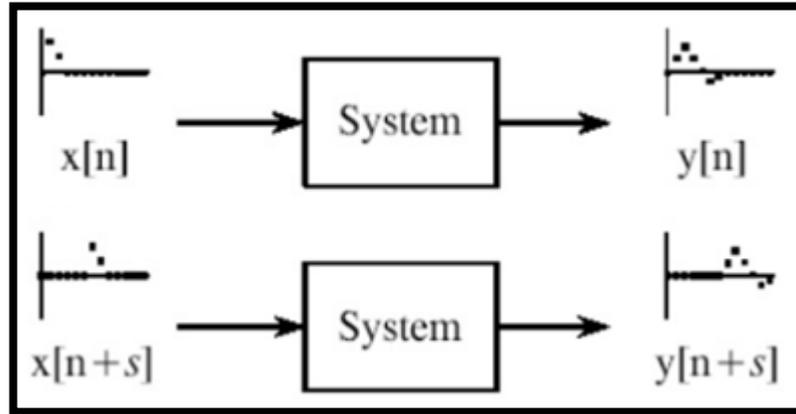
Fuente: Lara, C. (2008).²¹

Invariabilidad en el tiempo: Esto significa que al mover la señal de entrada en el tiempo produce un movimiento idéntico en la señal de salida.

²¹ Lara, C. (2008).

Si $x[n]$ produce $y[n]$
 Entonces $x[n + t]$ produce $y[n + t]$

Figura 19. Invariabilidad en el Tiempo.



Fuente: Lara, C. (2008).²²

3.6.4 Respuesta Paso Para Sistemas Lineales.

Con una entrada paso, la respuesta del sistema permite conocer empíricamente las características dinámicas del proceso, como el tiempo muerto (td), constante de tiempo (t), coeficiente de amortiguamiento zita entre otros.

3.6.5 Respuesta Paso A Sistemas De Primer Orden.

Si a un sistema de primer orden se le aplica una entrada escalón de amplitud M , la transformada de la Place de su respuesta es:

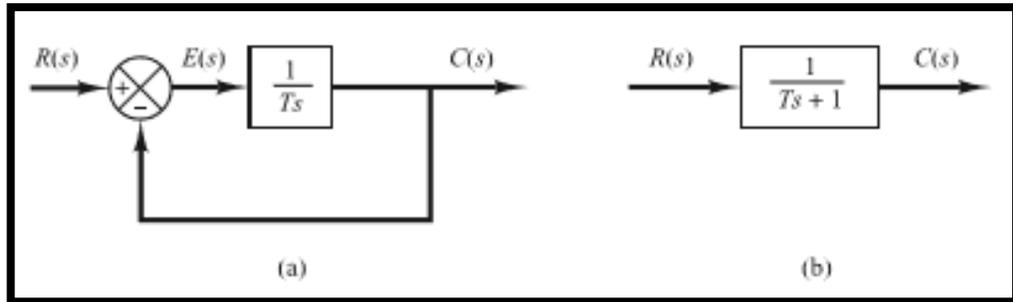
$$C(s) = \frac{K}{1+st} \frac{M}{s}$$

Si le aplicamos la anti transformada a la ecuación queda

$$C(t) = Mk(1 - e^{-\frac{t}{\tau}})$$

²² Lara, C. (2008).

Figura 20. a) Diagrama de Bloque de un Sistema de Primer Orden; b) Diagrama Simplificado



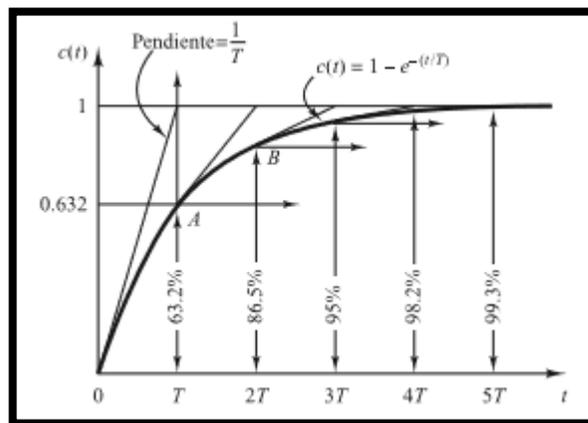
Fuente: Ogata, Benjamín C Kuo. (1998).

3.6.5.1 Constantes de Tiempo para Sistemas de Primer Orden

Se define constante para sistemas de primer orden, al tiempo que debe transcurrir para que la respuesta a escalón del sistema alcance 63.2% de su valor final, en la figura 20. Se puede notar que:

$$C(\tau) = 0.632 Mk$$

Figura 21. Respuesta Paso de un Sistema de Primer Orden



Fuente: Ogata, Benjamín C Kuo. (1998).

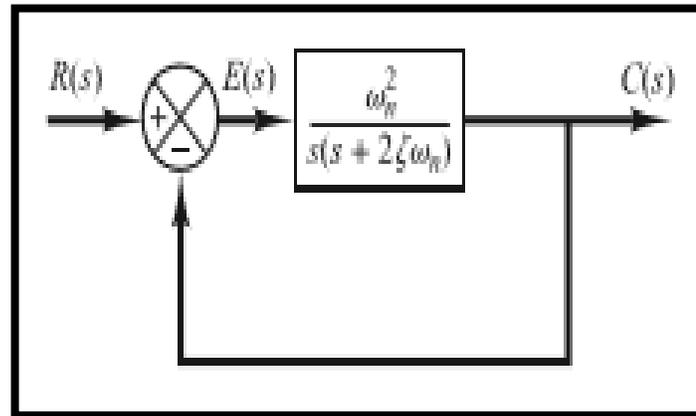
Obsérvese que, conforme más pequeña es la constante de tiempo T , más rápida es la respuesta del sistema. Otra característica importante de la curva de respuesta exponencial es que la pendiente de la línea de tangente en $t=0$ es $1/T$.

3.6.6 Respuesta Paso A Sistemas De Segundo Orden.

La función de transferencia en lazo cerrado de un sistema de segundo orden es de la forma:

$$G(s) = \frac{Wn^2}{s^2 + 2\zeta Wn + Wn^2} = \frac{C(s)}{R(s)}$$

Figura 22. Diagrama de bloque de un sistema de segundo orden.



Fuente: Ogata, benjamín CKuo. (1998).²³

Esta forma se denomina forma estándar del sistema de segundo orden. El comportamiento dinámico del sistema de segundo orden se describe a continuación en términos de dos parámetros zeta (ζ) y Wn . Si $0 < \zeta < 1$, los polos en lazo cerrado son complejos conjugados y se encuentran en el semiplano izquierdo del plano s . El sistema, entonces, se denomina subamortiguado y la respuesta transitoria es oscilatoria. Si $\zeta = 0$, la respuesta transitoria no se amortigua. Si $\zeta = 1$ el sistema se denomina críticamente amortiguado. Los sistemas sobre amortiguados corresponden $\zeta > 1$.

Ahora se obtendrá la respuesta del sistema que aparece en la Figura 20, para una entrada escalón unitario. Se considerarán tres casos diferentes: el subamortiguado ($0 < \zeta < 1$), el críticamente amortiguado ($\zeta = 1$) y el sobreamortiguado ($\zeta > 0$).

Por tanto la respuesta escalón del sistema de segundo orden estará expresada de la forma:

²³ Ogata, Benjamín C Kuo. (1998).

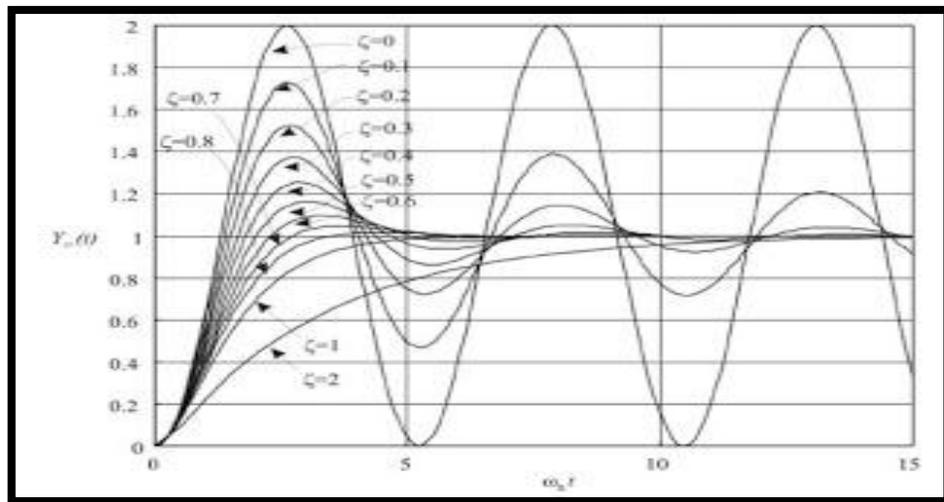
$$C(s) = \frac{Wn^2}{s^2 + 2\zeta Wn + Wn^2} * R(s)$$

$$C(s) = \frac{Wn^2}{s^2 + 2\zeta Wn + Wn^2} * \frac{k}{s}$$

Dónde: (Wn) Representa la frecuencia natural no amortiguada, (ζ) el factor de amortiguamiento relativo.

En la figura 23, se muestran las diferentes respuestas para cada uno de los casos anterior (considerando $K=1$, es decir para un escalón unitario).

Figura 23. Respuesta paso de un sistema de segundo orden con zita(ζ) variable.

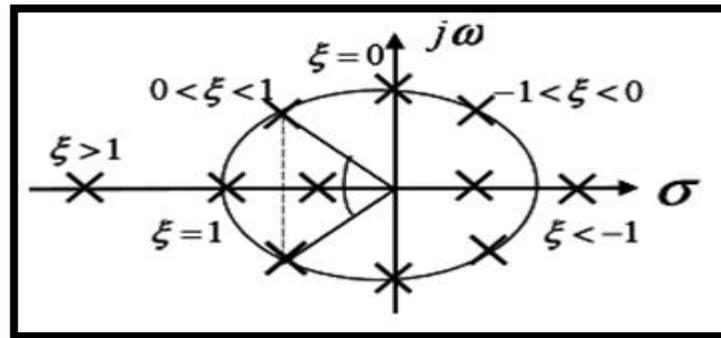


Fuente: Ogata, benjamín CKuo. (1998).

En la Figura 23, se observa que un sistema subamortiguado con (ζ) entre 0.5 y 0.8 se acerca al valor final con mayor rapidez que un sistema críticamente amortiguado o sobre amortiguado. Entre los sistemas que responden sin oscilación, un sistema críticamente amortiguado presenta la respuesta más rápida. Un sistema sobre amortiguado siempre es lento para responder a las entradas, en la figura 24, se evidencian la ubicación de los polos en el plano S, para los diferentes valores de zita (ζ)²⁴.

²⁴ Ogata, Benjamín C Kuo. (1998).

Figura 24. Ubicación de polos en un sistema de segundo orden en función de zeta (ζ).



Fuente: <http://www.gnu.org/copyleft/copyleft.html>

3.6.7 Especificaciones en el dominio temporal de los sistemas de segundo orden. En la teoría de control, la caracterización de la respuesta temporal de un sistema se suele hacer mediante lo que se conoce como especificaciones del sistema. Las especificaciones más empleadas son:

:

- Tiempo de retraso (Delay time). Es el tiempo necesario para que la respuesta alcance el 50% del valor final.
- Tiempo de subida (Rise time). Es el tiempo requerido para que la respuesta pase del 10 al 90% del valor final. También puede definirse como el tiempo de paso del 5% al 95% o del 0% al 100%.
- Tiempo de pico. Es el tiempo que pasa hasta alcanzarse el primer pico de sobrepaso.
- Sobrepico (Over shoot): Es el valor de pico máximo por unidad. Se suele expresar en porcentaje.
- Tiempo de establecimiento: Es el tiempo necesario para que la respuesta del sistema esté dentro de un porcentaje (sobre el 5%, aunque es variable según el autor) del valor final.

3.7 MODELOS

En el presente contexto, modelo es una abstracción matemática de un sistema, que permite sustituirlo cuando se estudia la relación entre las señales de entrada y salida. Ejemplos sencillos de modelos son las ecuaciones utilizadas para representar los componentes pasivos básicos en circuitos eléctricos. Así, un

resistor se modela con una ecuación lineal que relaciona tensión y corriente con una constante de proporcionalidad, mientras que para condensadores y bobinas se utilizan ecuaciones diferenciales para este fin. Los modelos normalmente simplifican la realidad y tienen validez solo para un rango restringido de puntos de operación. Por ejemplo, el modelo de una resistencia real como relación de proporcionalidad pierde validez si se utilizan frecuencias muy elevadas, pues efectos inductivos y capacitivos dejan de ser despreciables²⁵.

3.7.1 Clases De Modelos De Sistemas.

Existen diferentes tres clases de modelos de sistemas como son:

Modelo intuitivo: El cual no involucra formalización matemática y se relaciona con experiencias del pasado.

Modelos gráficos y tablas: Algunos sistemas describen sus propiedades por medio de gráficos o tablas, los sistemas lineales pueden ser descritos por una respuesta impulso, paso o respuesta de frecuencia.

Modelos matemáticos: Se definen como la relación que existe entre el mundo real y las matemáticas. Estos modelos describen la relación entre los sistemas variables en términos matemáticos como ecuaciones diferenciales para tiempo continuo o ecuaciones de diferencia para tiempo discreto.

3.7.2 Proceso De Modelización.

Ningún modelo matemático puede representar cualquier componente o sistema físicos con precisión. Siempre se involucran aproximaciones y suposiciones. Tales aproximaciones y suposiciones restringen el nivel de validez del modelo matemático. (El grado de aproximación puede determinarse solamente mediante experimentos). Así pues, al hacer una predicción acerca del funcionamiento del sistema, debe tenerse presente cualquier aproximación o suposición involucrada en el modelo²⁶.

3.7.3 Métodos Para Obtener Un Modelo.

Existen dos métodos para obtener un modelo: modelización matemática y sistemas de identificación.

Modelización Matemática: Esta se basa en leyes físicas, como las leyes de Newton y el balance de ecuaciones, a fin de describir el comportamiento dinámico de un fenómeno o proceso.

²⁵ Alvarado Moya, P. (2008).

²⁶ Ogata, Benjamín C Kuo. (1987).

Cualquier tentativa de diseño de un sistema debe empezar a partir de una predicción de su funcionamiento antes de que el sistema pueda diseñarse en detalle o construirse físicamente. Tal predicción se basa en una descripción matemática de las características dinámicas del sistema. A esta descripción matemática se le llama modelo matemático. Para los sistemas físicos, la mayoría de los modelos matemáticos que resultan útiles se describen en términos de ecuaciones diferenciales. La dinámica de sistemas trata del modelado matemático y el análisis de la respuesta de los sistemas dinámicos. Hoy en día, el diseño de ingeniería requiere de un concienzudo estudio de esa materia.

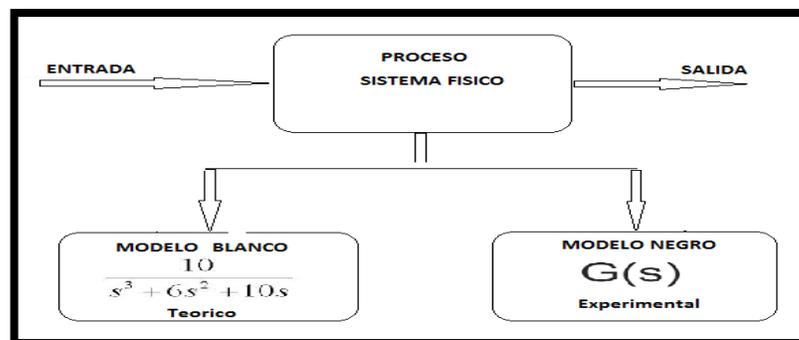
Sistemas de Identificación: Este es una aproximación experimental en el cual se desarrollan, algunos experimentos en el proceso y se determina un modelo con parámetros asignados que no tienen significado físico.

Los modelos en los sistemas de identificación se obtienen por medio de dos procedimientos:

Off-line: (fuera de línea), donde se hace un experimento enviando una señal de prueba al proceso, se toman los datos de las medidas correspondientes y se llevan al computador para ser procesada y a partir de estos datos hallar el modelo del sistema.

On-line: (en línea), donde el computador está conectado al proceso y este puede ser controlado, lo que quiere decir que si el proceso cambia el modelo también. Con base en lo anterior los modelos se pueden clasificar en modelos blancos y modelos de cajas negras, figura 25.

Figura 25. Tipos de Modelos.



Fuente: Elaboración propia.

Los modelos de identificación experimental (modelos de cajas negras) y los de la modelización matemática (modelos de cajas blancas), deben cumplir con ciertos criterios para su implementación, tabla 6.

Tabla 6. Métodos de Identificación de Sistemas

| MODELO MATEMATICO | EXPERIMENTAL |
|--|--|
| <ul style="list-style-type: none"> • Para procesos básicos. • Para procesos en el que existe un modelo verificado. • Cuando la planta no existe. • Cuando no se pueden realizar experimentos en el proceso. • Cuando se conocen las constantes que intervienen en el proceso. | <ul style="list-style-type: none"> • Para sistemas complejos. • Cuando la planta existe. • Cuando no se conoce la estructura interna del sistema. |

Fuente: Elaboración propia

3.8 CONSTRUCCIÓN DE MODELOS.

Para construir el modelo se deben obtener datos observados o datos experimentales, donde los datos de entrada como de salida son guardados y analizados en una computadora para posteriormente encontrar el modelo adecuado.

3.8.1 Señales De Prueba Típicas

Las señales de prueba que se usan regularmente son funciones escalón, rampa, parábola, impulso, etc. Con estas señales de prueba, es posible realizar con facilidad análisis matemáticos y experimentales de sistemas de control, ya que las señales son funciones del tiempo muy simples.

La forma de la entrada a la que el sistema estará sujeto con mayor frecuencia en una operación normal determina cuál de las señales de entrada típicas se debe usar para analizar las características del sistema. Si las entradas para un sistema de control son funciones del tiempo que cambian en forma gradual, una función rampa será una buena señal de prueba. Asimismo, si un sistema está sujeto a perturbaciones repentinas, una función escalón será una buena señal de prueba; y para un sistema sujeto a entradas de choque, una función impulso será la mejor. Una vez diseñado un sistema de control con base en las señales de prueba, por lo general el comportamiento del sistema en respuesta a las entradas reales es satisfactorio. El uso de tales señales de prueba permite comparar el comportamiento de todos los sistemas sobre la misma base.

Respuesta transitoria y respuesta en estado estacionario. La respuesta en el tiempo de un sistema de control consta de dos partes: la respuesta transitoria y la respuesta en estado estacionario. La respuesta transitoria se refiere a la que va del estado inicial al estado final. La respuesta en estado estacionario se entiende

la manera cómo se comporta la salida del sistema conforme t tiende a infinito. Por tanto, la respuesta del sistema $y(t)$ se puede escribir como:

$$y(t) = y_{tr} + y_{ss}$$

Donde el primer término del miembro derecho de la ecuación es la respuesta transitoria y el segundo término es la respuesta en el estado estacionario.

3.8.2 Estabilidad Absoluta, Estabilidad Relativa Y Error En Estado Estacionario.

Al diseñar un sistema de control, se debe ser capaz de predecir su comportamiento dinámico a partir del conocimiento de los componentes. La característica más importante del comportamiento dinámico de un sistema de control es la estabilidad absoluta, es decir, si el sistema es estable o inestable. Un sistema de control está en equilibrio si, en ausencia de cualquier perturbación o entrada, la salida permanece en el mismo estado. Un sistema de control lineal e invariante con el tiempo es estable si la salida termina por regresar a su estado de equilibrio cuando el sistema está sujeto a una condición inicial. Un sistema de control lineal e invariante con el tiempo es críticamente estable si las oscilaciones de la salida continúan de forma indefinida. Es inestable si la salida diverge sin límite a partir de su estado de equilibrio cuando el sistema está sujeto a una condición inicial.

En realidad, la salida de un sistema físico puede aumentar hasta un cierto grado, pero puede estar limitada por «detenciones» mecánicas, o el sistema puede colapsarse o volverse no lineal una vez que la salida excede cierta magnitud, por lo cual ya no se aplican las ecuaciones diferenciales lineales. Entre los comportamientos importantes del sistema (aparte de la estabilidad absoluta) que deben recibir una cuidadosa consideración están la estabilidad relativa y el error en estado estacionario.

Como un sistema de control físico implica un almacenamiento de energía, la salida del sistema, cuando este está sujeto a una entrada, no sucede a la entrada de inmediato, sino que muestra una respuesta transitoria antes de alcanzar un estado estacionario. La respuesta transitoria de un sistema de control práctico, con frecuencia, muestra oscilaciones amortiguadas antes de alcanzar un estado estacionario.

Si la salida de un sistema en estado estacionario no coincide exactamente con la entrada, se dice que el sistema tiene un error en estado estacionario. Este error indica la precisión del sistema. Al analizar un sistema de control, se debe examinar el comportamiento de la respuesta transitoria y el comportamiento en estado estacionario.

3.9 SEÑALES EN TIEMPO DISCRETO

Las señales en tiempo discreto surgen, si el sistema involucra la operación de muestreo de señales en tiempo continuo. La señal muestreada es $x(0), x(1), x(2T), \dots$, donde T es el periodo de muestreo. Dicha secuencia de valores que surgen de las operaciones de muestreo normalmente se escriben como $x(k)$. Si el sistema incluye un proceso iterativo realizado por una computadora, la señal involucrada es una secuencia de números $x(0), x(1), x(2), \dots$ la secuencia de números normalmente se escribe como $x(k)$, donde el arguanto k indica el orden en que se presentan los números en la secuencia, $x(0), x(1), x(2), \dots$. A un que $x(k)$, es una secuencia de números, esta se puede considerar como una señal muestreada de $x(t)$, cuando el periodo de muestreo T es de 1 segundo.

3.9.1 Tiempo De Muestreo.

Para convertir una señal analógica en digital, el primer paso consiste en realizar un muestreo (sampling) de ésta, o lo que es igual, tomar diferentes muestras de tensiones o voltajes en diferentes puntos de la onda sinusoidal. La frecuencia a la que se realiza el muestreo se denomina razón, tasa o también frecuencia de muestreo y se mide en kilo Hertz (kHz).

3.9.2 Teorema Del Muestreo.

Para reconstruir la señal original a partir de una señal muestreada, existe una frecuencia mínima que la operación de muestreo debe satisfacer.

El teorema de muestreo, establece que si ω_s , definida como $2\pi / T$, donde T es el periodo de muestreo, es mayor que $2\omega_1$, ($\omega_s > 2\omega_1$), donde ω_1 es la componente de más alta frecuencia presente en la señal de tiempo continuo $x(t)$, entonces la señal $x(t)$ se puede reconstruir completamente a partir de la señal muestreada $x^*(t)$ ²⁷.

En resumen el Teorema del Muestreo de Nyquist-Shannon dice que se puede reproducir de manera exacta una onda si la frecuencia de muestreo es, como mínimo, el doble de la más alta que se pueda escuchar. En el caso del oído humano esta frecuencia corresponde a 20.000 Hz por lo tanto la frecuencia de muestreo más adecuada será de 40.000 Hz. Algunos estudios aumentan esta cifra hasta los 44.100 Hz, que es la que se suele usar.²⁸

3.9.2.1 Método Alternativo Tiempo de Muestreo Matlab.

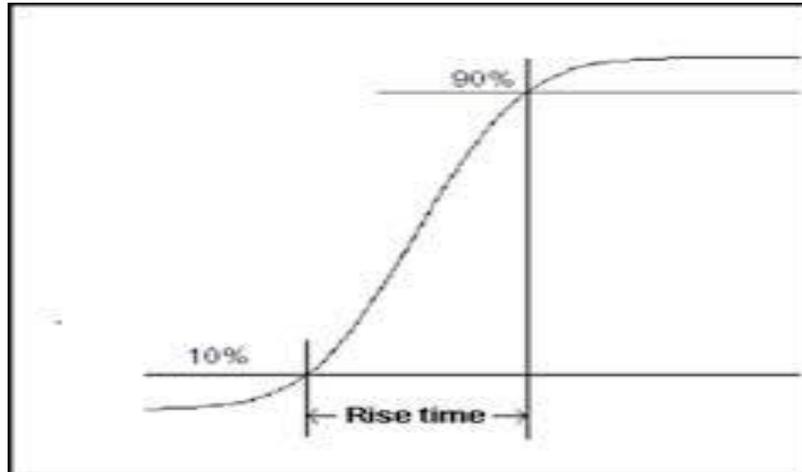
El Rise Time Es el tiempo necesario para que la salida del sistema alcance un determinado porcentaje del valor final de la referencia. Si no se especifica dicho porcentaje se entiende que se medirá hasta que alcance el 100% del valor final.

²⁷ Ogata, Benjamín C Kuo. (1998).

²⁸ WWW.FOTONOSTRA.COM/DIGITAL/MUESTREOAUDIO.HTM

Otra forma de establecer el tiempo de subida es mediante una horquilla de valores, es decir, medir el tiempo que transcurre desde un porcentaje inicial hasta uno final, figura 26.

Figura 26. Rise Time Orquilla de Valores de la Señal.



Fuente: S/A (2014).

Un método alternativo para hallar el Sampling time (tiempo de muestreo) consiste en graficar la señal del sistema con la ayuda del comando “step” de Matlab para así obtener el Rise Time (tiempo de subida). El rise time es el tiempo que tarda en alcanzar la señal 90% del valor final, este valor se divide por 10 para obtener a si el sampling time (tiempo de muestreo), este método simplifica la operación y determina correctamente el tiempo de muestreo de la señal, a continuación se dará un ejemplo para hallar el tiempo de muestreo utilizando este método.

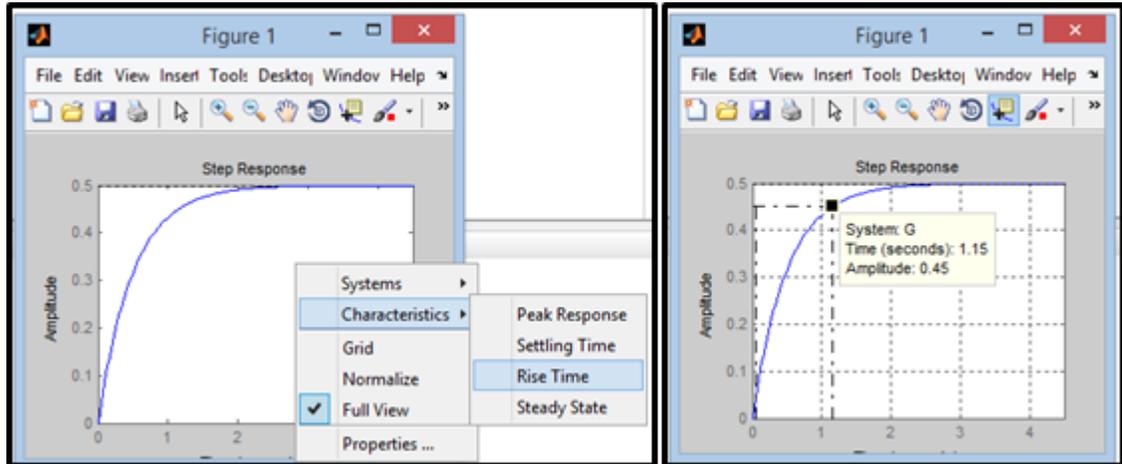
Modelo del sistema

$$G(s) = \frac{1}{s+2}$$

Instrucción para graficar la respuesta del sistema a una entrada escalo unitario.

```
num=1;    den=[1 2];    step(G)    G=tf (num,den);
```

Figura 27. Representación del Rise Time



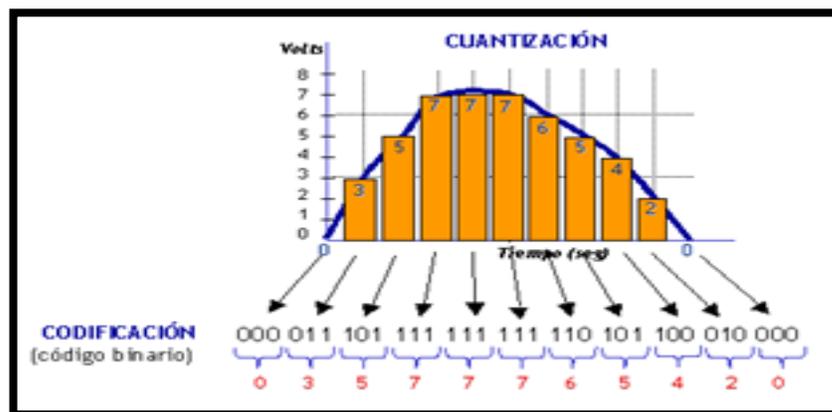
Fuente: Elaboración propia.

En la figura 27, se puede apreciar el valor que toma el Rise timen para esta señal el cual es de 1.1seg que al ser dividido por 10 nos da un tiempo de muestreo de 0.11seg.

3.9.3 Cuantificación De La Señal.

Una vez realizado el muestreo, el siguiente paso es la cuantificación de la señal analógica. Para esta parte del proceso los valores continuos de la sinusoide se convierten en series de valores numéricos decimales discretos correspondientes a los diferentes niveles o variaciones de voltajes que contiene la señal analógica original.

Figura 28. Cuantificación de una Señal Análoga.



Fuente: <http://www.eveliux.com/mx/Conversion-Analogico-Digital-ADC.html>.²⁹

²⁹ <http://www.eveliux.com/mx/Conversion-Analogico-Digital-ADC.html>

La codificación permite asignarle valores numéricos binarios equivalentes a los valores de tensiones o voltajes que conforman la señal eléctrica analógica original figura 28.

Un muestreo realizado a 8 bits tendrá menos calidad y respuesta de frecuencia que otro realizado a 16 bits, y se aproximará más a la forma que tenía la señal analógica original.

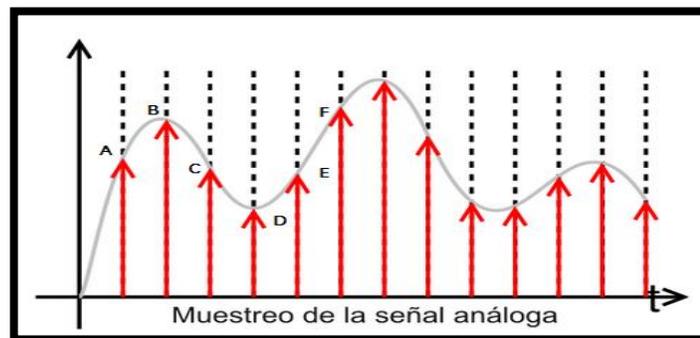
3.9.3 Transformada Z.

La transformada z tiene el mismo propósito para sistemas en tiempo discreto, lineal e invariantes en el tiempo, que la transformada de Laplace para tiempo continuo, lineales e invariantes en el tiempo. El método por computadora para el análisis de datos en tiempo discreto da como resultado ecuaciones en diferencias. Con el método de transformada z, las ecuaciones lineales e invariantes en el tiempo se pueden transformar en ecuaciones algebraicas es to facilidad el análisis de la respuesta transitoria de los sistemas de control digital.

3.9.4 Comportamiento De Los Sistemas En Tiempo Continúo Y Discreto.

Considerando una señal continua $Y(t)$, que será observada o medida en la computadora a intervalos de muestreo $t(k) = kT_s$, para $k=0,1,2,3,\dots$ el periodo en el cual el computador toma las muestras A,B,C,D,E,F se llama periodos de muestreo T_s , figura 29.

Figura 29. Comportamiento de una Señal en el Tiempo



Fuente: <http://anyelosistemas.blogspot.com.co>

El objetivo de los sistemas de identificación, es trabajar con señales en tiempo discreto, por ello las señales en tiempo continuo se pueden aproximar a tiempo discreto. Como ejemplo podemos tomar un sistema de primer orden en tiempo continuo, el cual se puede aproximar en tiempo discreto.

Debido a que el computador trabaja con ecuaciones en diferencia, (tiempo discreto) estas resultan de la conversión de las correspondientes ecuaciones diferenciales (tiempo continuo). Por esta razón se obtiene la ecuación de diferencia general de n-orden:

$$y(k) + A_1y(k - 1) + \dots + A_ny(k - n) = Bu(k) + B_1u(k - 1) + \dots + B_nu(k - n)$$

La función de transferencia en tiempo discreto es:

$$\frac{y(k)}{u(k)} = \frac{B(q^{-1})}{A(q^{-1})} = \frac{B(z)}{A(z)}$$

Para $q^{-1} = z^{-1}$, $B(q^{-1})$ es el polinomio B corresponde al operador de desplazamiento $(b_0 + b_1 q^{-1} + \dots + b_n q^{-n})$, el polinomio $A(q^{-1})$ con el operador de desplazamiento $(a_0 + a_1 q^{-1} + \dots + a_n q^{-n})$.

Un sistema lineal con perturbaciones se describe con la ecuación:

$$y(k) = G(q^{-1})u(k) + H(q^{-1})e(k)$$

Donde $G(q^{-1})$ representa la función de transferencia y $H(q^{-1})$ el aporte al ruido.

3.10 IDENTIFICACIÓN DE SISTEMAS DINÁMICOS

3.10.1 Modelos No Paramétricos Para Sistemas Lineales Invariantes En El Tiempo.

Los de identificación (modelización experimental) se estudian como modelos de cajas negras (Black-box), en los cuales se envían señales de prueba al sistema, para observar su salida sin saber lo que suceda al interior.

Los métodos de identificación se caracterizan porque, los modelos resultantes son gráficos o funciones las cuales no son necesariamente parametrizadas por vector de parámetros. La respuesta impulso, la respuesta paso o escalón y la respuesta en frecuencia, son algunos ejemplos de modelos no paramétricos, se obtienen a partir de la salida del proceso cuando ha sido aplicada una señal de entrada ya sea impulso, paso o sinusoidal.

Algunos de los modelos no paramétricos son:

Análisis de transcientes: La entrada es tomada como una señal paso o impulso y la salida constituye el modelo del sistema.

Análisis de correlación: La entrada es un ruido blanco (White-Noise). La estimación de los pesos de la función se obtiene a partir de la relación de una covarianza cruzada o normalizada (Cross-correlation) entre la salida y la entrada.

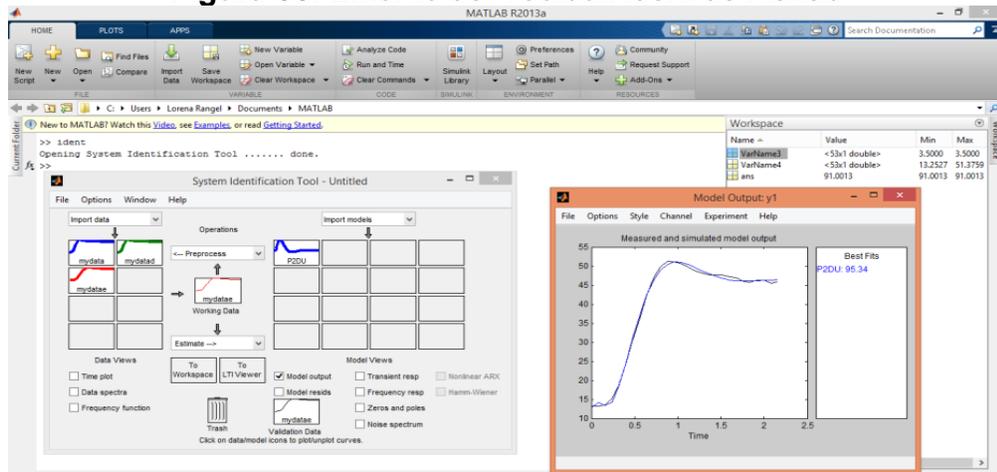
Análisis espectral: La respuesta en frecuencia puede ser estimada por entradas arbitrarias dividiendo el espectro cruzado (cross-spectrum) entre la salida y la entrada por el espectro de entrada.

3.10.2 IDENT (Toolbox de Matlab).

Este método nos ayuda a identificar o estimar un modelo matemático (función de transferencia) de una planta (ejemplo motor dc) sin necesidad de conocer los parámetros técnicos de la planta o de conocerla realmente, entonces habría que considerarlo como un proceso desconocido al cual a través de su adquisición de datos podremos encontrar un modelo con un grado de compatibilidad alto.

Para este fin la herramienta “Ident” de Matlab nos permite configurarlo para lograr la identificación de forma exitosa, figura 30.

Figura 30. Entorno del Toolbox Ident de Matlab



Fuente: Elaboración propia

3.11 SISTEMAS DE CONTROL.

Un sistema de control puede ser definido como el medio a través del cual una cantidad o variable cualquiera de interés en una máquina, mecanismo o proceso, es mantenido o alterado de acuerdo con un patrón de comportamiento deseado.

Control manual: El operador aplica las correcciones que cree necesarias.

Control automático: Un controlador automático compara el valor real de la salida de una planta con la entrada de referencia (el valor deseado), determina la desviación y produce una señal de control que reduce la desviación a cero o a un valor pequeño. La manera en la cual el controlador automático produce la señal de control se denomina acción de control.

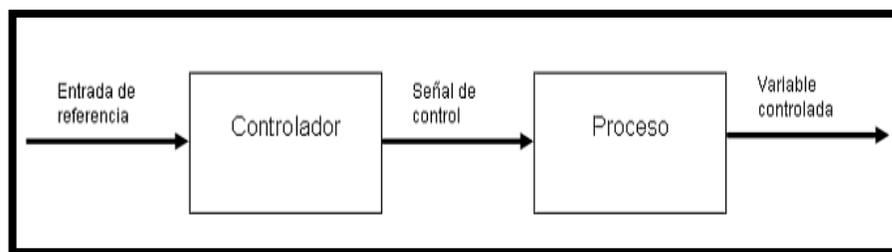
Control programado: Realiza todas las labores del control automático, pero su solución es programada. Se puede modificar su proceso de operación o ley de control.

3.11.1 Clases De Sistemas De Control.

Existen dos clases de sistemas de sistemas de control programado como son:
Sistemas de control en lazo abierto

Aquel en el que ni la salida ni otras variables del sistema tienen efecto sobre el control. NO TIENE REALIMENTACIÓN, figura 31.

Figura 31. Sistema de Control en Lazo Abierto.

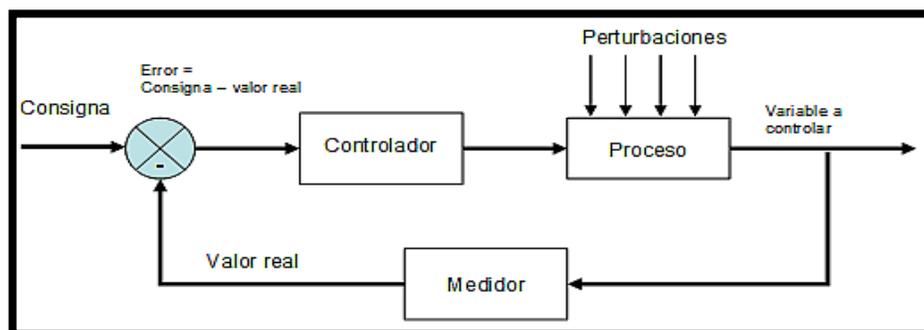


Fuente: http://www.infoagro.com/riegos/control_riego_y_fertilizacion.htm

Sistemas de control en lazo cerrado

En un sistema de control de lazo cerrado, la salida del sistema y otras variables, afectan el control del sistema. SI TIENE REALIMENTACIÓN, figura 32.

Figura 32. Diagrama de Bloques de un Proceso de Control en Lazo Cerrado.



Fuente: http://www.infoagro.com/riegos/control_riego_y_fertilizacion.htm

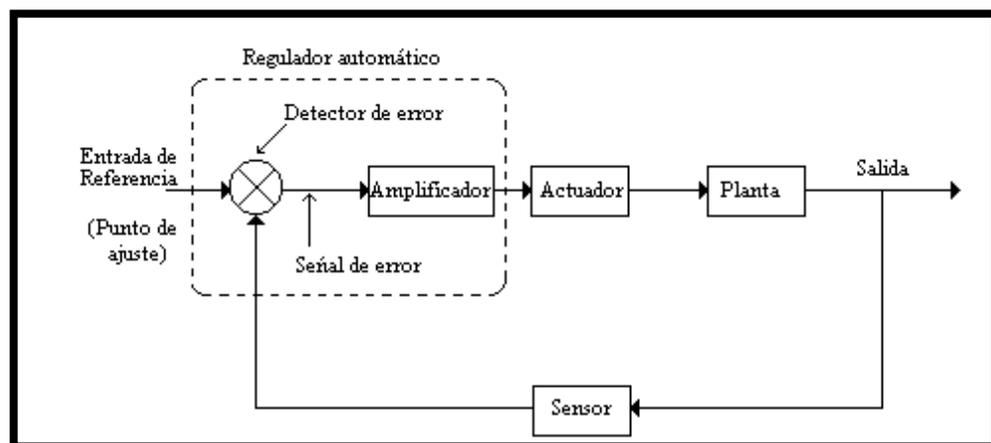
Los sistemas de control deben conseguir los siguientes objetivos:

1. Ser estables y robustos frente a perturbaciones y errores en los modelos.
2. Ser eficiente según un criterio preestablecido evitando comportamientos bruscos e irreales.

3.11.2 CONTROL ANÁLOGO

La Figura 33 es un diagrama de bloques de un sistema de control industrial que consiste en un controlador automático, un actuador, una planta y un sensor (elemento de medición). El controlador detecta la señal de error, que por lo general, está en un nivel de potencia muy bajo, y la amplifica a un nivel suficientemente alto.

Figura 33. Diagrama de bloques de un sistema de control industrial



Fuente: <http://www.monografias.com/trabajos93/controlador-electronico-velocidad-pid-usando-cpld/controlador-electronico-velocidad-pid-usando-cpld.shtml>

El *controlado automático* está constituido por un *detector de error* y un *amplificador*. También suele haber un *circuito de retroalimentación* adecuado, junto con un *amplificador*, que se utilizan para alterar la señal de error amplificándola, y a veces diferenciándola y/o integrándola, para producir una mejor señal de control.

El *actuador* es un dispositivo de potencia que produce la entrada a la planta, de acuerdo con la señal de control, de modo que la señal de retroalimentación corresponda a la señal de entrada de referencia. La salida de un controlador alimenta a un actuador, que bien pueden ser un motor.

El *sensor o elemento de medición*, es un dispositivo que convierte la variable de salida en otra variable adecuada, como un desplazamiento, o voltaje, que se utilizan para comparar la salida con la señal de entrada de referencia. Este elemento es el camino de retroalimentación en el sistema de lazo cerrado³⁰.

3.11.3 Sistemas De Control Con Controlador Digital.

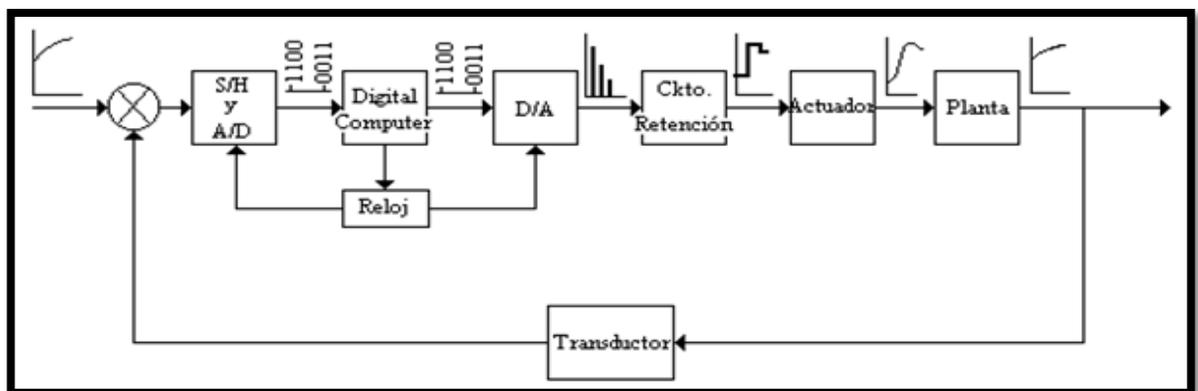
La señal de error se convierte a forma digital mediante el circuito de muestreo y retención y el convertidor analógico-digital. La conversión se hace en el tiempo de muestreo.

La computadora digital procesa las secuencias de números por medio de un algoritmo y produce nuevas secuencias de números. En cada instante de muestreo se debe convertir un número codificado (en general un número binario que consiste en ocho o más dígitos binarios) en una señal física de control, la cual normalmente es una señal en tiempo continuo o analógico.

El convertidor digital-analógico y el circuito de retención convierten la secuencia de números en código numérico a una señal continua por secciones. El reloj en tiempo real de la computadora sincroniza los eventos.

La salida del circuito de retención, una señal en tiempo continuo, se alimenta a la planta, ya sea de manera directa o a través de un actuador, para controlar su dinámica. La salida de la planta es una señal en tiempo continuo.

Figura 34. Diagrama de Bloques de un Sistema de Control Digital



Fuente: <http://www.monografias.com/trabajos93/controlador-electronico-velocidad-pid-usando-cpld/controlador-electronico-velocidad-pid-usando-cpld.shtml>

³⁰<http://www.monografias.com/trabajos93/controlador-electronico-velocidad-pid-usando-cpld/controlador-electronico-velocidad-pid-usando-cpld.shtml>

3.11.4 Comparativa Entre Control Análogo Frente Al Discreto.

Control análogo

- La ventaja del controlador analógico es que el sistema trabaja en tiempo real y puede tener un ancho de banda muy grande. Esto es equivalente a tener una frecuencia de muestreo infinita, de modo que el efecto del controlador siempre está presente.

Control discreto

- Los componentes digitales tienen menor susceptibilidad al envejecimiento y a las variaciones de las condiciones ambientales.
- Los componentes digitales son menos sensibles al ruido y las perturbaciones.
- Los procesadores digitales tienen un tamaño y peso menor.
- Un cambio en el control no requiere cambios en el hardware.
- Proporcionan una mayor sensibilidad a las variaciones en los parámetros.

3.12 TIPOS DE CONTROLADORES

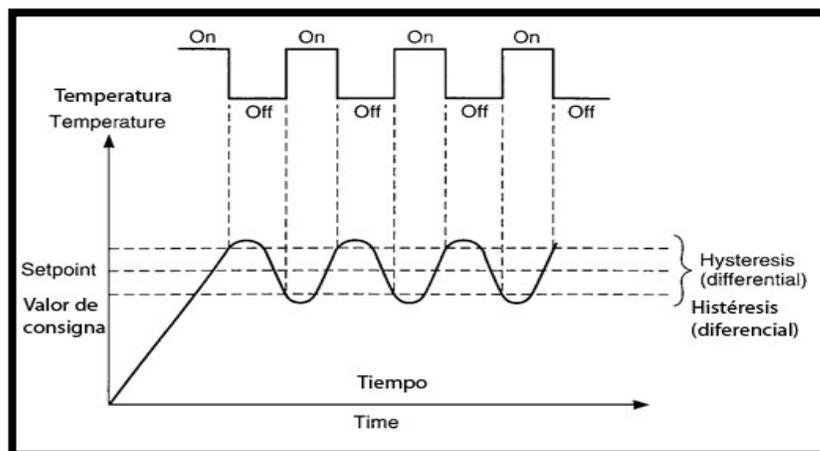
La mayoría de los controladores industriales emplean como fuente de energía la electricidad o un fluido presurizado, como el aceite o el aire. Los controladores también pueden clasificarse, según el tipo de energía que utilizan en su operación, como neumáticos, hidráulicos o electrónicos. El tipo de controlador que se use debe decidirse basándose en la naturaleza de la planta y las condiciones de operación, incluyendo consideraciones tales como seguridad, costo, disponibilidad, fiabilidad, precisión, peso y tamaño. Los controladores industriales se clasifican, de acuerdo con sus acciones de control, como:

1. De dos posiciones o controladores on-off
2. Controladores proporcionales
3. Controladores integrales
4. Controladores proporcionales-integrales
5. Controladores proporcionales-derivativos
6. Controladores proporcionales-integrales-derivativos

3.12.1 Controladores On-Off.

Los controladores «sí/no», también llamados de «encendido/apagado» o «todo/nada», son los sistemas de control más básicos. Estos envían una señal de activación («sí», «encendido» o «1») cuando la señal de entrada es menor que un nivel de referencia (definido previamente), y desactivan la señal de salida («no», «apagado» o «0») cuando la señal de entrada es mayor que la señal de referencia figura 35.

Figura 35. Control On-Off



Fuente: www.tecnoficio.com/electricidad/instrumentacion_industrial4.php

3.12.2 El Controlador PID.

Si se puede obtener un modelo matemático de la planta, es posible aplicar diversas técnicas de diseño con el fin de determinar los parámetros del controlador que cumpla las especificaciones del transitorio y del estado estacionario del sistema en lazo cerrado. Sin embargo, si la planta es tan complicada que no es fácil obtener su modelo matemático, tampoco es posible un método analítico para el diseño de un controlador PID. En este caso, se debe recurrir a procedimientos experimentales para la sintonía de los controladores PID³¹.

Es interesante señalar que más de la mitad de los controladores industriales que se usan hoy en día utilizan esquemas de control PID o PID modificado. Los controladores PID analógicos, son principalmente de tipo hidráulico, neumático, electrónico, eléctrico o sus combinaciones. En la actualidad, muchos de estos se transforman en formas digitales mediante el uso de microprocesadores³².

Un controlador PID se caracteriza por combinar tres acciones (P, I y D) mediante el siguiente algoritmo de control:

³¹ Ogata, Benjamin C Kuo. (2010).

³² <http://www.monografias.com/trabajos27/analogico-y-digital/analogico-y-digital.shtml>

$$u(t) = Kp \left[e(t) + \frac{1}{Ti} \int e(t)dt + Td \frac{de(t)}{dt} \right] = P + I + D$$

Dónde:

$u(t)$ = Entrada de control del proceso.

$e(t)$ = Error de la señal

Kp = Ganancia proporcional,

Ti = Constante de tiempo integral

Td = Constante de tiempo derivativa.

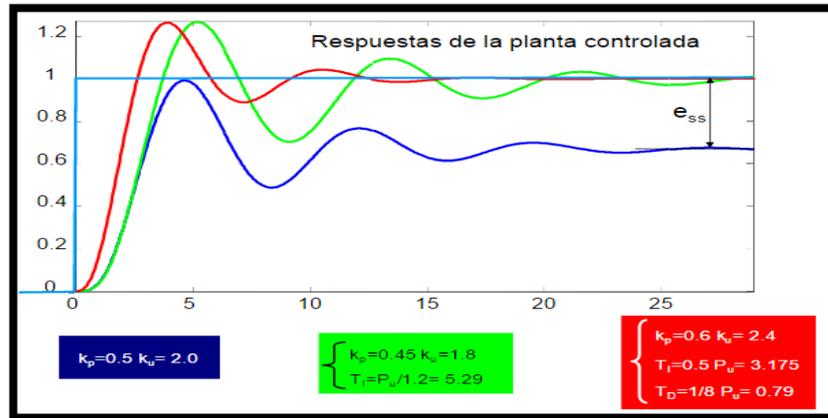
En el dominio de la frecuencia, el controlador PID se puede escribir como:

$$u(z) = Kp \left[1 + \frac{1}{Ti(s)} + Td(s) \right] E(s)$$

Este algoritmo está considerado como el PID estándar por la ISA (Instrument Society of America). A continuación se resumen los términos básicos:

- Acción proporcional (P): es la acción que produce una señal proporcional a la desviación de la salida del proceso respecto al punto de consigna.
- Acción integral (I): es la acción que produce una señal de control proporcional al tiempo que la salida del proceso ha sido diferente del punto de consigna.
- Acción derivativa (D): es la acción que produce una señal de control proporcional a la velocidad con que la salida del proceso está cambiando respecto del punto de consigna.
- Constante de tiempo integral (T_i): es el tiempo, generalmente expresado en minutos, que debe transcurrir para que la acción integral alcance (iguale o repita) a la acción proporcional.
- Constante de tiempo derivativa (T_d): es el intervalo de tiempo, generalmente expresado en minutos, en el que la acción derivativa adelanta a la acción proporcional [4].

Figura 36. Representación Gráfica de las Acciones De Control



Fuente: <http://slideplayer.es/slide/3079472/> ³³

Tal como se muestra en la figura 36, cada acción de control tiene una respuesta característica. La tabla 7, muestra cómo varían la estabilidad, la velocidad y el error en estado estacionario cuando se modifican los parámetros del controlador. Es necesario señalar que esta tabla contiene un conjunto de reglas heurísticas y, por tanto, hay excepciones.

Tabla 7. Característica de las acciones de control PID

| | Kp aumenta | Ti disminuye | Td aumenta |
|--------------------------------|-------------------|---------------------|-------------------|
| Estabilidad | Se reduce | Disminuye | Aumenta |
| Velocidad | Aumenta | Aumenta | Aumenta |
| Error est. estacionario | No eliminado | Eliminado | No eliminado |

Fuente: Elaboración propia.

3.12.2.1 Controlador PID discreto.

Aplicando la transformada z a la función de transferencia para el controlador PID en tiempo continuo se convierte en:

$$U(z) = Kp \left[1 + \frac{T}{T_i(1-z^{-1})} + T_d \frac{(1-z^{-1})}{T} \right] E(z)$$

La función de transferencia discreta, también puede ser representada como:

³³ <http://slideplayer.es/slide/3079472/>

$$\frac{U(z)}{E(z)} = a + \frac{b}{1-z^{-1}} + c(1-z^{-1})$$

Donde:

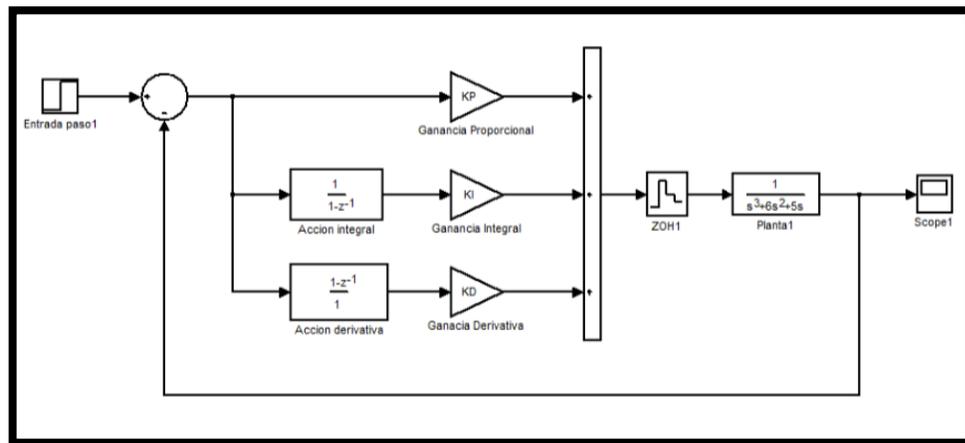
$$a = K_p \quad b = \frac{K_p T}{T_i} \quad c = \frac{K_p T_d}{T}$$

La función de transferencia de un control PID discreto en la forma de posición está definido como:

$$u(z) = Kp + \frac{KI}{1-z^{-1}} + KD(1-z^{-1})$$

En la figura 37, se muestra la configuración del lazo de control

Figura 37. Lazo de Control de la Forma de Posición Simulink



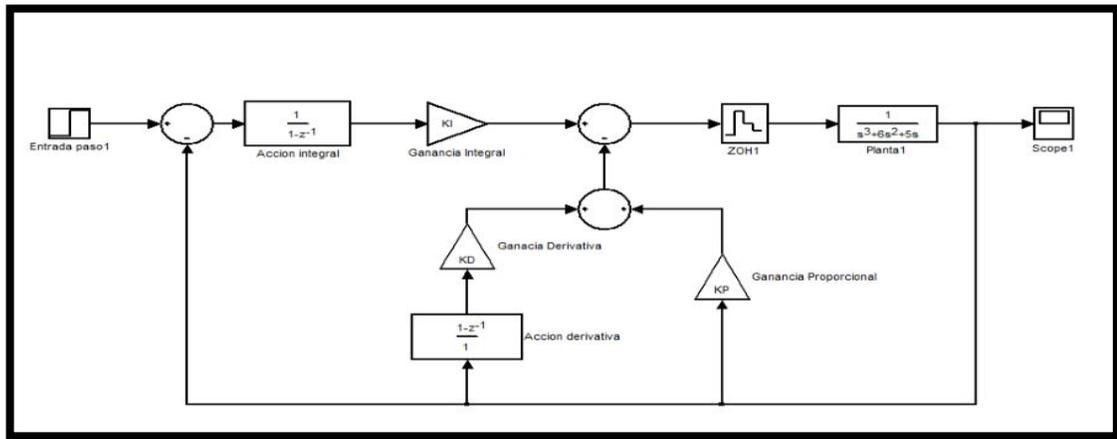
Fuente: Elaboración propia

En la figura 38, se muestra la configuración del control Las función de transferencia de un control PID discreto en la forma de velocidad el cual está definido como:

$$u(z) = -Kp Y(z) + \frac{KI}{1-z^{-1}} E(z) - KD(1-z^{-1})Y(z)$$

Dónde: Y (z) es la salida del proceso y E (z) es la señal de error

Figura 38. Lazo de Control de la Forma de Velocidad Simulink



Fuente: Elaboración propia.

3.13 TÉCNICAS DE SINTONIZACIÓN PID.

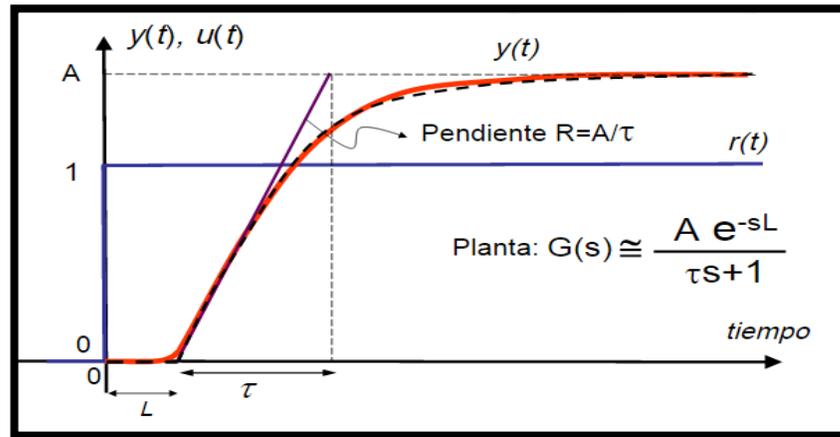
3.13.1 Métodos De Sintonización De Controladores Mediante Ziegler-Nichols (ZN). Ziegler y Nichols propusieron una serie de reglas para afinar controladores PID con base a una respuesta experimental. Definieron dos métodos³⁴.

3.13.1.1 Primer método.

Se obtiene experimentalmente la respuesta de la planta a una entrada escalón y si la respuesta no tiene oscilaciones y además posee un retardo tal que se forma una “ese”, puede obtenerse los parámetros del controlador PID utilizando el primer método. En la figura 39, se observa la respuesta en forma de S, esta respuesta se caracteriza por tener un tiempo de atraso τ y la constante de tiempo τ_c . Y se puede representar como un sistema de primero orden con retardo.

³⁴ Ogata, Benjamín C Kuo. Ingeniería de control moderno, (2010).

Figura 39. Respuesta en Forma de S.



Fuente: <http://slideplayer.es/slide/3079472/>³⁵

Para obtener L y T , se traza una recta tangente al punto de inflexión de la respuesta, la intersección con el eje del tiempo y con el valor final de la amplitud forman las distancias L y T

Con L y T , se obtienen los parámetros del controlador PID utilizando la tabla 8.

Tabla 8. Valores de Sintonización ZN Método 1

| Tipo de controlador | K_p | T_i | T_d |
|---------------------|-------------------|-----------------|--------|
| P | $\frac{T}{L}$ | ∞ | 0 |
| PI | $0.9 \frac{T}{L}$ | $\frac{L}{0.3}$ | 0 |
| PID | $1.2 \frac{T}{L}$ | $2L$ | $0.5L$ |

Fuente: Ingeniería-de-Control-Moderna-Ogata-5ed.pdf.³⁶

3.13.1.2 Segundo método.

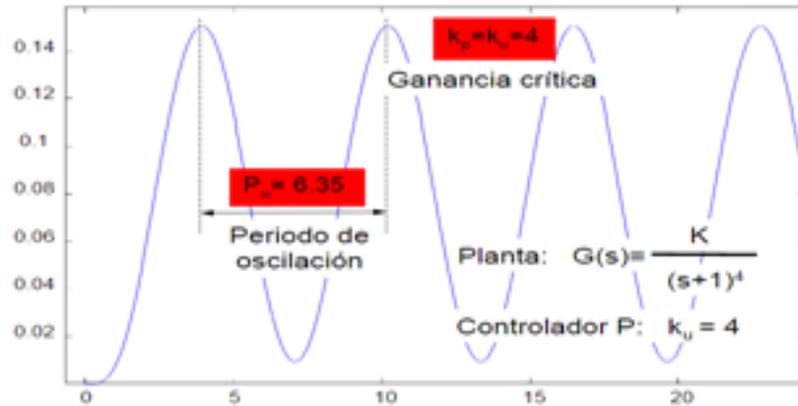
Se utiliza para sistemas que pueden tener oscilaciones sostenidas. Primero se eliminan los efectos de la parte integral y derivativa. Después, utilizando solo la ganancia k_p , haga que el sistema tenga oscilaciones sostenidas. El valor de

³⁵ <http://slideplayer.es/slide/3079472/>

³⁶ Ogata, Benjamí, C Kuo. Ingeniería de control moderno, (2010).

ganancia con que se logre esto se llama ganancia crítica k_{cr} , que corresponde a un periodo crítico P_{cr} , figura 40. Ingeniería-de-Control-Moderna-Ogata-5ed.pdf

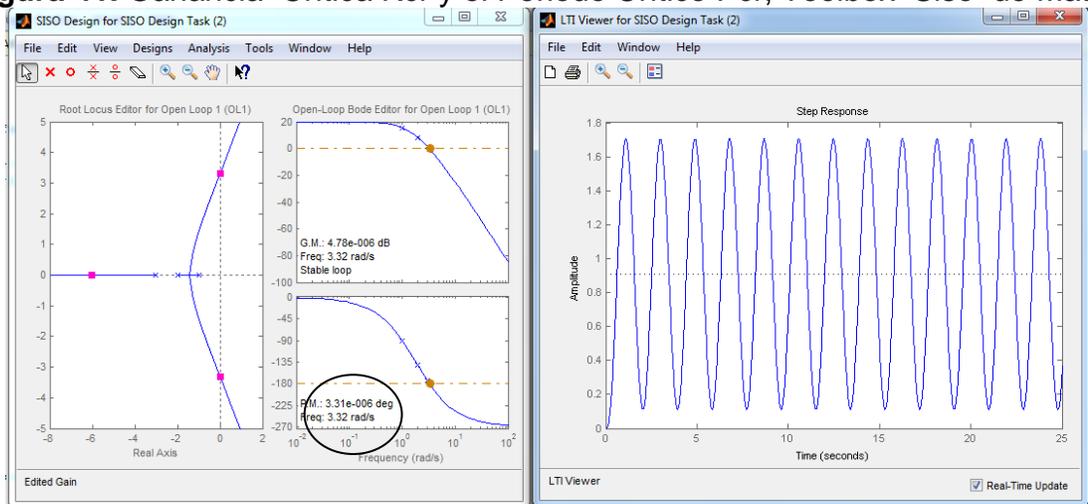
Figura 40. Oscilación Sostenida.



Fuente: <http://slideplayer.es/slide/3079472/> .³⁷

Toolbox Ident para hallar la ganancia crítica y el periodo crítico método 2 de ZN:
Para hallar la ganancia crítica (K_{cr}) y el periodo crítico (P_{cr}), se utiliza el toolbox SISOTOOL de Matlab, a través del cual se ajusta la ganancia (K_p) hasta que la señal de salida del sistema se vuelva oscilante figura 41.

Figura 41. Ganancia Crítica K_{cr} y el Periodo Crítico P_{cr} , Toolbox Siso de Matlab



Fuente: Elaboración propia.

³⁷ <http://slideplayer.es/slide/3079472/>

Con los valores de k_{cr} y P_{cr} se calculan las ganancias del controlador PID, utilizando la tabla 9.

Tabla 9. Valores de Sintonización ZN método 2.

| Tipo de controlador | K_p | T_i | T_d |
|---------------------|--------------|-----------------------|---------------|
| P | $0.5K_{cr}$ | ∞ | 0 |
| PI | $0.45K_{cr}$ | $\frac{1}{1.2}P_{cr}$ | 0 |
| PID | $0.6K_{cr}$ | $0.5P_{cr}$ | $0.125P_{cr}$ |

Fuente: Ingenieria-de-Control-Moderna-Ogata-5ed.pdf.³⁸

3.14. FUNCIÓN DE TRASFERENCIA DEL CONTROLADOR DISCRETO.

Las ganancias de control **PID discreto** se adquieren a partir de las constantes **Kp**, **Ti** y **Td** obtenidas por el método 2 de ZN.

$$u(z) = Kp + \frac{KI}{1 - z^{-1}} + KD(1 - z^{-1})$$

Dónde:

$$KP = Kp - \frac{Kp(Ts)}{2(Ti)}$$

$$KI = \frac{Kp(Ts)}{Ti}$$

$$KD = \frac{Kp(Td)}{Ts}$$

3.14.1 Ecuaciones En Diferencia Algoritmo Del Controlador PID.

Frecuentemente se han empleado métodos de transformación para simplificar el análisis y síntesis de sistemas gobernados por ecuaciones diferenciales o en diferencias. La transformada Z es una regla por la cual una secuencia de números son convertidos a una función de la variable compleja z. Debido a su estructura

³⁸ Ogata, Benjamín C Kuo. Ingeniería de control moderno, (2010).

básica, la transformada Z posee propiedades que facilitan la solución de ecuaciones en diferencias lineales usando simplemente manipulaciones algebraicas.

3.14.1.1 Acción proporcional.

$$\frac{P(z)}{E(z)} = KP$$

Despejando $P(z)$

$$P(z) = KP * E(z)$$

$$P(k) = KP * E(k)$$

3.14.1.2 Acción integral.

$$\frac{I(z)}{E(z)} = \frac{KI}{1 - z^{-1}}$$

Ahora se procede a despejar $I(z)$

$$I(z)(1 - z^{-1}) = E(z) * KI$$

$$I(z) - I(z^{-1}) = E(z) * KI$$

$$I(k) = KI * E(k) + I(k - 1)$$

3.14.1.3 Acción derivativa.

$$\frac{D(z)}{E(z)} = KD \frac{1 - z^{-1}}{1}$$

Despejando $D(z)$ tenemos que:

$$D(z) = (KD - KD(z^{-1}))E(z)$$

Aplicando la ley distributiva de la suma nos queda que:

$$D(z) = KD E(z) - KD E(z^{-1})$$

$$D(k) = KD E(k) - KD E(k - 1)$$

Factorizamos KD

$$D(k) = KD(E(k) - E(k - 1))$$

3.14.1.4 Acción de control discreto.

La acción de control discreta $U(z)$, se obtiene sumando las ganancias KP, KI, KD obtenidas a partir de las ecuaciones en diferencia de cada una de ellas.

$$U(z) = KP * E(k) + (KI * E(k) + I(k - 1)) + KD * (E(k) - E(k - 1))$$

Dónde:

$E(k)$ = Error actual

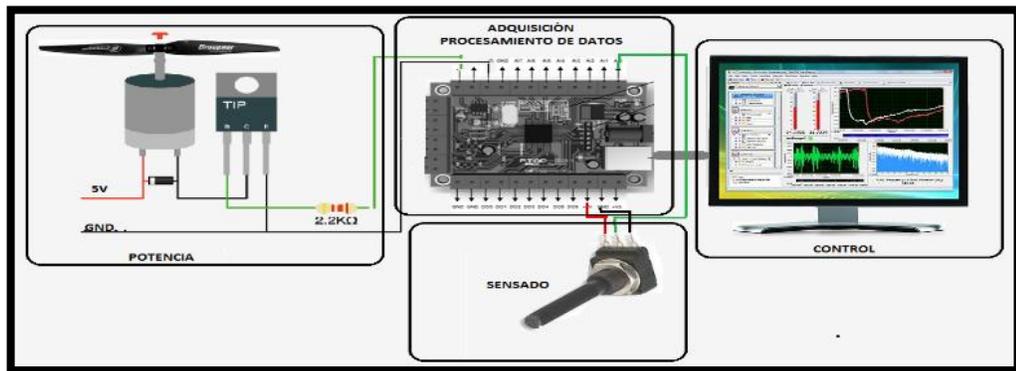
$E(k - 1)$ = Error anterior

$I(k - 1)$ = Acción integral anterior

4. DESARROLLO DEL PROYECTO

El sistema implementado está conformado por tres partes, una interfaz gráfica, una parte eléctrica y una mecánica, estas tres partes en conjunto forman la planta. Esta planta tiene un solo grado de libertad y deberá responder de manera estable a los valores consignados por las variables de control y las referencias deseadas. Ver figura 42.

Figura 42. Representación Gráfica del Sistema Implementado.



Fuente: Elaboración propia.

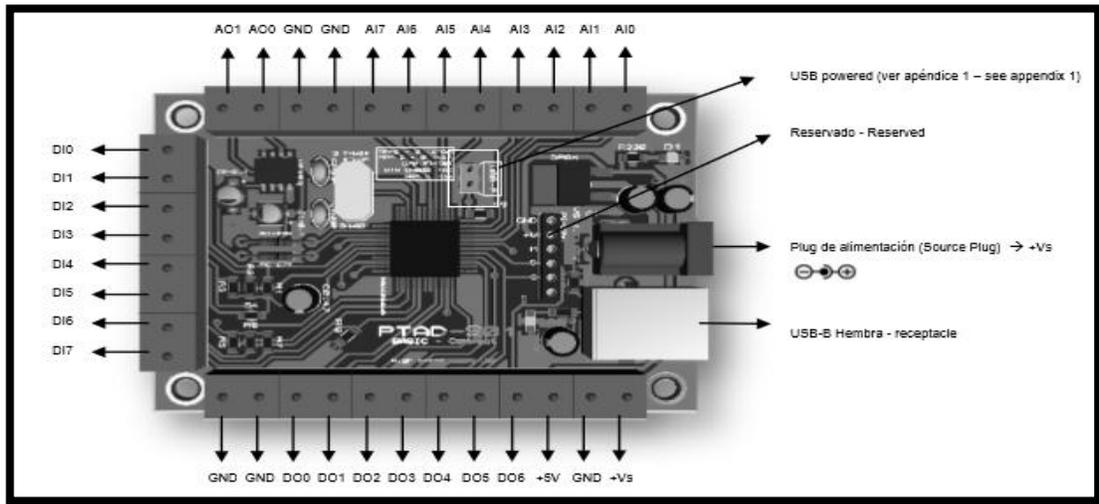
El prototipo se caracteriza por ser una herramienta didáctica ya que posee diferentes puntos de prueba en los modelos de sistemas de control de primer y segundo orden, de esta forma se permite la verificación de las señales de entrada, salida y alimentación.

El sistema propuesto permite analizar el comportamiento del balancín motor hélice; la señal de entrada es una señal paso con amplitud variable por el usuario, la cual generara una señal de salida correspondiente al tipo de sistema.

4.1 ACONDICIONAMIENTO DE SEÑAL.

Para el acondicionamiento de señal normalmente se utilizan circuito Electrónicos que nos ofrecen una señal apta para ser representada o registrada y que posteriormente se pueda procesar, para nuestro sistema se utilizó la tarjeta de adquisición PTAD-S01 y un instrumento virtual (Vi) diseñado en LabVIEW, el cual nos da la facilidad de hacer todos esos procesos gracias a la compatibilidad entre estos dos sistemas figura 43.

Figura 43. Tarjeta de Adquisición PTAD-S01.



Fuente: Elaboración propia.

4.1.1 ESCALIZACIÓN Y ACONDICIONAMIENTO DE SEÑAL

Lectura del sensor: Para determinar la linealidad del sensor se optó por comparar el movimiento del balancín y su posición sobre una medida que se hace a través de un transportador ubicado en el bastidor, figura 44.

Figura 44. Sistema Mecánico de Visualización de Posición.



Fuente: Elaboración propia.

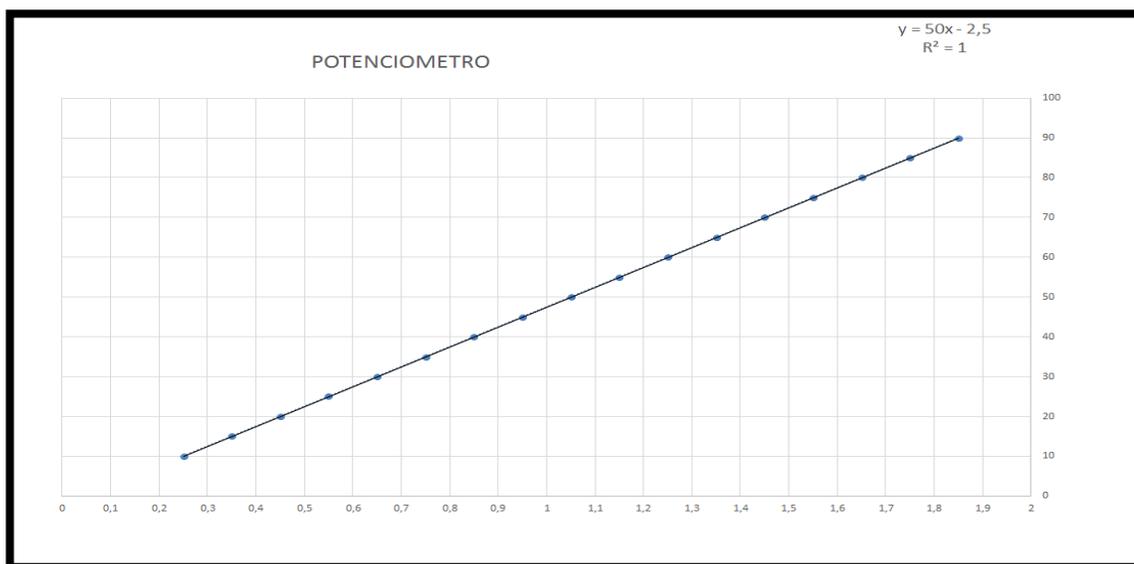
Las lecturas del sensor con respecto al Angulo de inclinación se hicieron en el laboratorio, las lecturas fueron tomadas cada 5 grados de inclinación, ver tabla 10. Una vez obtenidos los datos y registrados en una hoja de cálculo se procedió a graficar la relación de las dos medidas para poder así determinar su linealidad, grafica 1.

Tabla 10. Lecturas del Sensor Tomadas Cada 5 Grados de Inclinación

| VOLT SALIDA SENSOR | ANGULO INCLINACION |
|--------------------|--------------------|
| 0,25 | 10º |
| 0,35 | 15º |
| 0,45 | 20º |
| 0,55 | 25º |
| 0,65 | 30º |
| 0,75 | 35º |
| 0,85 | 40º |
| 0,95 | 45º |
| 1,05 | 50º |
| 1,15 | 55º |
| 1,25 | 60º |
| 1,35 | 65º |
| 1,45 | 70º |
| 1,55 | 75º |
| 1,65 | 80º |
| 1,75 | 85º |
| 1,85 | 90º |

Fuente: Elaboración propia.

Gráfica 1. Representación Gráfica de los Datos y su Respectiva Ecuación de la Recta.



Fuente: Elaboración propia.

4.1.2 Conversión De Voltaje A Grados.

Para la conversión de voltaje del sensor utilizamos la ecuación de la recta que fue generada a través de la herramienta línea de tendencia de Excel opción “representar la ecuación del grafico” como lo muestra la gráfica 1. Donde el valor

de salida del sensor (voltaje) se convierte en la posición en grados del balancín así:

$$y = 50 * x - 2.5$$

Dónde:

$x = \text{voltaje del sensor}$

$y = \text{Angulo de inclinacion del balancin}$

Ejemplo:

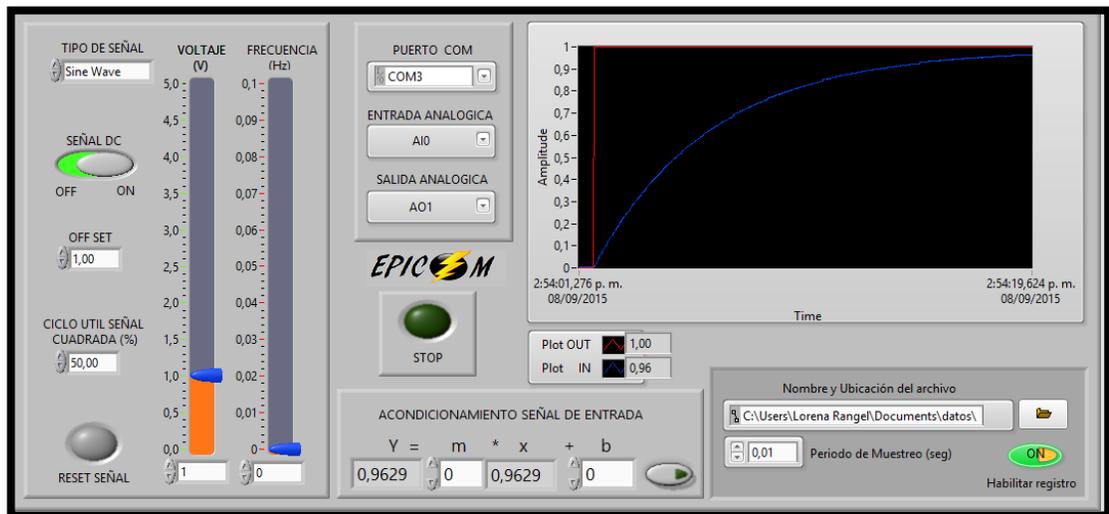
Para $x = 1.35 \text{ Vol}$

$$y = 50 * 1.35 - 2.5 = 65^\circ$$

4.2 DISEÑO DEL INSTRUMENTO VIRTUAL PARA EL ACONDICIONAMIENTO DE SEÑAL Y REGISTRO DE DATOS.

El instrumento virtual para la adquisición, acondicionamiento y registro de datos del sistema está conformado por un panel frontal de instrumentos y un bloque de funciones desarrollados en LabVIEW 2011, donde se incorpora en un lenguaje gráfico, el algoritmo encargado de registrar y almacenar los valores tanto de entrada $\square(\square)$, como de salida $\square(\square)$ de la planta, tal como se muestra en la figura 45.

Figura 45. Panel Frontal y Bloque de Funciones para la Adquisición y Registro de Datos

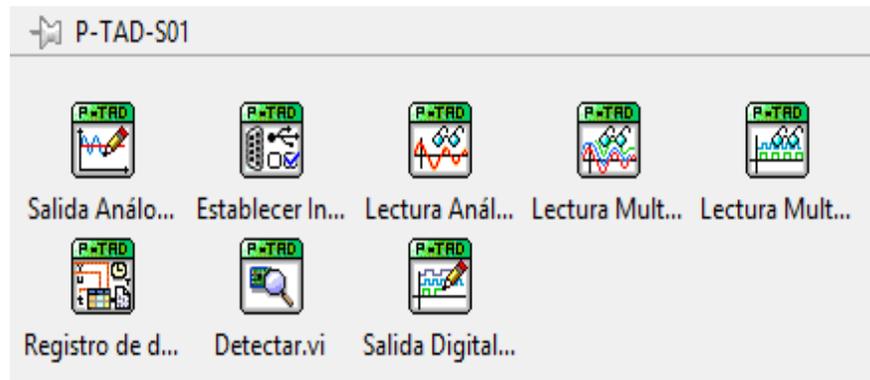


Fuente: Elaboración propia.

Este instrumento es conformado por bloques especiales soportados para LabVIEW 2011, figura 46, llamados SubVI, algunos bloque tienen como funciones especiales las de establecer la comunicación entre la tarjeta ADQ y la PC, hacer la conversión análogo digital (A/D) de las entradas y la conversión digital análogo

(D/A) de las salidas de la tarjeta ADQ. Cuenta con un bloque especial de registro de datos donde se configura la ruta para el almacenamiento de la información generada por la entrada y salida de la planta para posteriormente ser procesados en Matlab.

Figura 46. Panel Frontal y Bloque de Funciones para la Adquisición y Registro de Datos



Fuente: Elaboración propia.

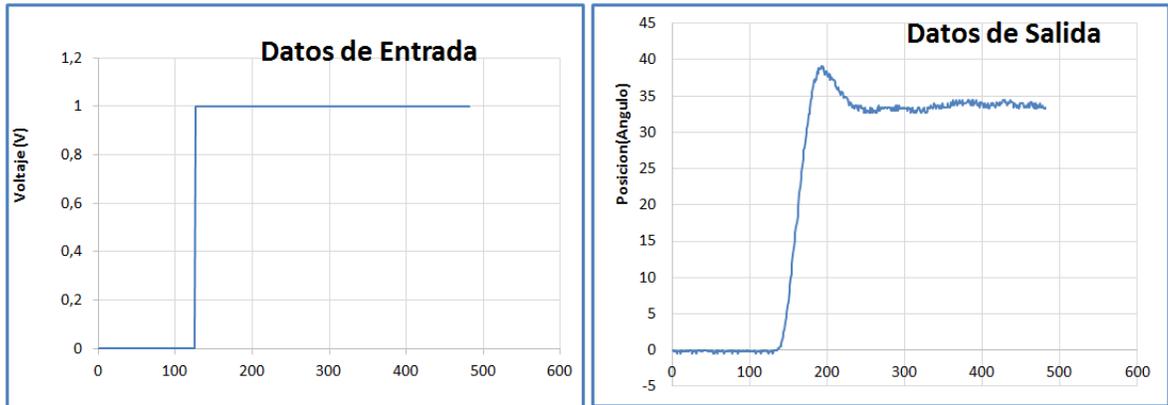
4.3 IDENTIFICACIÓN DEL MODELO

La identificación no paramétrica de nuestro modelo se realizara en lazo abierto, donde se aplica una entrada paso al sistema de 1 volt, el cual generara una respuesta a su salida, estos datos generados son registrados y almacenados para posteriormente ser procesados.

Los datos almacenados con extensión .txt, se pueden exportar a Excel para ser graficados, esto se hace con el fin de determinar el comportamiento del sistema y verificar el orden de la señal (1, 2 o superior), grafica 2.

Como se observa en el gráfico, la señal de orden 2 sub amortiguada con retardo, estos datos son fundamentales ya que IDENT los necesita para realice una identificación lo más precisa.

Gráfica 2. Gráficos en Excel Respuesta del Sistema a una Entrada Escalón

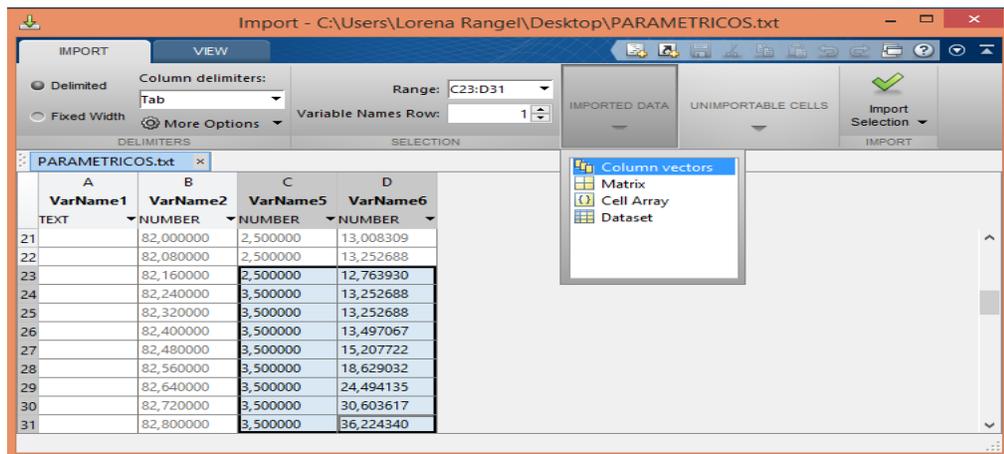


Fuente: Elaboración propia.

4.4 PROCESAMIENTO DE DATOS Y VALIDACIÓN DEL MODELO

Una vez obtenidos los resultados de la prueba, se procede a utilizar el TOOLBOX IMPORT de Matlab para la importación de los datos. Los datos de entrada y salida serán guardados como vector columna, figura 47.

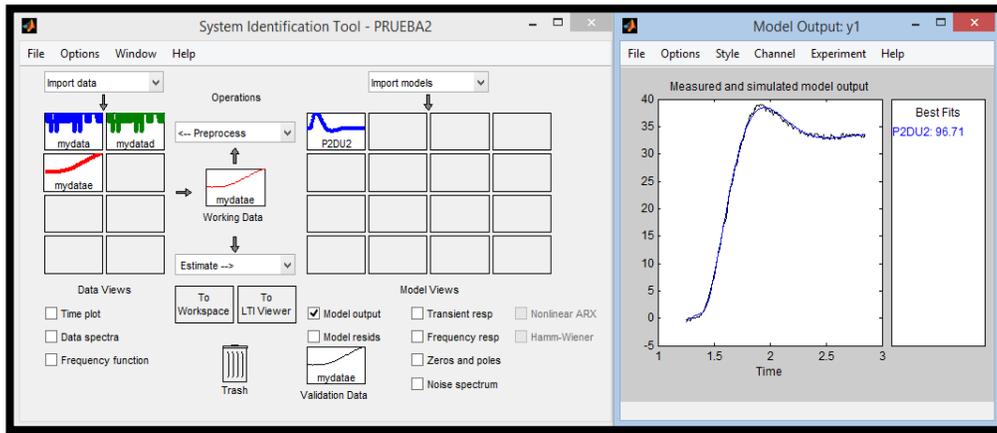
Figura 47. Toolbox Import de Matlab.



Fuente: Elaboración propia.

Con los datos dentro de una variable se procede a identificar el modelo de nuestro proceso o sistema con la ayuda del toolbox IDENT de Matlab, esta herramienta nos da la facilidad de procesar los datos y arrojar un modelo matemático estimado de nuestro proceso, Figura 48.

Figura 48.Estimación y validación del modelo con IDENT de Matla}



Fuente: Elaboración propia.

Los datos arrojados por el IDENT nos da una estimación del 96.71% de efectividad lo cual es suficiente para dar como identificado y validado nuestro modelo, los datos procesados son enviados al “WorkSpace” de Matlab para obtener la función de transferencia estimada del modelo.

4.5 FUNCIÓN DE TRASFERENCIA DEL PROCESO

La variable con la información de nuestro modelo (P2DU2) es exportada al Command Windows de Matlab a través de IDENT. Una vez obtenido la variable con los datos del modelo se Aplica el siguiente comando, para obtener la función de transferencia de nuestro sistema.

$$G = tf(P2DU2)$$

Función de transferencia del sistema

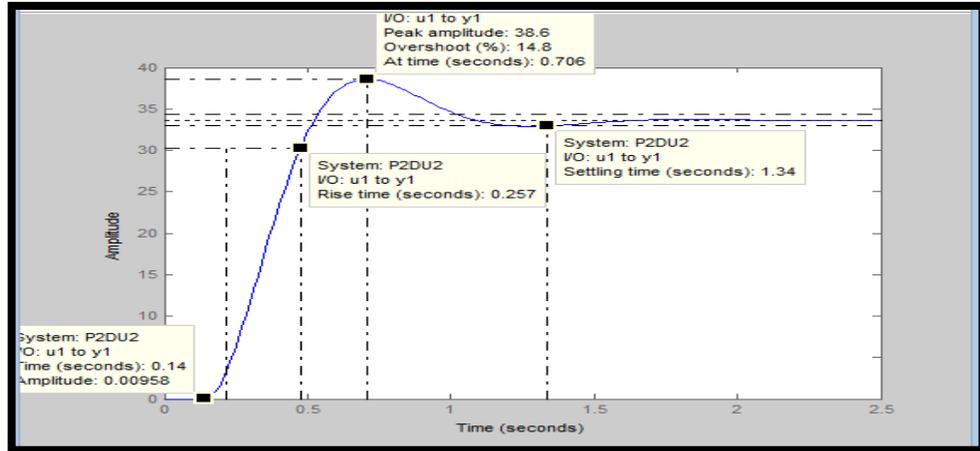
$$G(s) = \frac{33.61}{0.02354s^2 + 0.1596s + 1} e^{-0.142s}$$

La variable (G) contiene los datos de nuestro modelo. Con la siguiente instrucción graficamos la respuesta de nuestro modelo ante una entrada escalón unitario en lazo abierto, con el fin de hallar el Rise time y así determinar el tiempo de muestreo.

```
Num=[33.61]; den=[0.02354 0.1596 1];
td=0.142; G=tf(num,den);
G.outputdelay=td;
```

step(G)

Gráfica 3. Respuesta del Sistema a una Entrada Escalón Unitario



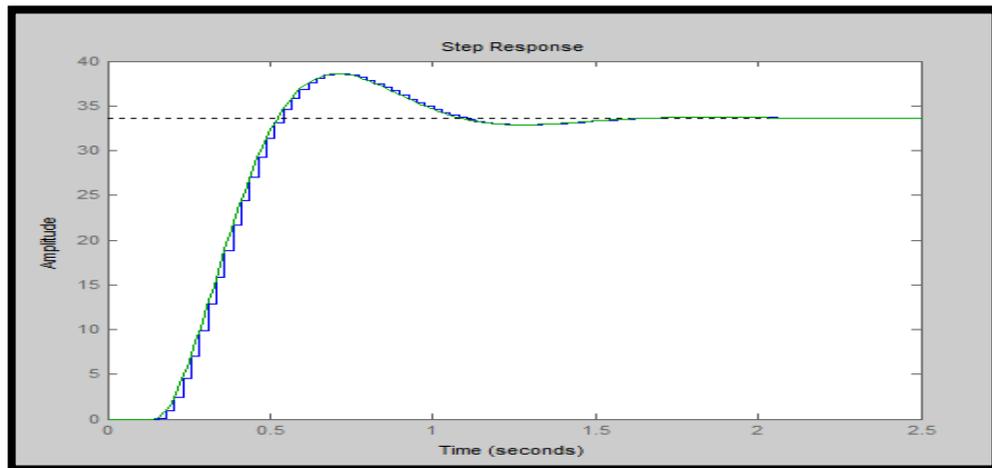
Fuente: Elaboración propia.

En la gráfica 3, Se evidencia el valor del Rise time el cual es de 0.257 seg. Este valor lo dividido por 10 para obtener un Sampling time ($St=0.0257$).

Con la siguiente instrucción graficamos la respuesta discreta de nuestro modelo

```
St=0.0257;           // Tiempo de muestreo  
H=c2d(G,st,'zoh')   // Convierte la señal de continuo a discreto  
step(G,H)           // se grafican las 2 señales
```

Gráfica 4. Discretización del señal, para un $T_s=0.0257$



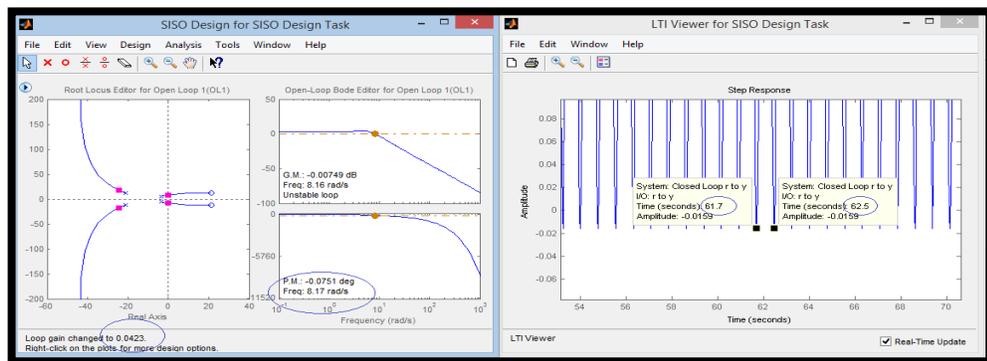
Fuente: Elaboración propia.

En la gráfica 4, se puede notar que la señal discreta tiene una gran similitud con la señal continua, esto nos garantiza que el tiempo de muestreo hallado cumple con los requerimientos.

4.6 DISEÑO DEL CONTROLADOR PID

Como vimos en la gráfica 3, la señal de respuesta del sistema es oscilante, y se procede a aplicar el método 2 de sintonización de Zeigler-Nichols. Donde aplicamos una ganancia hasta que el sistema tenga oscilaciones permanentes, esto con el fin de determinar los valores de la ganancia crítica y el periodo crítico; para este proceso se utiliza el toolbox SISOTOOL de Matlab, figura 49.

Figura 48. Toolbox SISO de Matlab



Fuente: Elaboración propia.

En la figura 49, A la izquierda se puede apreciar encerrados con un círculo azul los valores que toman la Ganancia crítica (K_{cr}) y la velocidad angular (w) al aumentar la ganancia del sistema, a la derecha el periodo critico (P_{cr}).

Dónde: $K_{cr} = 0.0423$ $w = 8.17 \text{ rad/s}$

Si tenemos en cuenta que: $P_{cr} = T = \frac{1}{f}$

Y que:

$$f = \frac{w}{2\pi} = \frac{8.17 \text{ rad/s}}{2\pi} = 1.3 \text{ Hz}$$

Entonces: $P_{cr} = \frac{1}{f} = \frac{1}{1.3} = 0.769 \text{ seg}$

4.6.1 Sintonización Del Controlador.

Obtenidos los valores de $k_{cr} = 0.0423$; $P_{cr} = 0.769 \text{seg}$ se hallan las ganancias del controlador PID en tiempo continuo aplicando el segundo método de ZN.

Dónde:

$$K_p = 0.6(K_{cr}) = 0.6(0.0423)$$

$$\mathbf{K_p = 0.0254}$$

$$T_i = 0.5(P_{cr}) = 0.5(0.769)$$

$$\mathbf{T_i = 0.3845}$$

$$T_d = 0.125(0.769)$$

$$\mathbf{T_d = 0.0961}$$

4.6.2 Conversión Del Controlador Continúo A Discreto.

Una vez obtenido las ganancias del controlador en tiempo continuo se procede a convertir estos valores a tiempo discreto aplicando las siguientes ecuaciones.

Para un tiempo de muestreo $t_s = 0.0257$

$$K_P = K_p - \frac{K_p(T_s)}{2(T_i)} = 0.025 - \frac{0.0254(0.0257)}{2(0.3845)}$$

$$\mathbf{K_P = 0.0245}$$

$$K_I = \frac{K_p(T_s)}{T_i} = \frac{0.0254(0.0257)}{0.3845} b$$

$$\mathbf{K_I = 0.0017}$$

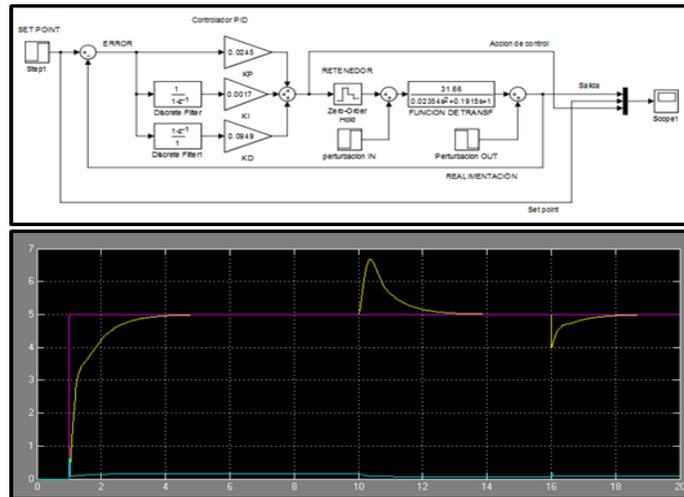
$$K_D = \frac{K_p(T_d)}{T_s} = \frac{0.0254(0.0961)}{0.0257}$$

$$\mathbf{K_D = 0.0949}$$

4.6.3 Simulación En Lazo Cerrado Del Modelo Obtenido.

Para la simulación del modelo en lazo cerrado se utiliza la herramienta Simulink, añadiremos el controlador PID de posición con los valores de las ganancias discretas y a la vez unas perturbaciones tanto a la entrada como a la salida para ver su comportamiento, figura 50.

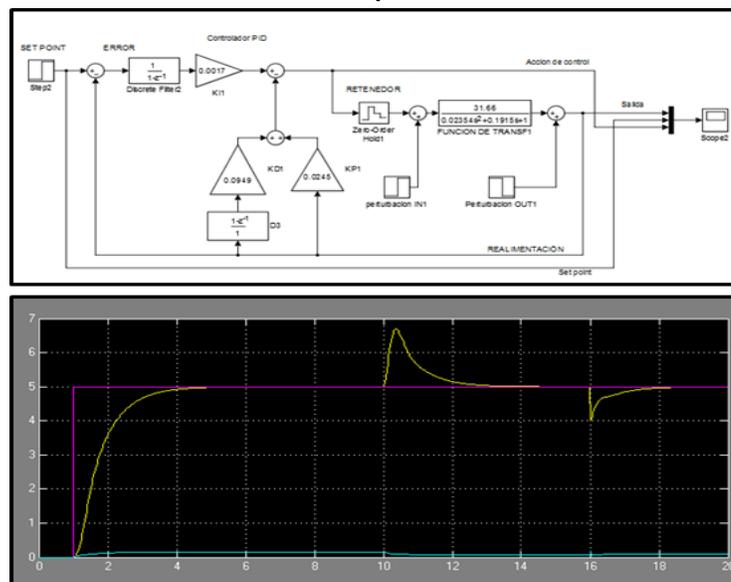
Figura 49. Lazo de Control y Respuesta del Sistema Simulink.



Fuente: Elaboración propia.

La figura 50, muestra el comportamiento dinámico del sistema en lazo cerrado con un controlador PID discreto de posición. La respuesta temporal nos muestra que el error en estado estacionario es 0, no tiene sobre impulso y el tiempo de establecimiento es de 2 seg y como se puede notar tiene rechazo a perturbaciones.

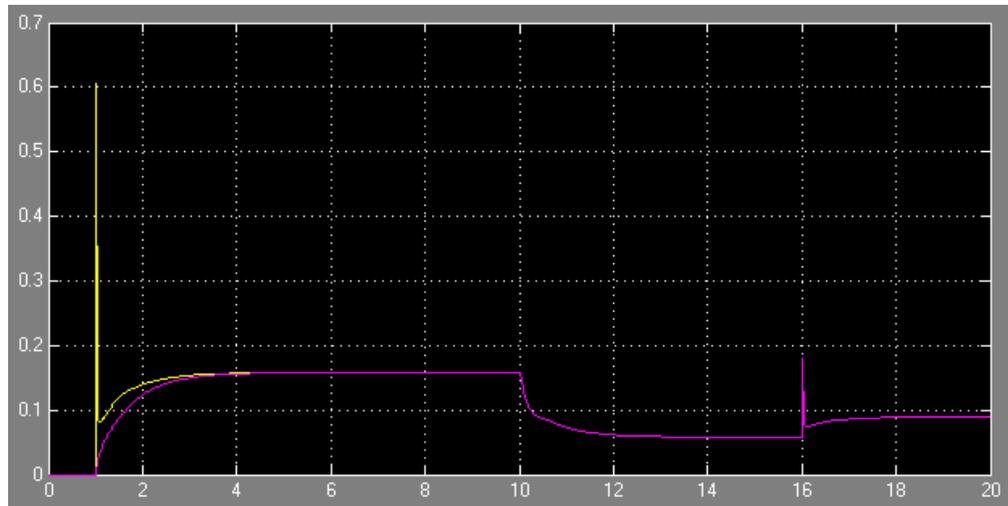
Figura 50. Lazo de Control y Respuesta del Sistema Simulink.



Fuente: Elaboración propia

La figura 51. Muestra el comportamiento dinámico del sistema en lazo cerrado con un controlador PID discreto de Velocidad. La respuesta temporal de sistemas nos muestra, que tiene las mismas características del controlador PID de posición. La diferencia se encuentra en la acción de control, la cual es menos brusca, esto hace que la respuesta del sistema tenga forma de "S".

Figura 51. Acción de Control de Posición y Velocidad



Fuente: Elaboración propia.

En la figura 52, se evidencia el comportamiento de las dos acciones de control. En amarillo la acción de control de posición y en violeta la acción de control de velocidad.

4.7 ECUACIÓN EN DIFERENCIA DE LA ACCIÓN DE CONTROL DE POSICIÓN

Aplicando ecuaciones en diferencia del lazo de control y Reemplazando las ganancias por sus valores, la acción de control discreta queda:

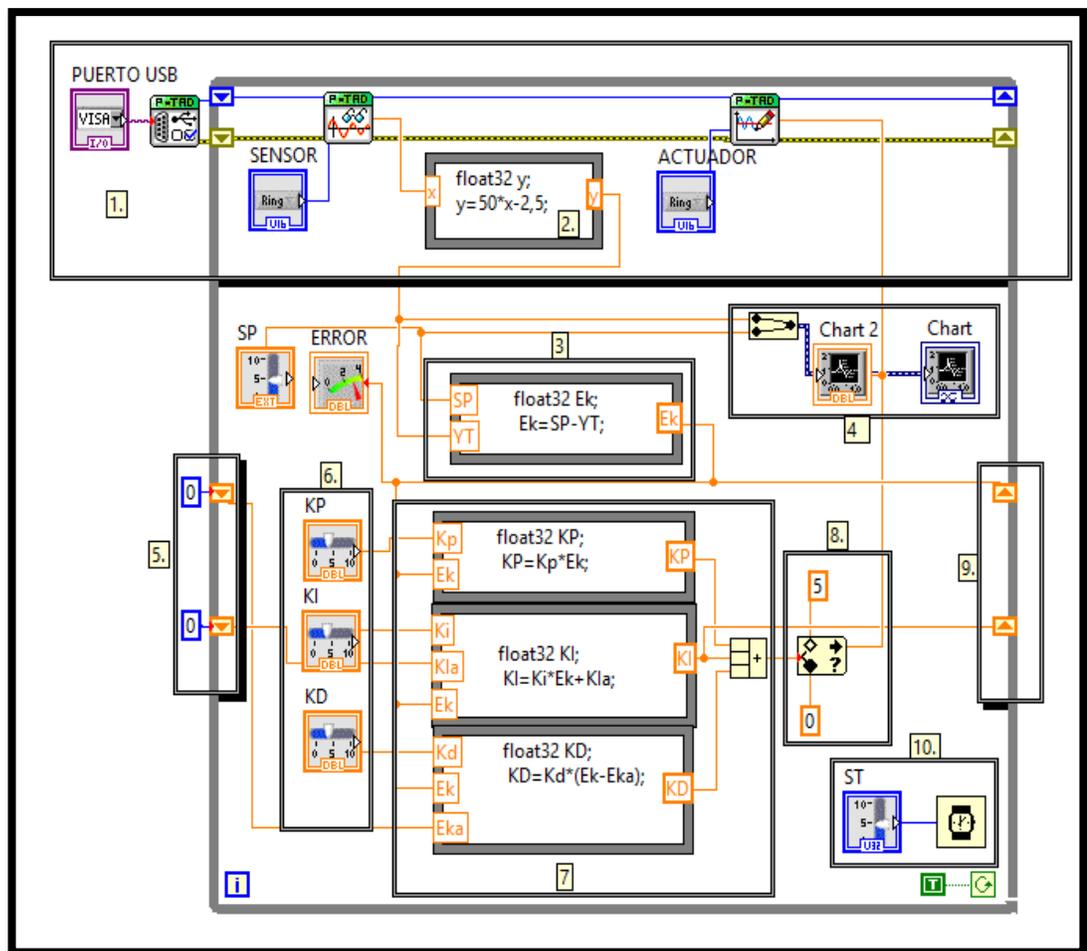
$$U(k) = 0.0245 * E(k) + (0.007 * E(k) + I(k - 1)) + 0.0949 * (E(k) - E(k - 1))$$

5. DISEÑO DE LA INTERFAZ GRÁFICA PARA EL MONITOREO Y CONTROL DEL SISTEMA

5.1 BLOQUE DE FUNCIONES

Para el control del sistema se procede a implementar el algoritmo de control compuesto principalmente por un ciclo while el cual contiene internamente una serie de etapas figura 53.

Figura 52. Bloque de Funciones, Algoritmo de Control. LabVIEW 2011



Fuente: Elaboración propia.

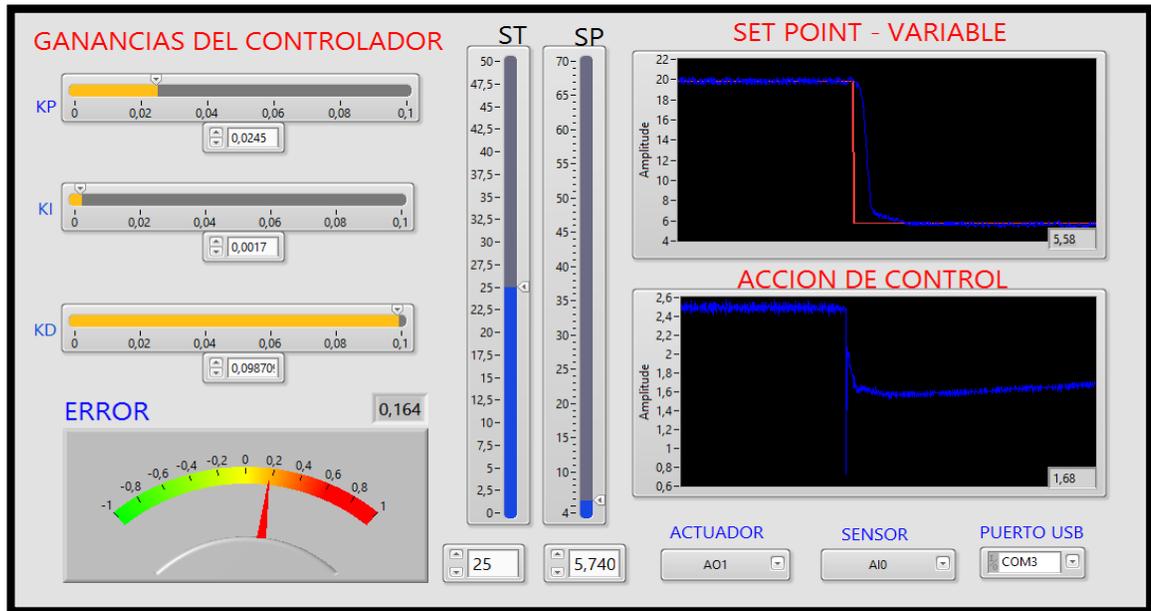
Como se muestra en la figura 53, cada etapa tiene asignado un número, este número representa una función dentro del programa. A continuación se describe cada una de estas funciones que hacen parte del algoritmo de control.

1. *Interfaz de comunicación puerto serial (USB), bloques de funciones de lectura y escritura análoga, configurable por el usuario.*
2. *NODO FORMULA, se implementa para representar la posición actual del balancín, Convirtiendo el voltaje del sensor en el ángulo de inclinación.*
3. *NODO FORMULA, implementado para hallar el error actual ($E(k)$).*
4. *Este bloque lo componen dos INDICADORES GRAFICOS. Utilizados para visualizar el comportamiento de la variable y la acción del controlador.*
5. *SHIFT REGISTER. Asignado al bloque While loop, es implementado para retener los valores pasados del error $E(k-1)$ y la acción integral $KI(k-1)$.*
6. *CONTROLES TIPO SLIDER, hacen parte del panel frontal, encargados de asignar las ganancias al controlador PID discreto.*
7. *Bloque compuesto por tres NODOS FORMULA Y un bloque COUNPOUND, este último es el encargado de sumar las ganancias KP , KI y KD para obtener la acción de control $U(t)$.*
8. *BLOQUE IN RANGE AND COERCE FUNCTION, limita el nivel máximo y mínimo de la acción de control.*
9. *SHIFT REGISTER. Asignado al bloque While loop, se implementan para retener los valores actuales del error $E(k)$ y la acción integral $KI(k)$.*
10. *BLOQUE TIEMPO DE MUESTREO, es el encargado de hacer el retardo del WHILE LOOP para la toma de muestras, lo compone un control tipo Slider y un bloque Wait, configurable desde el panel frontal.*

5.2 PANEL FRONTAL

El panel frontal es el complemento del bloque de funciones, es el encargado de suministrar la información necesaria al usuario del comportamiento del proceso, aquí se asignan todos los instrumentos necesarios para llevar a cabo el control de la variable, figura 54.

Figura 53. Panel frontal de supervisión y control del proceso. LabVIEW 2011



Fuente: Elaboración propia.

Compuesto principalmente por indicadores gráficos y controles tipo slider. En él se ajustan las ganancias del controlador (KP, KI, KD), el tiempo de muestreo (ST), asignar el SET POINT (SP), configurar el canal de entrada análoga (A0-A7), salida análoga (A0-A1) y el puerto de comunicación (COM).

6. DISEÑO DE LA INTERFAZ Y ALGORITMO DE CONTROL (SIMULINK-ARDUINO)

6.1 ACONDICIONAMIENTO DE SEÑAL SIMULINK

Las lecturas del sensor con respecto al Angulo de inclinación fueron tomadas cada 5 grados, esto con el fin de convertir el voltaje generado por el sensor en el ángulo de inclinación del balancín. Los datos generados fueron graficados teniendo como resultado la siguiente ecuación:

$$Y = X * 67.5 - 3.7$$

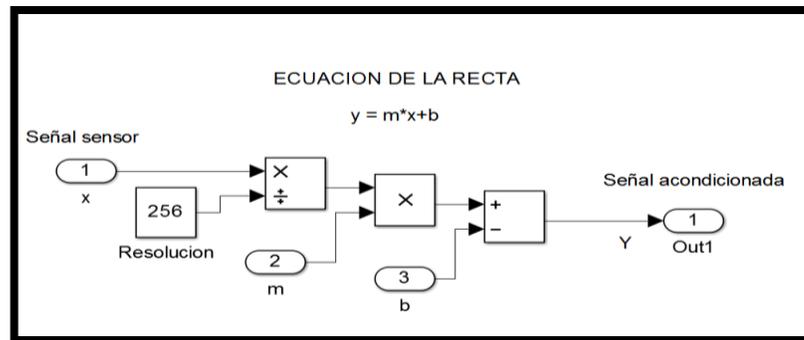
Donde

X= señal del sensor

Y= el ángulo de inclinación del balancín

Ahora se implementa en Simulink esta ecuación por medio de bloques donde la señal del sensor es escalizada antes de entrar al acondicionamiento, ver figura 55.

Figura 54. Acondicionamiento de señal, bloques Simulink

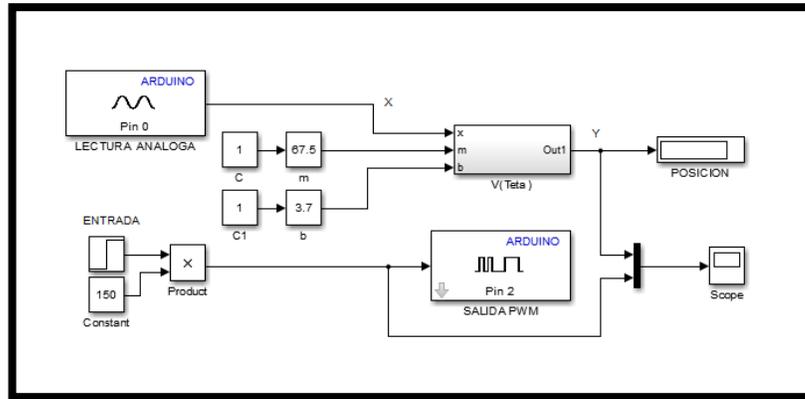


Fuente: Elaboración propia.

6.2 ADQUISICIÓN Y REGISTRO DE DATOS SIMULINK

Para el registro de los datos excitaremos la planta a través de una señal PWM generada por el bloque PWM pin 2 de la tarjeta Arduino mega 2560 el cual tiene un rango de trabajo entre 0 y 256 que hacen referencia al 0 y el 100% del ciclo útil (Duty Cycle) de la señal PWM, la respuesta del sistema es captada por medio del bloque lectura analógica el cual tiene configurado el pin analógico 0 de la placa Arduino mega, ver figura 56.

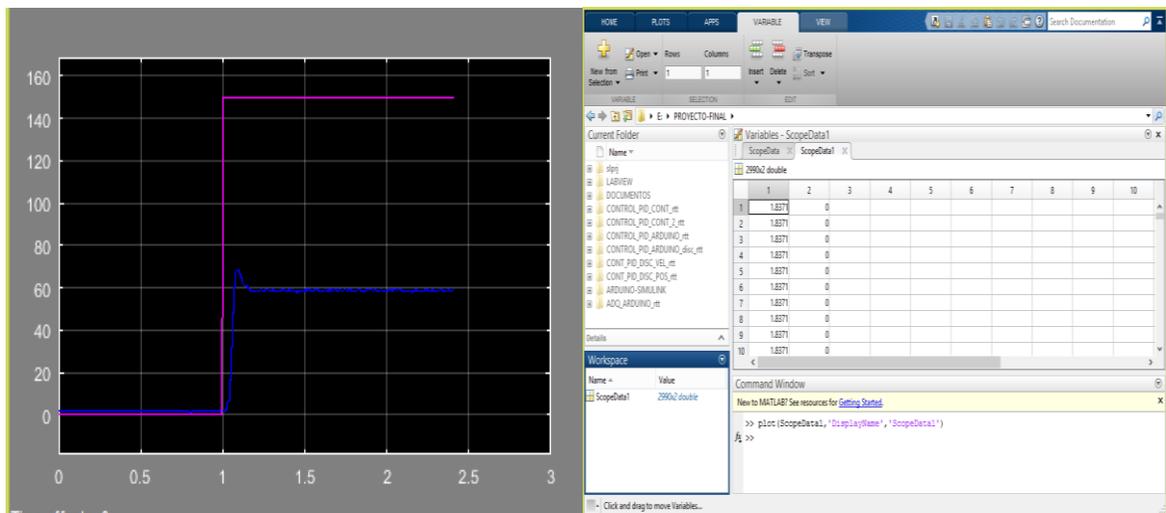
Figura 55. Diagrama de bloques para el registro de datos.



Fuente: Elaboración propia.

El registro de los datos se hace por medio del bloque Scope que además de visualizar las señales de entrada y salida se configura para que los datos sean exportados automáticamente al “Workspace” al finalizar cada simulación, figura 57.

Figura 56. Datos importados al Workspace a través de bloque “Scope” de Simulink.

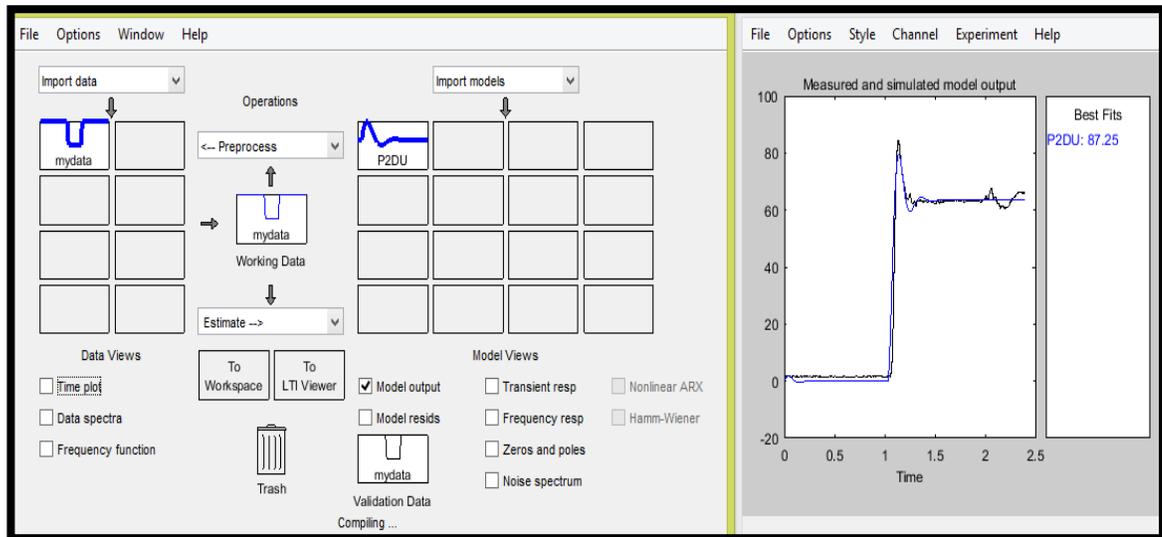


Fuente: Elaboración propia.

6.3 IDENTIFICACIÓN DEL MODELO (FUNCION DE TRANSFERENCIA) IDENT/MATLAB

Una vez se hallan registrado los datos se procede a realizar la identificación de nuestro modelo por medio de la herramienta Ident de Matlab.

Figura 57. Ventana del Ident. Modelo estimado



Fuente: Elaboración propia.

Los datos arrojados por el IDENT nos da una estimación del 87.25% de efectividad lo cual es suficiente para dar como identificado y validado nuestro modelo. La variable que contiene los datos (P2DU) es enviada al "WorkSpace" de Matlab, luego con la instrucción `tf(P2DU)` obtenemos la función de transferencia del modelo, ver figura 59

Figura 58. Ventana del command Windows con la Función de transferencia del modelo

```
>> tf(P2DU)

ans =

    From input "u1" to output "y1":
           0.423
    exp(-0.03*s) * -----
                0.001039 s^2 + 0.02597 s + 1

Name: P2DU
Continuous-time transfer function.

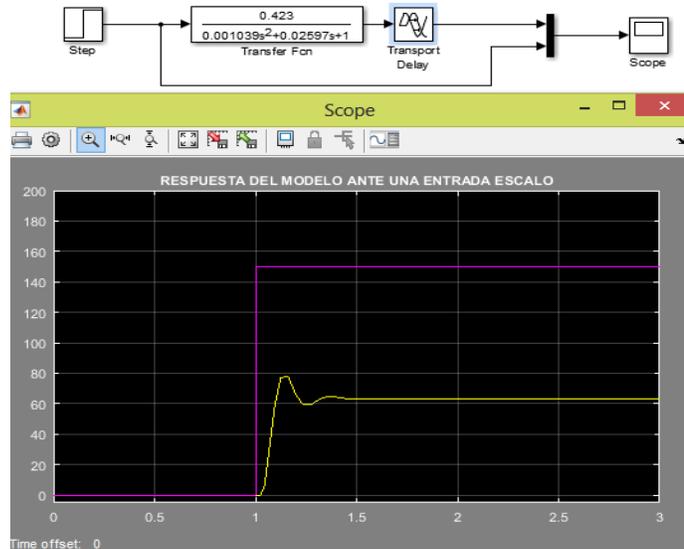
>> step(P2DU)
>> |
```

Fuente: Elaboración propia.

6.4 SIMULACION EN LAZO ABIERTO

Una vez hallada la función de transferencia se procede a simular el comportamiento nuestro modelo implementado un diagrama de bloques en Simulink, donde excitemos la planta con una entrada escalón de 150, ver figura 60.

Figura 59. Respuesta modelo visualización de la señal de entrada y salida.



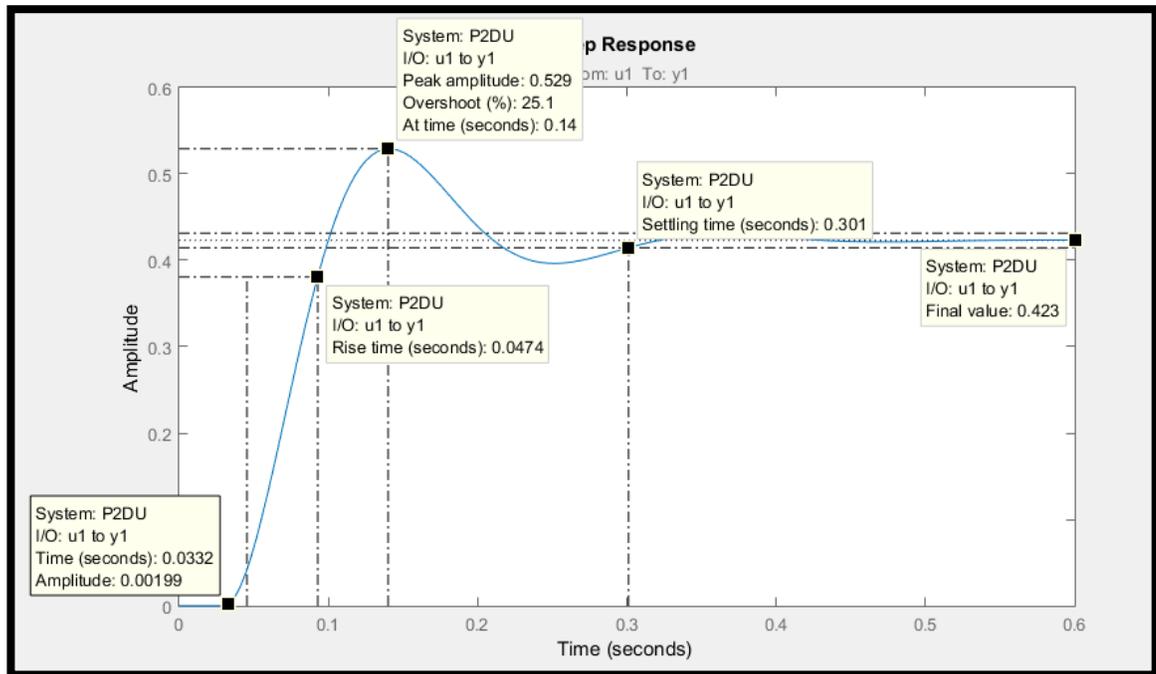
Fuente: Elaboración propia.

En la figura 59, se puede apreciar el comportamiento que tiene la señal ante una entrada escalón de 150 la cual tiende a comportarse de la misma forma que el modelo real percibido en la figura 60.

6.5 CARACTERÍSTICAS DE LA RESPUESTA TEMPORAL DEL MODELO

A través del comando step (P2DU), graficamos la respuesta de nuestro modelo ante una entrada escalón, ver figura 61.

Figura 60. Características de la respuesta temporal del modelo



Fuente: Elaboración propia.

Al analizar la respuesta de del sistema en lazo abierto podemos determinar que la planta tiene las siguientes características:

Tiempo de retardo (Delay time)= 0.03 s.

Tiempo de subida (Rise time) = 0.0474 s,

Sobrepaso (Overshoot)= 25.1%

Tiempo de establecimiento (Settling time)= 0.3.

Error en estado estacionario (ess) = 58%

Tiempo de muestreo (Sampling time) = 0.004 s

6.6 SINTONIZACION METODO ZN

Teniendo en cuenta que la respuesta de nuestro sistema tiene oscilaciones sostenidas se aplica el segundo método de ZN, donde se halla la ganancia crítica K_{cr} y el periodo crítico P_{cr} para aplicar las reglas de la tabla 11.

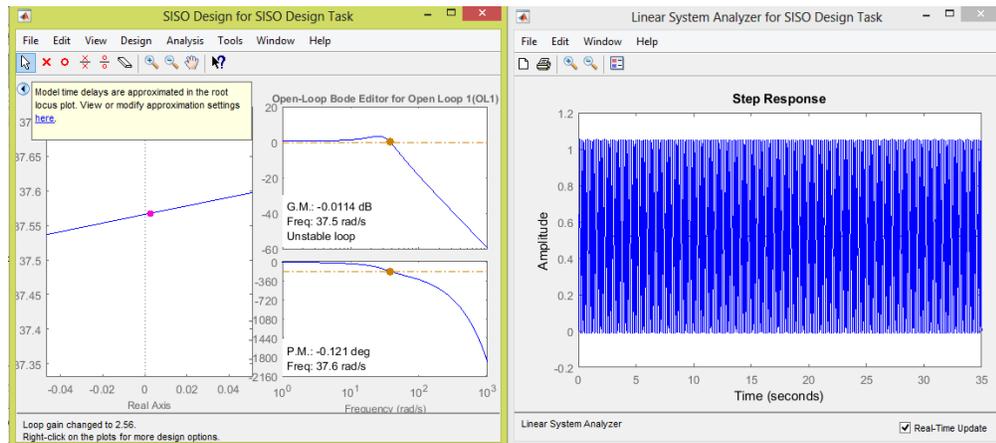
Tabla 11. Reglas de sintonización segundo método ZN

| Tipo de controlador | K_p | T_i | T_d |
|---------------------|--------------|-----------------------|---------------|
| P | $0.5K_{cr}$ | ∞ | 0 |
| PI | $0.45K_{cr}$ | $\frac{1}{1.2}P_{cr}$ | 0 |
| PID | $0.6K_{cr}$ | $0.5P_{cr}$ | $0.125P_{cr}$ |

Fuente: Elaboración propia.

Par hallar la ganancia crítica K_{cr} y el periodo crítico P_{cr} se utiliza el toolbox SISO de Matlab donde aumentamos la ganancia del sistema hasta obtener oscilaciones permanentes, figura 62.

Figura 61. Ventana de la herramienta SISO tool de Matlab



Fuente: Elaboración propia.

Los datos obtenidos a través del SISOTOOL fueron:

$$K_{cr} = 2.56$$

$$P_{cr} = 37.5 \text{ r/s}$$

$$T_{cr} = 0.1675 \text{ s}$$

Para obtener las ganancias del controlador PID se aplicaron las reglas de sintonización de la tabla 1, donde:

$$K_p = 0.6(K_{cr}) = 0.6(2.56)$$

$$K_p = 1.536$$

$$T_i = 0.5(P_{cr}) = 0.5(0.1675)$$

$$T_i = 0.8375$$

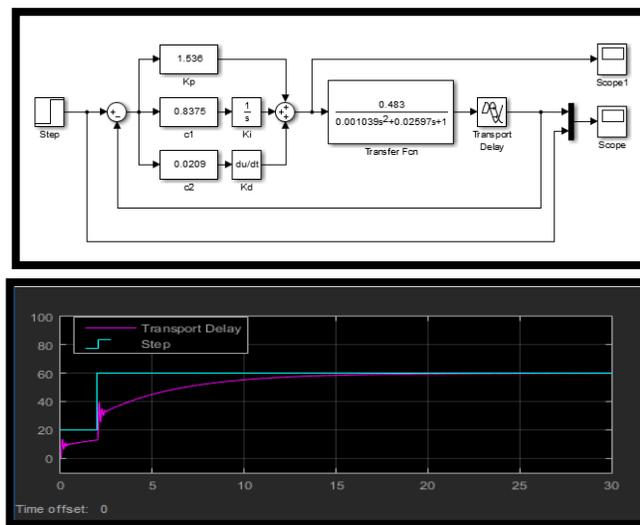
$$T_d = 0.125(0.1675)$$

$$T_d = 0.0209$$

6.7 SIMULACIÓN DEL LAZO DE CONTROL DEL MODELO.

Para la simulación del modelo en lazo cerrado se utiliza la herramienta Simulink, donde añadiremos el controlador PID de posición con los valores de las ganancias continuas obtenidas por el método 2 de ZN, figura 63.

Figura 62. Diagrama de bloques del lazo de control en Simulink

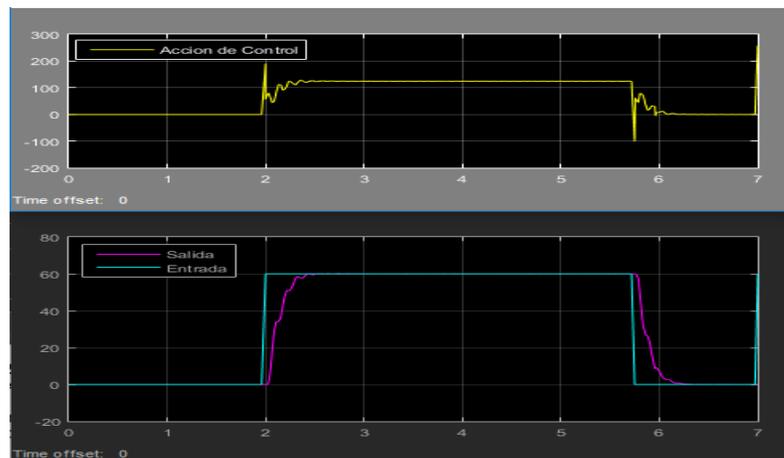


Fuente: Elaboración propia.

Analizando el comportamiento de la señal de la figura 63. Se puede determinar que las ganancias del controlador PID hacen que el sistema responda de manera muy lenta, alcanzando el estado estacionario en un tiempo mayor a los 15 segundos, lo que conlleva a realizar ajuste a los valores de estas ganancias para obtener el comportamiento esperado.

Ahora se procede a ajustar las ganancias del controlador con los siguientes valores: $K_p=0.935$ $K_i = 16.6221$, $K_d= 0.09$

Figura 63. Visualización de la acción de control y la respuesta del sistema



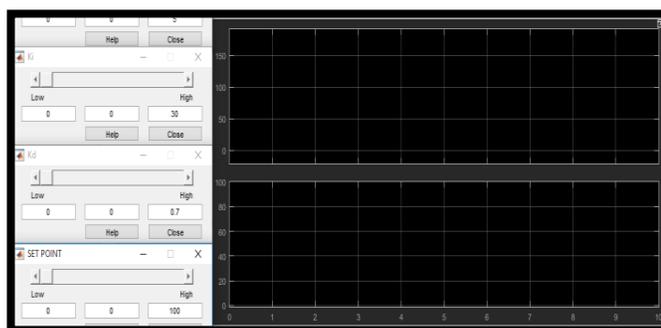
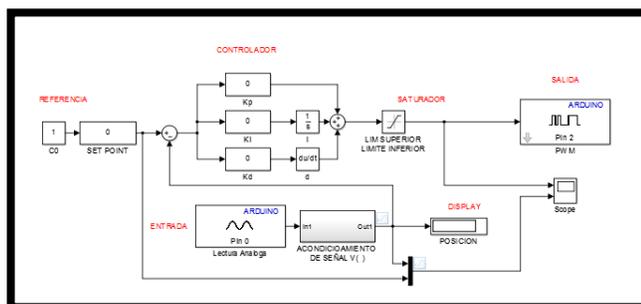
Fuente: Elaboración propia.

En la figura 64, se muestra el comportamiento del sistema simulado donde se han ajustado las ganancias del controlador PID, dando como resultado una respuesta más eficiente a los cambios en la referencia.

6.7 INTERFAZ DE MONITOREO Y CONTROL.

Para la interfaz de monitoreo y control se modifica el diagrama de bloques que implemento para el registro dato, al cual se le asigna al bloque PWM un bloque slider para la configuración del Set Point, los bloques proporcional, integral y derivativo para la acción de control, un bloque de saturación para limitar la acción de control, la señal de realimentación que proviene de la salida del bloque de acondicionamiento, un bloque tipo Display para la visualización de la posición del balancín y un bloque Scope para la visualización de la acción de control y la respuesta del proceso, figura 65.

Figura 64. Interfaz gráfica de monitoreo y control Simulink



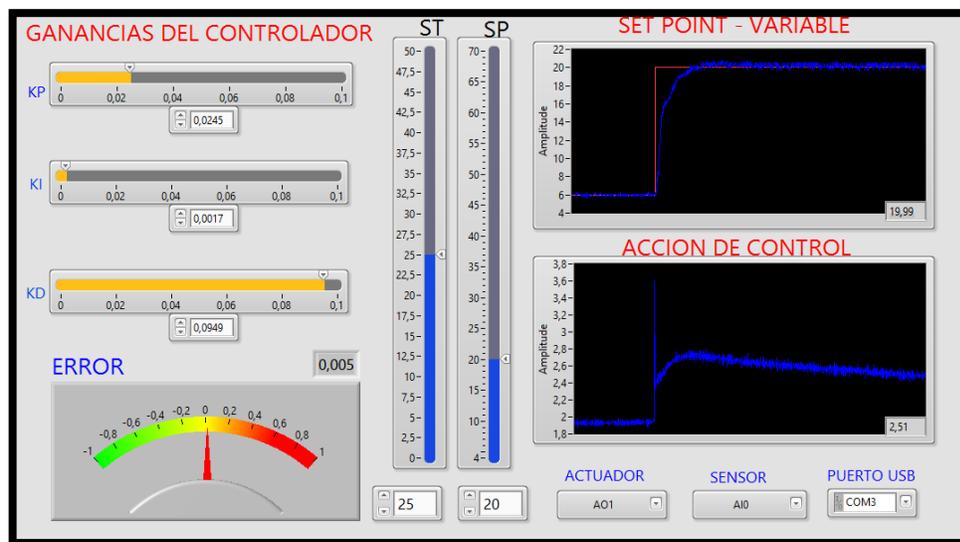
Fuente: Elaboración propia.

7. RESULTADOS

7.1 SISTEMA DE CONTROL LABVIEW - PTAD-S01

Al realizar pruebas sobre el sistema se obtuvieron los resultados que se muestran en la figura 66. En rojo tenemos la referencia la cual es de 20° para la barra, el azul es la posición de la barra leída por el sensor y en la parte inferior en color azul se observa la señal de control producida por el regulador PID discreto.

Figura 65. Respuesta del sistema real.



Fuente: Elaboración propia.

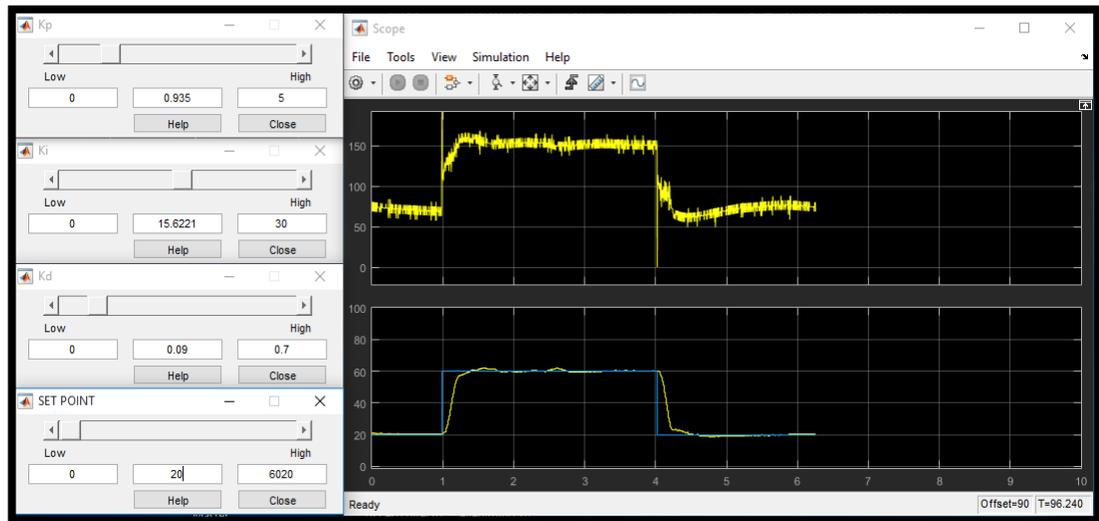
Como se puede observar el sistema tiene una buena respuesta, la referencia se alcanza sin que haya sobrepasos en la evolución de la posición, la cual se incrementa prácticamente de forma lineal con cierto retraso, hecho que nos permite aproximar el comportamiento de este sistema de segundo orden al de un sistema de primer orden con retardo. Cabe destacar que se observa algo de ruido hecho que es provocado por las vibraciones que introduce el movimiento de giro del motor.

7.2 SISTEMA DE CONTROL SIMULINK - ARDUINO

La prueba realizada consistió en configurar la interfaz de control con las mismas ganancias del controlador utilizadas en la simulación de la figura 63, donde se

excita el sistema la planta con una señal escalón inicial de 20 y final de 60, figura 67.

Figura 66. Comportamiento de la Acción de control y la respuesta del sistema real



Fuente: Elaboración propia.

En la figura 67, se muestra la interfaz donde se han ajustados las ganancias del controlador, en el que la señal alcanza la referencia a los 0.610 segundos sin sobrepasos y con un error en estado estacionario menor al 2% equivalente al 0,005. Como se puede evidenciar tiene el mismo comportamiento que el sistema simulado del figura 63, donde la acción de control y a respuesta del sistema comparte las mismas características.

8. MATERIALES Y MÉTODOS

8.1 MATERIALES

- SOFTWARE MATLAB, LABVIEW, SIMULINK.
- SENSOR ANALOGO (POTENCIOMETRO LINEAL).
- BALANCIN, MOTOR DC, HELICE.
- TARJETA ADQ PTA - S01.
- PLACA ARDUINO MEGA 2560
- CIRCUITO POTENCIA TIP41C.
- ADPTADOR DC 5 VOLTIOS.
- PC

8.2 MÉTODOS

7.2.1. Método científico.

El proyecto se realizara con ciclo de uso en cascada implementando las siguientes fases y actividades.

8.2.2. Análisis.

- Análisis para la identificación de sistemas dinámicos.
- Identificación de la función de transferencia del modelo con las técnicas implementadas en la guía.
- Experimentar en el prototipo la variable a controlar.
- Considerar los puntos críticos en la operación del sistema, para su puesta a punto.

8.2.3. Diseño

- Diseño de la interface para el control de la planta.
- Diseño del algoritmo para la comunicación y el proceso de datos entre la PC y la tarjeta de adquisición.
- Diseño de guía con las técnicas vistas durante el curso para la identificación del modelo, el controlador y el procesamiento de los datos.

8.2.4. Verificación

Se hará control del grado de inclinación de un balancín, por medio de un motor DC que hace girar una hélice a cierta velocidad. Esto con el fin de comprobar que el sistema es funcional.

9. CRONOGRAMA

| ACTIVIDADES A REALIZAR | JUNIO | | | | JULIO | | | | AGOSTO | | | | SEPTIEM | | | | OCTUBRE | | | |
|--|-------|---|---|---|-------|---|---|---|--------|---|---|---|---------|---|---|---|---------|---|---|---|
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Elaboración del prototipo y diseño de la interfaz | | | x | x | | | | | | | | | | | | | | | | |
| Pruebas de laboratorio y registro de datos | | | | | x | x | | | | | | | | | | | | | | |
| Identificación del sistema, diseño del controlador y simulación. | | | | | | | x | x | | | | | | | | | | | | |
| Implementación del controlador en el prototipo | | | | | | | | | x | x | | | | | | | | | | |
| Pruebas del sistema y ajustes | | | | | | | | | | | x | x | | | | | | | | |
| Documentación | x | x | x | x | x | x | x | x | x | x | x | x | x | x | | | | | | |
| Presentación de informe | | | | | | | | | | | | | | | x | x | | | | |
| Sustentación | | | | | | | | | | | | | | | | | x | | | |

Fuente: Elaboración propia.

11. CONCLUSIONES

Se diseñó la interfaz gráfica en LABVIEW y SIMULINK para el procesamiento de los datos y la ley de control (PID), Desde el punto de vista teórico, al comparar el modelo matemático simulado en lazo cerrado con el sistema real en cuanto a respuesta temporal y la acción de control vemos que ambas respuestas tienen comportamientos muy similares,

Se estableció la comunicación entre la PC y la Planta, Implementando la tarjeta de adquisición de datos DAQ PTDA – S01 y la placa ARDUINO MEGA 2560. Implementando en LabVIEW y Simulink la configuración del controlador en forma de velocidad angular ayudando que la acción de control sea más suave mejorando el desempeño del actuador (Motor DC).

Con este proyecto se han puesto en práctica gran parte de los conocimientos adquiridos durante los años de formación en la institución ayudándonos satisfactoriamente en el Diseño y construcción de los circuitos necesarios para el sensor y la etapa de potencia.

Se logró Construir y entregar prototipo (Balancín Motor hélice) para el laboratorio de electrónica con su respectiva demostración y comprobación.

Se pudo constatar con la aplicación de las guías de laboratorio realizadas dentro del proyecto son muy prácticas y de fácil manejo para el usuario realizando el paso a paso para el control de la planta.

El rango de operación del sistema se encuentra entre los 6° y 60° . Por la dinámica del sistema en relación a la estabilidad, esta se pierde al sobrepasar los 60° grados de inclinación del balancín, aunque se observaron mejores resultados el sistema implementado en Simulink - Arduino logrando alcanzar estabilidad a los 80° grados de inclinación.

De la prueba con el sistema Simulink - Arduino podemos concluir que las ganancias obtenidas por el método ZN son un punto de referencia para realizar ajustes necesarios y así obtener nuevas ganancias que hagan que el sistema sea más eficiente.

en el área de la Ingeniería en Electrónica Industrial. Me ha aportado un punto de vista analítico a la hora de reconocer como utilizar los conocimientos adquiridos para resolver cierto problema de la vida real, hecho que me servirá de experiencia para mi desarrollo como profesional de la rama electrónica.

11. RECOMENDACIONES

El sistema de control ha sido implementado para un mecanismo con un solo grado de libertad, que al compararlo con un helicóptero sería el de ascenso y descenso (giro respecto al eje vertical), para futuros proyectos no estaría mal añadir otro grado de libertad que bien puede ser el de desplazamiento (giro respecto al eje horizontal), con esto ya estaríamos creando un sistema con un mayor grado de complejidad y algo más cercano al helicóptero tradicional con un rotor principal y el rotor secundario o rotor de cola.

12. REFERENCIAS BIBLIOGRÁFICAS

- Alvarado Moya, P. (2008). *Señales y Sistemas. Fundamentos Matemáticos. Ediciones Centro de Desarrollo de Material Bibliográfico. Primera edición. Instituto Tecnológico de Costa Rica. Cartago, Costa Rica. Web Site: <http://www.ie.itcr.ac.cr/palvarado/Modelos/modelos.pdf>*
- Ghidella, J. (1994). *Simulink. Simultion and Model Based Desing. Web Site: <http://www.mathworks.com/products/simulink/>*
- Lara, C. (2008). *Sistemas Lineales. Web Site: http://es.slideshare.net/catita_potter/sistemas-lineales-presentation*
- *Manual Técnico (s/f). P-TAS S01 BASIC USB DAQ Multifunción. Revisión 8. Web Site: <http://181.143.31.26/pub/internet/nuevo/ptads01.pdf>*
- Ogata, K. (1987). *Dinámica de Sistemas. Primera Edición. Editorial Prentice Hall. México. Web Site: <http://es.slideshare.net/Angelnoquez/dinamica-de-sistemas-katsuhiko-ogata-45947258>*
- Ogata, K. (1998). *Ingeniería de Control Moderna. Controles PID e Introducción al Control Robusto. Página 669. Tercera Edición. Editorial Prentice Hall. Web Site: [http://www.matcuer.unam.mx/~victor/Sistemas/ingenieria de control moderna.pdf](http://www.matcuer.unam.mx/~victor/Sistemas/ingenieria_de_control_moderna.pdf)*
- Ogata, K. (2010). *Ingeniería de Control Moderna. Quinta Edición. Editorial Prentice Hall. Madrid. Web Site: <http://es.slideshare.net/faisyvillalobos1/ingenieria-decontrolmodernaogata5ed>*
- Ogata, K. (1998). *Sistemas de Control en Tiempo Discreto. . Segunda Edición. Editorial Prentice Hall. México. Web Site: <http://es.slideshare.net/angeljimenez984/sistemas-de-control-en-tiempo-discreto-katsuhiko-ogata>*
- S/A (2014). *Analog Sampling Basics. Web Site: <http://www.ni.com/white-paper/3016/en/>*
- *Telecomunicaciones Prodetel (2015). Sistemas HMI / SCADA. Web Site: <https://prodetel.net/sections/scada.html>*

- Viltres la Rosa, V. (2012). Control de Posición de un Balancín con Motor y Hélice. Escuela de Ingeniería Industrial. Universidad de Valladolid. España. Web Site: <http://uvadoc.uva.es/handle/10324/1895>
- Quanser Headquarters and Mailing Address, disponible en internet: http://www.quanser.com/products/2dof_helicopter.
- David Rebordinos, 20/07/2012. disponible en internet: <http://blog.deinventos.com/construyendo-un-cuadricoptero-i/>
- <http://sine.ni.com/nips/cds/view/p/lang/en/nid/210347>
- <http://www.ni.com/white-paper/52079/en/>
- <http://preciod.com/ar/potenciometro-de-cermet-100k-lineal-2-watts-bourns-91a-104-akODS.precio>
- <http://www.aliexpress.com/popular/brushed-dc-motor.html>
- <http://articulo.mercadolibre.com.co/MCO-416539479-micro-motor-dc-modelo-cc-carbon-12mmx395mm-n60-2535- JM>
- <http://www.uv.es/marinjl/electro/transistores.html>
- <http://www.taringa.net/posts/hazlo-tu-mismo/16149902/Leds-Audioritmicas-en-mi-PC.html>
- <http://spanish.alibaba.com/product-gs/transistor-tip41-363414024.html>
- https://es.wikipedia.org/wiki/Adquisici%C3%B3n_de_datos
- <http://www.automatiksolutions.com/servicios/soluciones-en-software/sistema-de-supervision-scada>
- http://www.isa.cie.uva.es/~felipe/docencia/ra12itielec/tema1_trasp.pdf
<http://attila.sdsu.edu/me205/modules/1labview/03loops/2tutorial.html>
- <https://es.wikipedia.org/wiki/MATLAB>
- https://es.wikipedia.org/wiki/Sistema_LTI
- <http://www.eveliux.com/mx/Conversion-Analogico-Digital-ADC.html>

- http://www.infoagro.com/riegos/control_riego_y_fertilizacion.htm
- <http://www.monografias.com/trabajos93/controlador-electronico-velocidad-pid-usando-cpld/controlador-electronico-velocidad-pid-usando-cpld.shtml>
- http://www.tecnoficio.com/electricidad/instrumentacion_industrial4.php
- <http://www.monografias.com/trabajos27/analogico-y-digital/analogico-y-digital.shtml>
- <http://slideplayer.es/slide/3079472/>

13. ANEXOS

ANEXO 1. GUIAS DE LABORATORIO

Las GUIAS de Laboratorio CDLAB (Laboratorio de Control Digital) permiten a los estudiantes afianzar los conocimientos adquiridos en los cursos de sistemas dinámicos y control automático, a través de experiencias de laboratorio en las que debe desarrollar sistemas de control pasando por varias etapas como son:

- Etapa de adquisición de señales analógicas.
- Etapa de construcción de modelos a partir de datos de computadora.
- Etapa de diseño de la estrategia de control.
- Etapa de implementación y la validación de la solución desarrollada.

Para esto se han diseñado e implementado instrumentos virtuales elaborados en LabVIEW 2011 y Simulink, para la generación y adquisición de señales a través de la tarjeta de adquisición PTAD-S01 y la placa Arduino mega 2560 para controlar el ángulo de inclinación de una planta real conformada por un motor balancín hélice.

OBJETIVOS

- Construir modelos de sistemas dinámicos a partir de principios físicos y datos experimentales.
- Implementar herramientas de software para la identificación y control de sistemas dinámicos (Matlab, Simulink, LabVIEW).
- Diseñar algoritmos de control tipo PID para cumplir especificaciones predeterminadas.
- Implementar leyes de control de forma análoga y digital.

GUIA No. 1

GENERACIÓN Y ADQUISICIÓN DE SEÑALES
Instituto Tolimense de Formación Técnica profesional
Facultad de Ingeniería y Ciencias Agro-Industriales
Departamento de Electrónica Laboratorio de Control

INTRODUCCIÓN

IVGARD (Instrumento virtual para Generación Adquisición y Registro de Datos) lo integra, un instrumento virtual desarrollado en LABVIEW y la tarjeta TPAD-S01 para la adquisición y registro de datos. Estos datos serán analizados y comparados con la ayuda de herramientas como IPORT DATA de Matlab para su visualización.

OBJETIVOS

- Conocer el entorno de trabajo del Instrumento Virtual de generación de señales y registro de datos IVGARD.
- Generar y adquirir señales a través de la tarjeta de Adquisición TPAD-S01. Almacenar y manipular los datos de señales adquiridas con el IVGARD
- Conocer la herramienta IPORT DATA de Matlab para la importación y procesamiento de los datos generados por el proceso.

EQUIPO NECESARIO

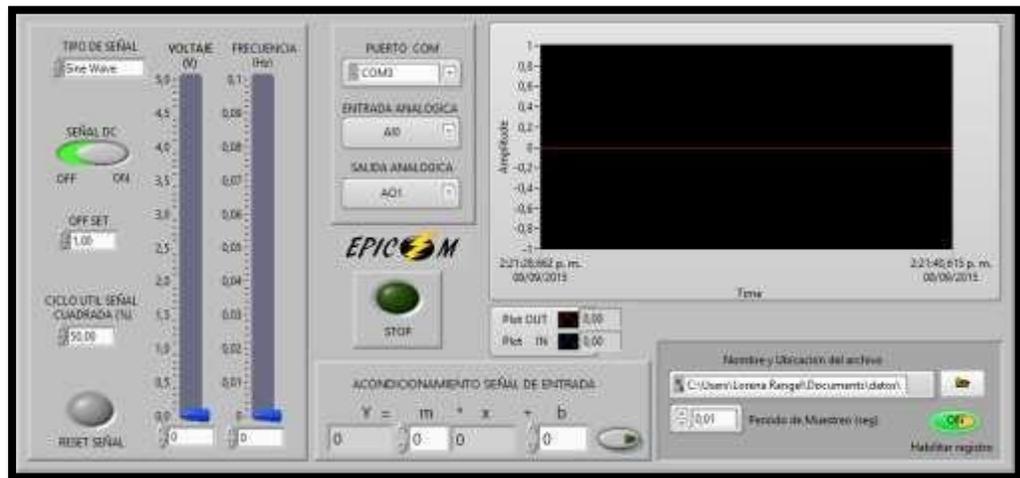
- Computadora con Windows 7, LabVIEW y Matlab 2013 o superior
- Tarjeta de adquisición PTAD-S01 + interfaz cable USB
- Instrumento Virtual de generación de señales, adquisición y registro de datos IVGARD

DESCRIPCIÓN DEL SISTEMA

Instrumento virtual “IVGARD”

El IVGARD es una interfaz gráfica diseñada para generar, adquirir, visualizar y registrar señales. Esta interfaz está conformada por cinco bloques, cada uno de ellos con funciones específicas dentro del programa, ver figura 1.

Figura 1. Instrumento virtual IVGARD.



A continuación se hace una breve descripción de cada uno de los bloques que hacen parte del panel frontal del instrumento.

Bloque de periféricos

Encargado de configurar las entradas y salidas análogas de la tarjeta de adquisición y el puerto (COM), por el cual fluyen los datos, figura 2.

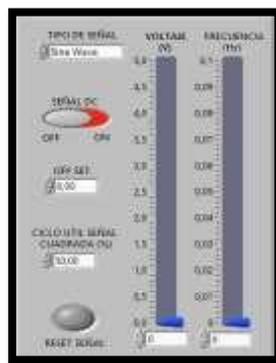
Figura 2. Bloque periféricos.



Bloque de generador de señales.

Como su nombre lo indica es el encargado de generar las señales requeridas por el usuario, estas señales se pueden configurar con ayuda de los diferentes controles (Slider, Numérico, Text y Booleano), de los cuales está conformado el bloque, figura 3.

Figura 3. Bloque Generador de señales.



Bloque de registro de datos.

Este bloque es el encargado de almacenar los datos que son generados y adquiridos a través de las entradas y salidas análogas de la tarjeta de adquisición. En este bloque también se configura el tiempo que tarda el sistema en tomar cada muestra para su respectivo registro, figura 4.

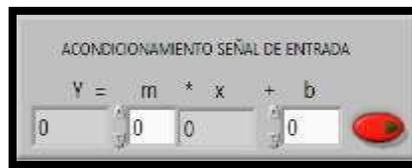
Figura 4. Bloque registro de datos.



Bloque acondicionamiento de señal.

Conformado por controles e indicadores numéricos, es el encargado de representar la señal captada a un nivel o valor adecuado para su interpretación, figura 5.

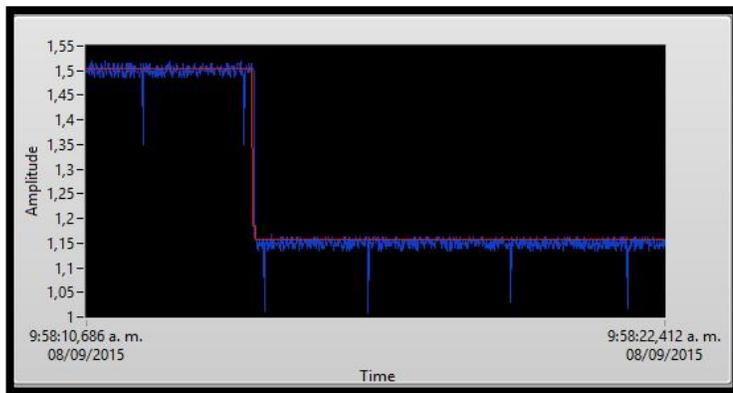
Figura 5. Bloque de acondicionamiento de señal.



Bloque de visualización

Conformado por un control tipo chart (grafico) encargado de representar la forma que tiene tanto la señal de entrada, como la señal de salida, figura 6.

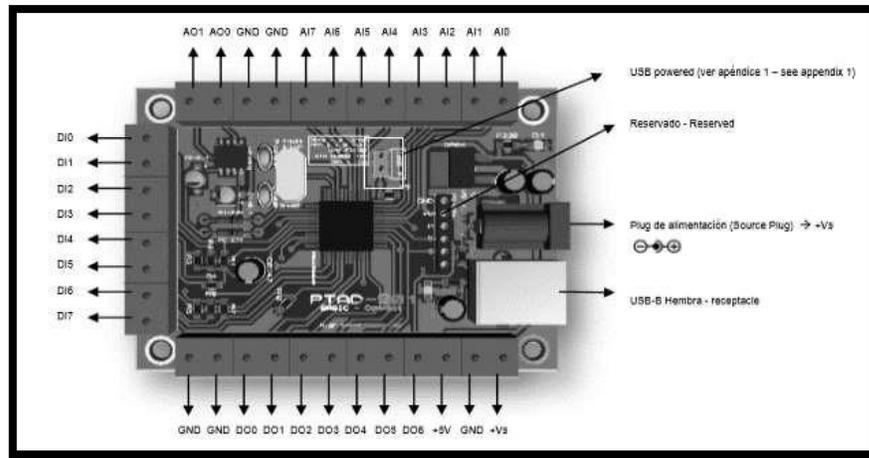
Figura 6. Bloque de visualización. En color azul la señal de entrada, en color rojo la señal de salida.



TARJETA DE ADQUISICIÓN.

La tarjeta adquisición PTAD-S01 es un sistema USB de adquisición de datos y control para PC orientado hacia la robótica y el control de procesos, el cual permite digitalizar señales en un rango de 0 a 5 voltios, ver figura 7.

Figura 7. Disposición de pines de la tarjeta adquisición PTAD-S01



También cuenta con canales digitales de entrada y salida TTL que le permite controlar eventos externos como por ejemplo actuadores, control de relés y otros sistemas digitales, tabla 1.

Tabla 1. Especificaciones básicas de la tarjeta de adquisición PTAD-S01

| Especificaciones - Specifications | |
|--|------------------------------------|
| Rango de voltaje de operación - Operating Voltage Range (V) | 9 - 12 |
| Corriente de alimentación Min - Min Operating Current Range (mA) | 200 |
| Voltaje de salida (E/S) - Output Voltage Range I/O (V) | 0 - 5 |
| Corriente de salida (E/S) - Output Current Range I/O (mA) | 25 mA |
| USB (Velocidad, Cumplimiento, CDC) - (speed, compliance, CDC) | Full Speed 12Mb/s, USB 2.0, Serial |
| Temperatura de operación - Temperature Range (C) | -40 to 85 |
| Rendimiento del CPU - CPU Core Performance | 48 MHz (12 MIPS) |
| Canales análogos de entrada - Analog inputs channels | 8 (AI0-AI7) |
| Canales análogos de salida - Analog outputs channels | 2 (AO0, AO1) |
| Canales digitales de entrada - Digital inputs channels | 8 (DI0-DI7) |
| Canales digitales de salida - Digital outputs channels | 7 (DO0-DO6) |
| Resolución de entrada - Input Resolution (bits) | 10 |
| Máxima velocidad de muestreo - Max Sampling Rate (kS/s) | 25 |
| Tipo de medida | Voltaje (V) |
| Software objetivo (Windows) | LabVIEW®, MATLAB® & Simulink® |
| Sistema operativo compatible - Compatible operative system | Windows, Linux, MAC |

PASO 1. ACOPLE DE LA TARJETA A LA PC

Con la ayuda del cable USB, conecte la tarjeta de adquisición PTAD-S01 a la PC, Una vez conectado el sistema establecerá automáticamente el puerto de comunicación para la recepción y el envío de datos, ver figura 8.

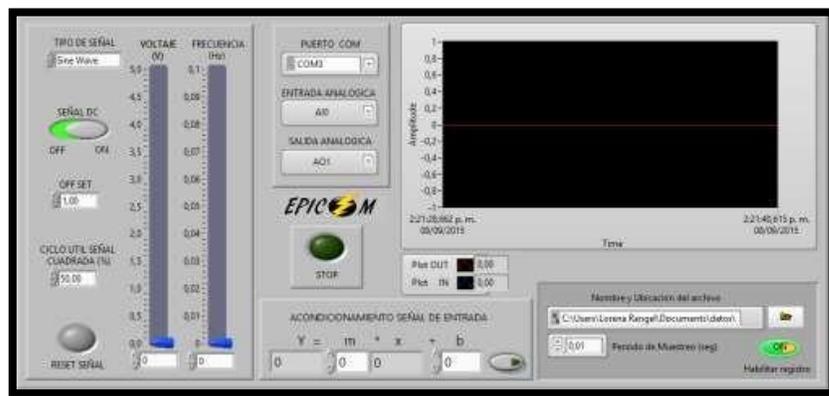
Figura 8. Tarjeta de adquisición PTAD-S01 y PC.



PASO 2. EJECUCIÓN DEL PROGRAMA.

Ejecute el instrumento virtual para la generación y adquisición de señales dando clic sobre ICONO, ubicado en el escritorio con el nombre "IVGARD" el cual visualizara el panel frontal del instrumento, ver figura 9.

Figura 9. Panel frontal del instrumento virtual



PASO 3. CONFIGURACIÓN DE BLOQUES DEL SISTEMA.

Una vez ejecutado el programa y visualizado en la pantalla se procede a configurar cada uno de los bloques de la siguiente forma.

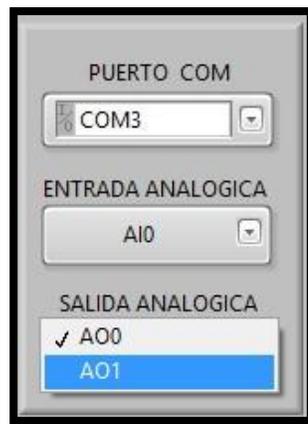
Configuración Bloque periféricos.

Despliegue el menú del control identificado con el nombre “PUERTO COM” y seleccione el puerto (COM#), que fue creado automáticamente por el sistema al acoplar la tarjeta a la PC. Luego ubíquese sobre el control identificado con el nombre

“ENTRADA ANALOGA” y despliegue el menú que le mostrara las diferentes entradas análogas que tiene la tarjeta de adquisición, seleccione la entrada análoga “A10” para la lectura. Por ultimo ubíquese sobre el control identificado con el nombre

“SALIDA ANALOGA” y despliegue el menú, seleccione la salida análoga (A01) para la escritura, ver figura 10.

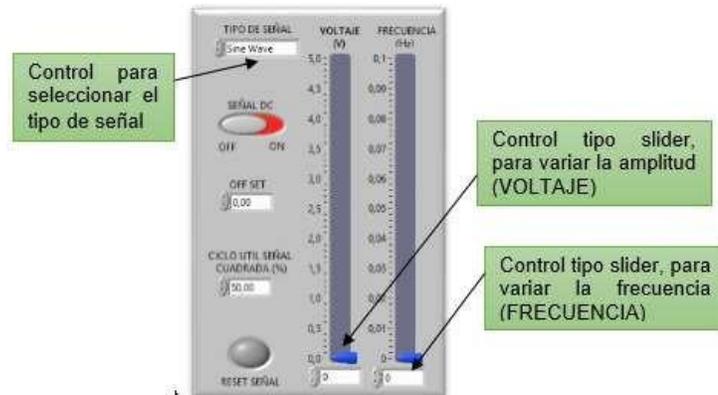
Figura 10. Configuración salida análoga como escritura.



Configuración del Bloque generador de señales

Configure una señal tipo senoide A través del control “TIPO DE SEÑAL” ubicado en la parte superior del bloque, por medio de los pulsadores ubicados en la parte izquierda del control navegue hasta que se visualice la opción “Sine wave”. Verifique que los controles tipo slider identificados como “VOLTAJE” y “FRECUENCIA” se encuentren con valores iniciales cero (0), figura 11.

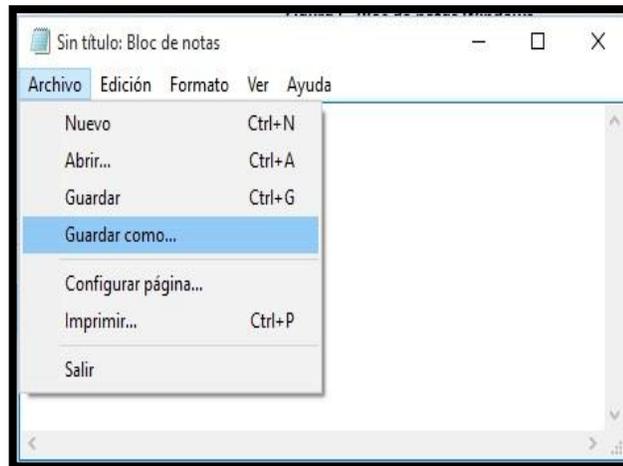
Figura 11. Configuración de la señal sinusoidal.



Configuración bloque registro de datos

Para la configuración de este bloque deberá crear un archivo con extensión .txt con la ayuda del “Bloc de notas de Windows”. Abra el bloc de notas desde el botón de INICIO de Windows > Todos los programas > Accesorios > Bloc de notas. Ejecute programa y una vez abierto seleccione en la barra de Menús la opción Archivo > Guardar como, figura 12.

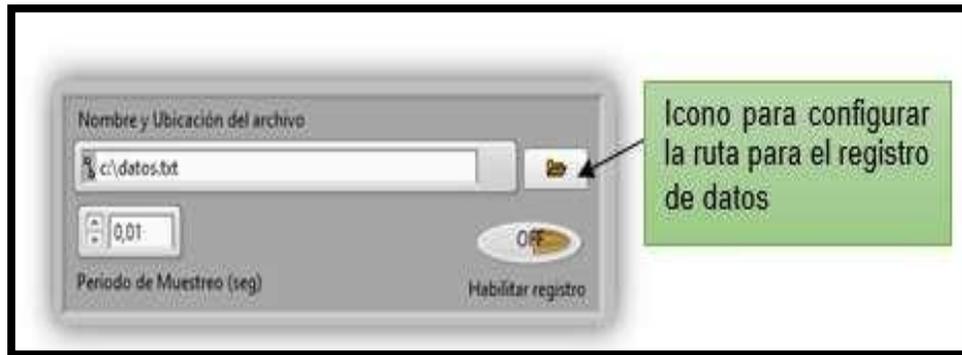
Figura 12. Ventana del Bloc de notas de Windows.



Esta acción desplegará la ventana “Guardar como” donde deberá buscar la carpeta “Mis documentos”, y guardar allí el archivo con el nombre “Datos”.

Una vez creado el archivo, diríjase al bloque “Registro de datos” e Ingrese la ruta donde quedo guarda el archivo dando clic sobre botón forma de carpeta, ver figura 13.

Figura 13. Configuración de ruta para el registro de datos.



Configuración periodo de muestreo

La configuración del periodo de muestreo se hace a través del control numérico ubicado en la parte inferior izquierda del bloque “Registro de datos”, esto se hace con el fin de indicarle al instrumento cada cuanto tiempo debe tomar muestras de la señal generada por el proceso, para ser almacenadas dentro de la carpeta “Datos.txt”, figura 14.

Figura 14. Configuración periodo de muestreo a 10 ms.



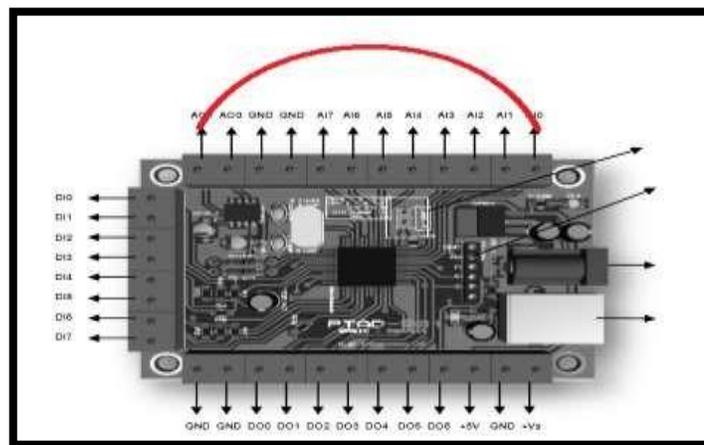
PASO 4. VERIFICACIÓN DE PULSADORES

Verifique que todos los pulsadores o interruptores del panel frontal del “IVGARD” se encuentren apagados (estado OFF), visualmente cuando están encendidos (estado ON) emiten una luz de color verde.

PASO 5. PROCEDIMIENTO

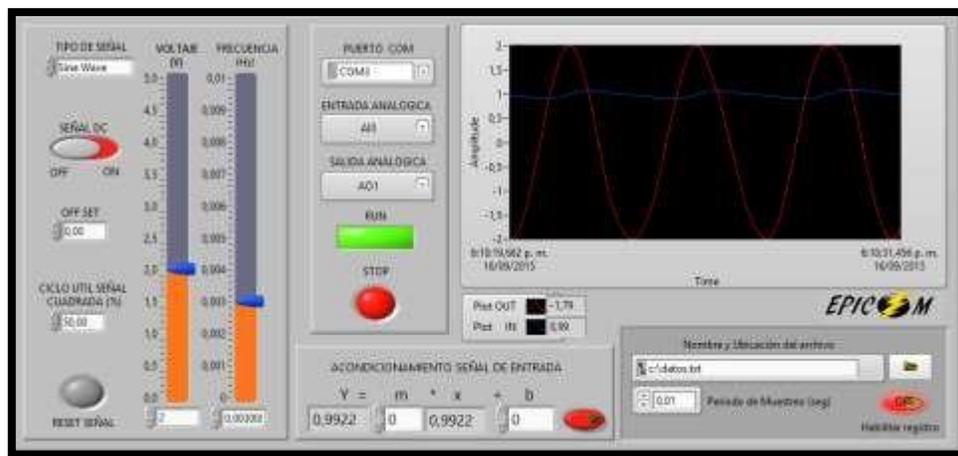
Usando un cable como puente, interconecte la salida análoga “AO1” a la entrada análoga “AI0” de la tarjeta de adquisición, figura 15.

Figura 15. Puente entre la salida análoga y la entrada análoga de la tarjeta de adquisición.



Genere una señal sinusoidal de frecuencia 0.003 Hz y amplitud 2 Voltios pico a través del instrumento virtual, utilizando los slider del “Bloque generador de señales”, figura 16.

Figura 16. Configuración de la Señal de prueba.



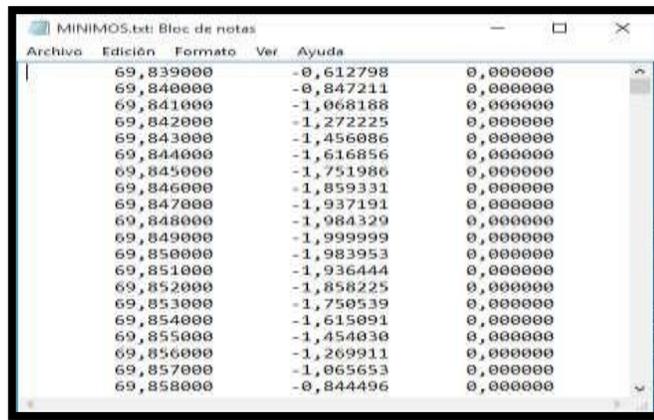
Luego de haber configurado la señal, proceda al realizar el registro de datos oprimiendo por un periodo no mayor a 10 segundos el interruptor “Habilitar registro”, el cual está ubicado en la parte inferior derecha del bloque “Registro de datos”, restablezca su posición, figura 17. luego oprima botón “STOP” ubicado en el panel frontal.

Figura 17. Bloque registro de datos.



Verifique que los datos fueron registrados correctamente, abriendo el archivo txt creado en el paso 3, este deberá contener 3 columnas, donde la primera columna tomada de izquierda a derecha corresponde al tiempo trascurrido entre muestras, para este caso es de 10 ms, la segunda son los datos de la señal de salida (señal paso) y la tercera los datos de la señal de entrada, figura 18.

Figura 18. Ejemplo de un archivo txt con datos ya registrados



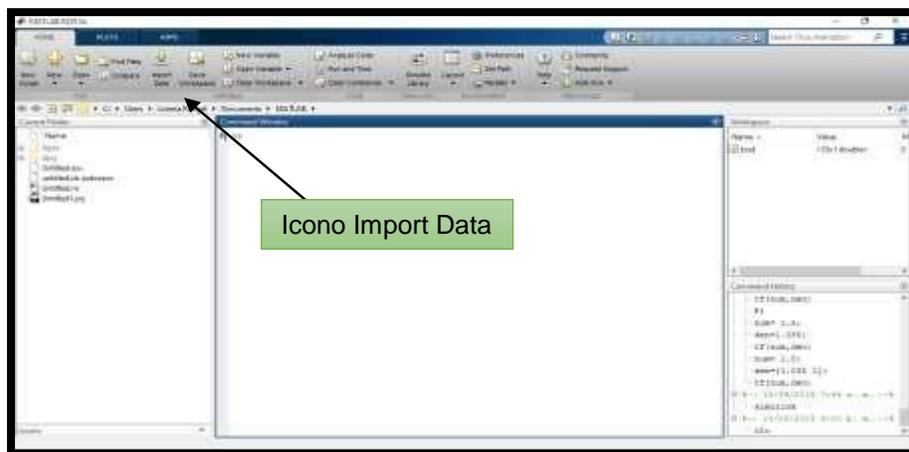
Una vez confirmado que los datos se registraron, ejecute el programa MATLAB dando doble clic sobre el icono ubicado en el escritorio, figura 19.

Figura 19. Icono de Matlab.



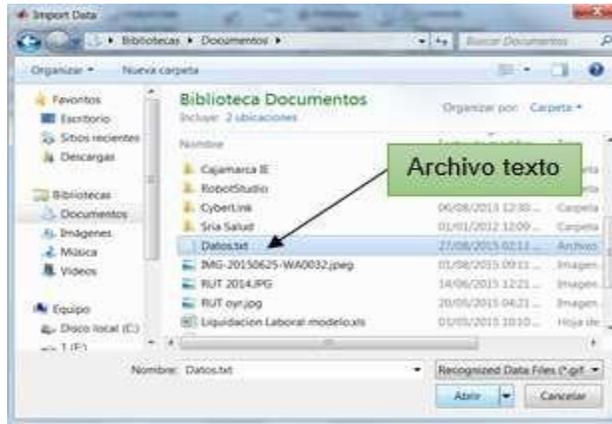
Este desplegará una ventana en la que se visualiza el entorno de trabajo del programa y los iconos de algunos toolbox como el "Import data" ubicados en la barra de herramientas figura 20.

Figura 20. Entorno MATLAB Visualización del Icono del toolbox "Import Data"



Haga clic sobre el icono “Import Data” y seleccione el archivo “Datos.txt”, que contiene registro de los datos, figura 21.

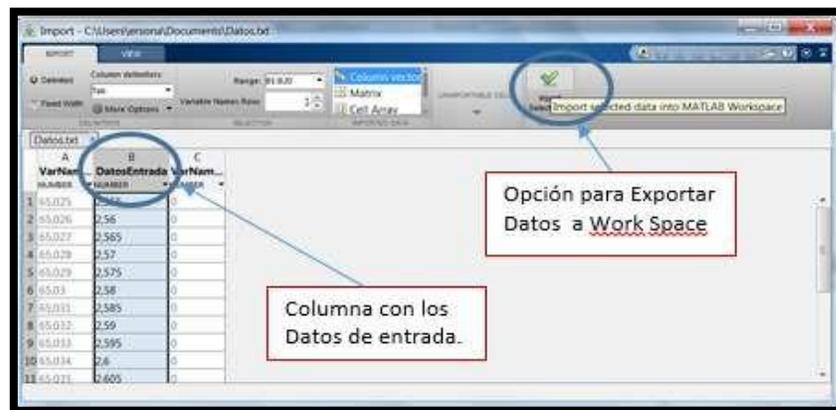
Figura 21. Ventana del “Import Data”



Una vez seleccionado el archivo oprima el botón abrir, esta acción importara los datos y abrirá la ventana de la herramienta “Import” la cual contiene las columnas con los datos registrados por el sistema, seleccione la columna que contiene los datos de entrada dando clic sobre la letra que la identifica. Cambie el nombre de la columna haciendo doble clic sobre “VarName” y remplaceela por “DatosEntrada”.

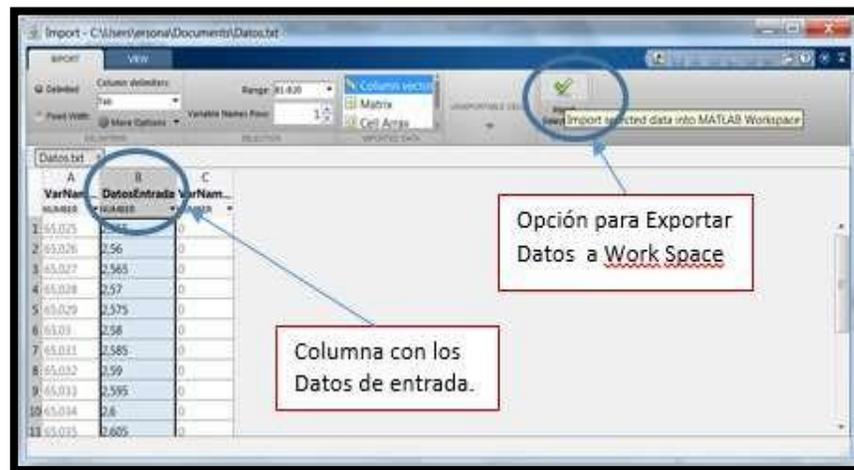
Para poder exportar los datos al “Workspace” de Matlab Haga clic sobre el icono “Import Selection” que se encuentra en la parte superior derecha de la barra de herramientas, figura 22

Figura 22. Ventana de la herramienta “Import” .



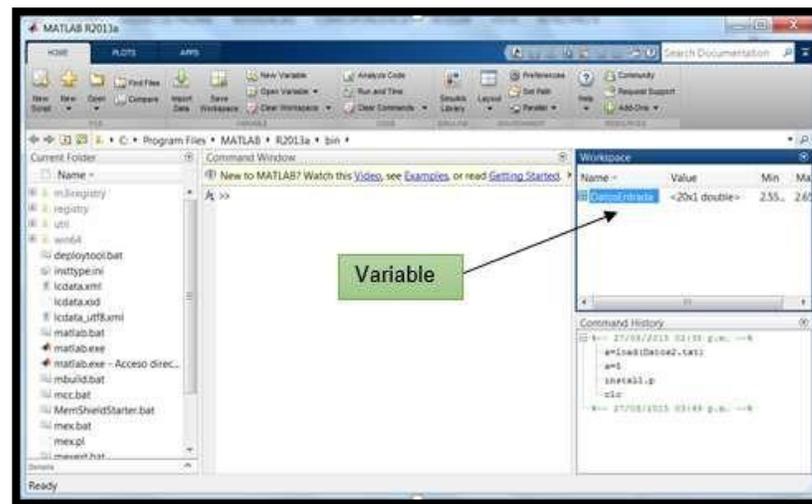
Una vez importados los datos, el sistema creara la variable con el nombre “DatosEntrada” y la enviará al “Workspace”, figura 23.

Figura 23. Ventana de la herramienta “Import”.



Una vez importados los datos, el sistema creara la variable con el nombre “DatosEntrada” y la enviará al “Workspace”, figura 24.

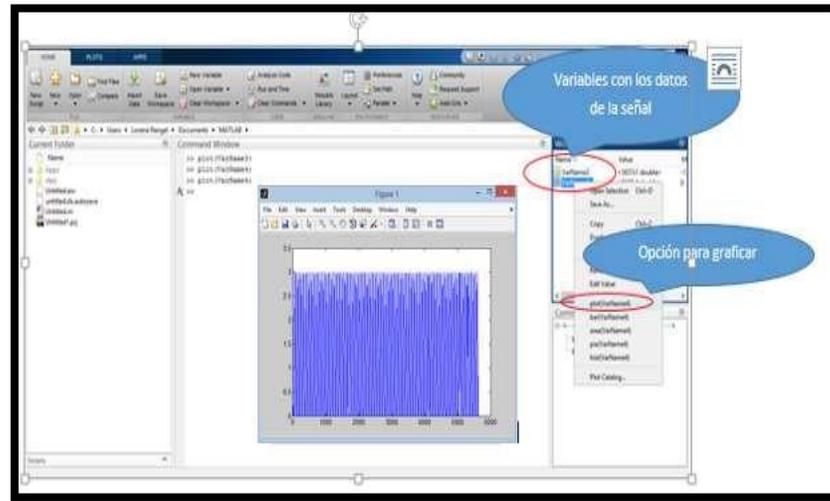
Figura 24. Workspace de Matlab.



Ubique el curso sobre la variable y oprima el clic derecho del mouse, esta acción desplegara una ventana donde deberá seleccionar la opción

“plot(DatosEntrada)”, luego oprima la tecla “Enter” para genera el gráfico que representara la señal generada por el instrumento, figura 25.

Figura 25. Visualización de la señal a través del “plot” de Matlab.



Con la herramienta “Zoom In” ubicado en la barra del “plot” agrandamos la señal dando clic izquierdo del mouse sobre la esta. Para conocer los valores que toma la señal en su pico más alto, seleccionamos la herramienta “Data Cursor” ubicada en la barra del plot, figura 26.

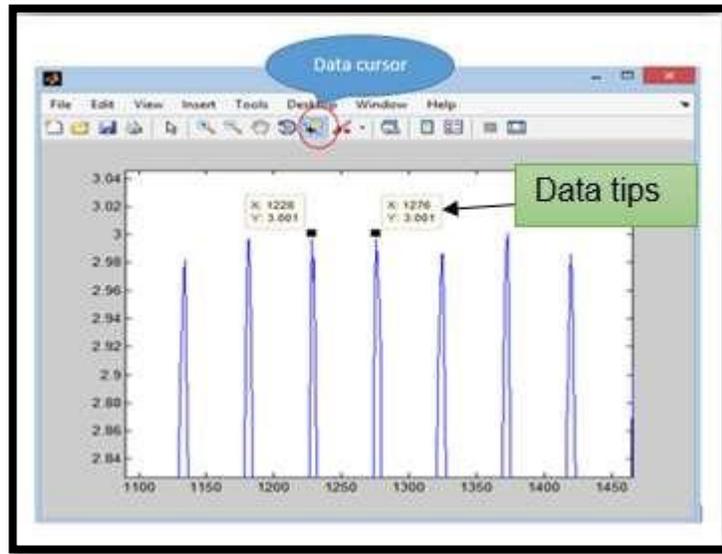
Figura 26. Barra de menús de la ventana del plot.



Luego ubicamos el cursor sobre la señal y hacemos clic para implantar un “Data tip”, el cual visualiza los valores (x,y), ver figura 27.

NOTA: Para implantar otro “Data tip” sobre el “Plot”, deberá dar clic derecho del mouse sobre la gráfica y dar clic sobre la opción “Crear nuevo data tip”

Figura 27. Visualización de los valores pico de la señal a través del “Data tips”.



Para obtener el periodo (T) de la señal, se restan los valores de X visualizados en los “Data tips” de cada pico, luego una vez hallado el periodo se aplica la siguiente fórmula para determinar la frecuencia de la señal.

$$f = \frac{1}{T}$$

NOTA: La frecuencia hallada deberá ser igual a la configurada a través del slider de “FRECUENCIA”

GUIA No. 2

IDENTIFICACION DE SISTEMAS DINAMICOS
Instituto Tolimense de Formación Técnica profesional
Facultad de Ingeniería y Ciencias Agro-Industriales
Departamento de Electrónica Laboratorio de Control

INTRODUCCIÓN

En la siguiente práctica se podrá analizar el comportamiento dinámico de un sistema eléctrico, conformado por un circuito RC, el cual es excitado por medio de una señal tipo escalón unitario que será generada a través del instrumento virtual IVGARD y la tarjeta de adquisición PTAD –S01.

La curva de respuesta del sistema será registrada y almacenada para posteriormente ser procesada con la aplicación del toolbox “Ident” de Matlab, este nos dará el modelo aproximado del sistema (función de transferencia).

El comportamiento de este modelo será simulado con la herramienta Simulink de Matlab para posteriormente ser comparada con el modelo real.

Objetivos

- Generar y adquirir señales a través de la tarjeta de Adquisición TPAD-S01.
- Manipular y almacenar señales adquiridas con el sistema.
- Conocer el entorno de la herramienta Ident de Matlab para la identificación de sistemas dinámicos.
- Comparar el sistema simulado con el real.

Equipo disponible

- PC
- Tarjeta de adquisición PTAD-S01
- Instrumento Virtual de generación de señales, Adquisición y registro de datos LabVIEW IVGARD
- Matlab con Simulink
- Condensador 4700uF
- Resistencia de 1K Ω
- Cables

PASO1. CONFIGURACIÓN DEL SISTEMA

Para la configuración del sistema se deberán realizar los pasos del 1 al 6 vistos en la GUIA 1.

PASO 2. MONTAJE FISICO DE LA PLANTA

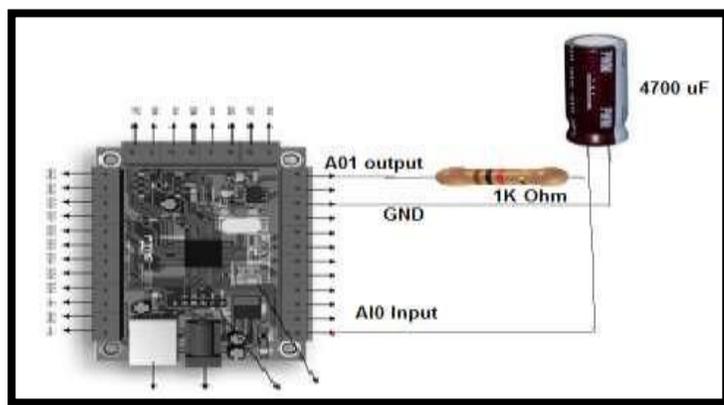
Realice el montaje físico conectando un extremo de la resistencia de $1K\Omega$ al conector de la salida análoga AO1 de la tarjeta PTAD-S01.

Suelde el otro extremo de la resistencia al pin positivo del condensador de 4700 uF .

Suelde el extremo de un conductor al pin positivo del condensador de 4700 uF y el otro extremo conéctelo a la entrada análoga AI0 de la tarjeta PTAD-S01.

Conecte el pin negativo del condensador al conector GND de la tarjeta.

Figura 1. Montaje del Circuito RC en la tarjeta ADQ PTAD-S01.



PASO 3. SEÑAL PASO Y REGISTRO DE DATOS

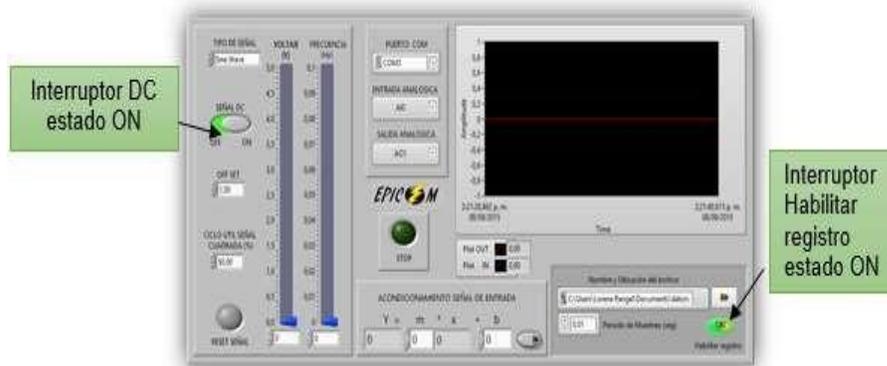
Abra el Instrumento Virtual para la Generación Adquisición y Registro de Datos (IVGARD), dando clic sobre el icono ubicado en el escritorio de Windows.

Figura 2. Icono IVGARD.



Configure el instrumento virtual para generar una señal DC por medio del interruptor “SEÑAL DC” llevándolo a la posición ON, luego habilite el registro de datos activando el interruptor “Habilitar registro” ubicado en el bloque Registro de datos, ver figura 3.

Figura 3. Ubicación en el panel los interruptores SEÑAL DC y habilitar registro.



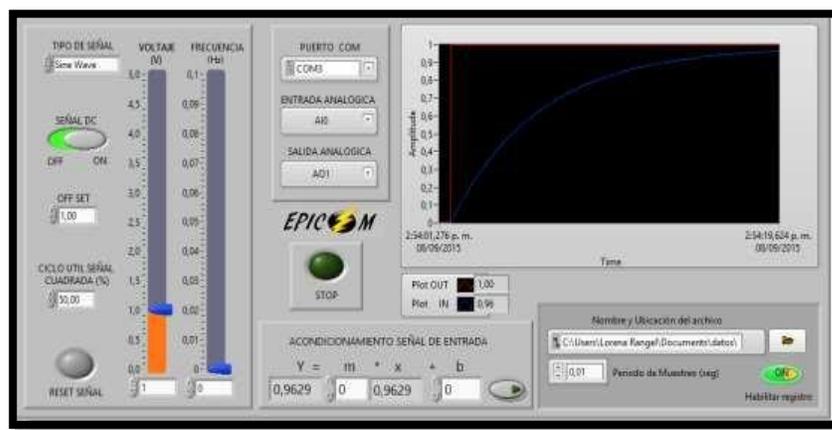
Genere una señal escalón unitario, cambiando el estado inicial “0” del Display digital a “1”, el cual se encuentra ubicado en la parte inferior del Slider identificado con el nombre de VOLTAJE, ver figura 4.

Figura 4. Escalón unitario.



Verifique que se visualicen la señal paso y la señal generada por el proceso en el bloque de visualización, figura 5.

Figura 5. Visualización de las señales. En rojo la señal paso, en azul la respuesta del sistema.



Una vez que la señal se estabilice, deshabilite el registro de datos restableciendo la posición del interruptor al estado OFF.

Verifique que los datos se hallan registrado correctamente abriendo el archivo “Datos.txt” ubicado en la carpeta Mis documentos, recuerde que el archivo debe contener 3 columnas siendo la primera el tiempo de muestreo, la segunda los datos de entrada (señal paso) y la tercera los datos de salida (respuesta del proceso).

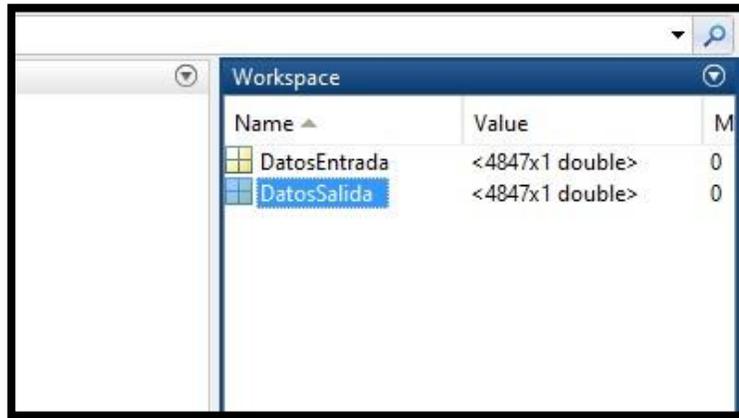
PASO 4. PROCESAMIENTO DE DATOS Y ANÁLISIS DE LA RESPUESTA TEMPORAL

Importe los datos de entrada y salida con la herramienta “Import data” de Matlab, Cambien el nombre de la columna 2 por “Datos Entrada” y la columna 3 por “Datos Salida”.

Exporte los datos al "WorkSpace" a través del "Import Selection" ubicada en la barra de herramientas de la ventana del "Import data".

Verifique en el "WorkSpace" que se hayan creado las variables "DatosEntrada", "DatosSalida", ver figura 6.

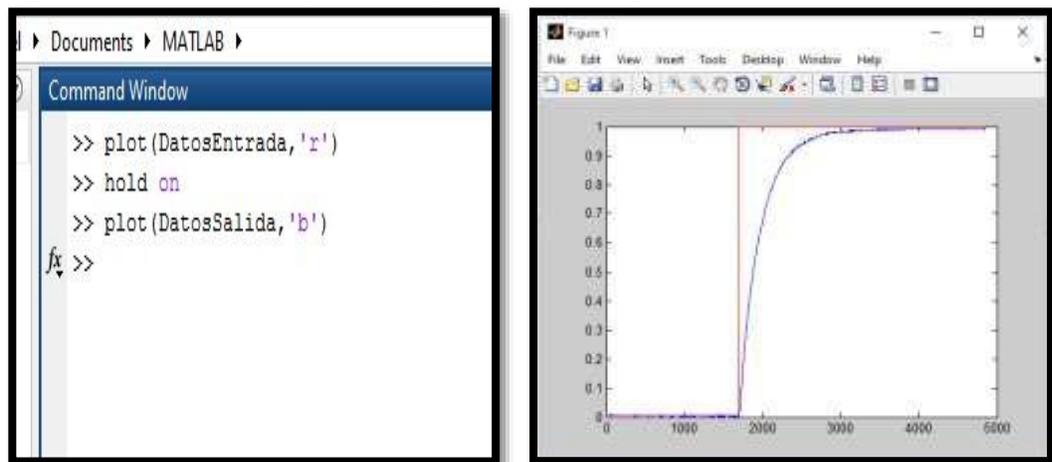
Figura 6. Importación de variables al "WorkSpace" de Matlab.



NOTA: En la práctica 1, se explica detalladamente la forma de llevar a cabo los paso 1, 2, 3, y 4.

Para visualizar los datos de entrada y salida en una misma gráfica, Digite en el "Command window" la instrucción que se muestra en la ventana izquierda, donde de la letra 'r' y 'b' representan el color de la señal, figura 7.

Figura 7. Instrucción para la representación gráfica de los datos.

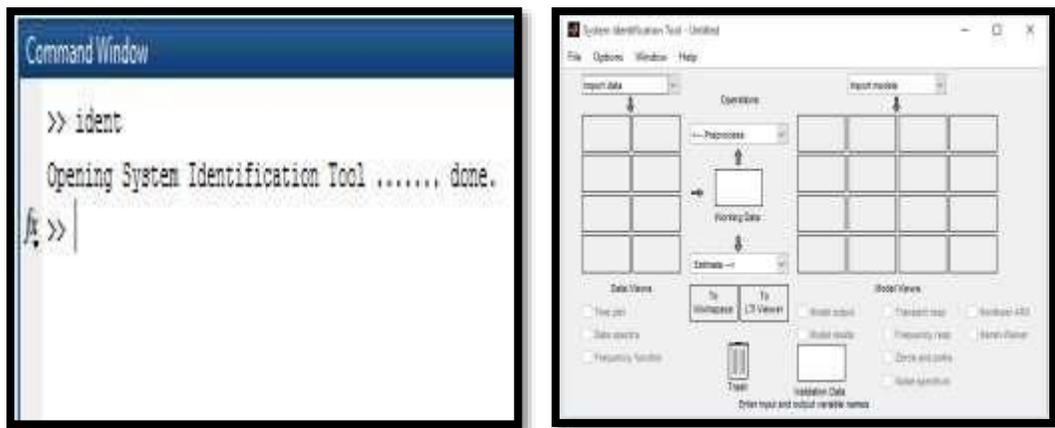


PASO 5. IDENTIFICACIÓN DEL MODELO DEL SISTEMA MÉTODO “IDENT”

Una vez procesados los datos, se procede a identificar la función de transferencia del modelo, utilizando el Toolbox “Ident” de Matlab.

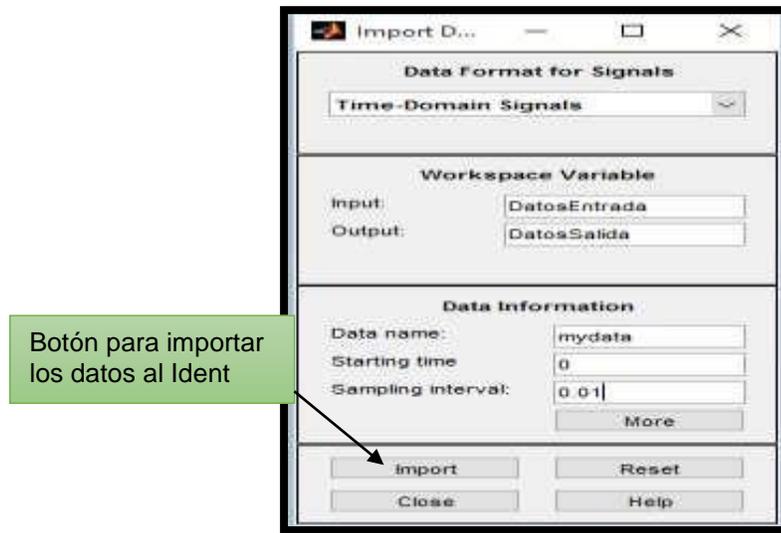
Para generar la ventana de la herramienta “Ident” ubíquese en el “Command window” de Matlab y digite la siguiente instrucción: `>> Ident`, luego oprima la tecla Enter, ver figura 8.

Figura 8. A la izquierda Command Window, a la derecha ventana del ident.



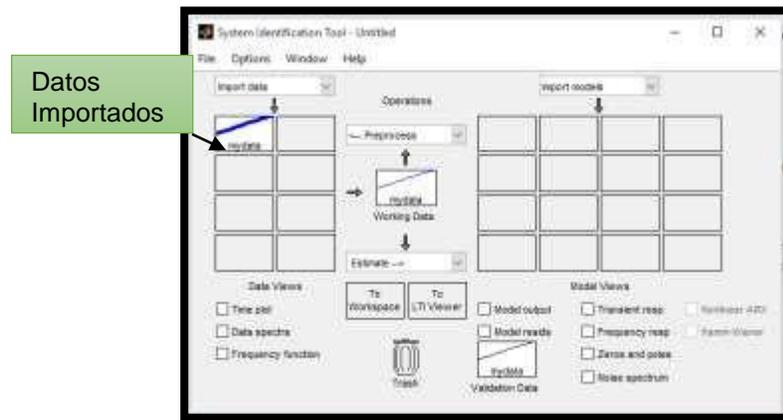
Abra la pestaña del “Import data” y de clic en la opción “Time domain data” la cual abre una ventana donde nos pide que ingresemos el nombre de las variables que contienen los datos de entrada y salida ubicadas en el “WorkSpace”, el tiempo inicial (Starting time) = 0 y el intervalo de muestreo (Sampling interval) = 0.01, ver figura 9.

Figura 9. Parámetros del Import Data.



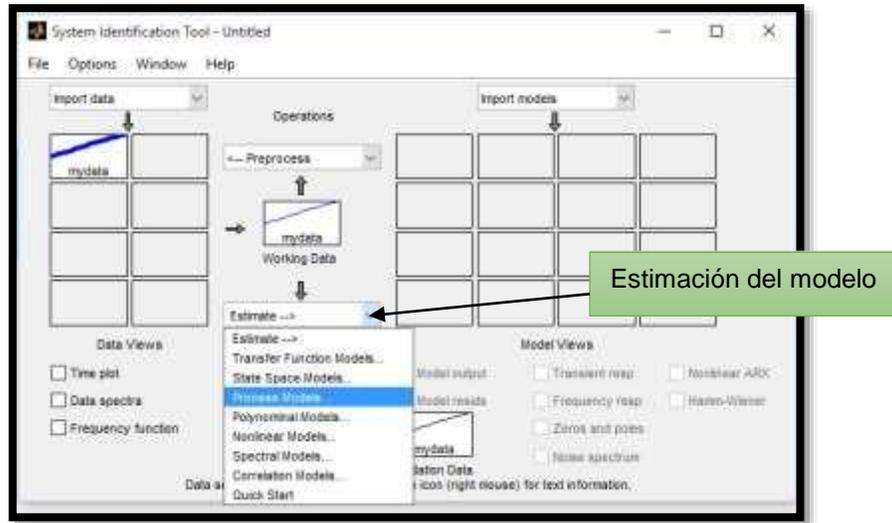
Para la importación de los datos al Ident, oprima el botón “Import” ubicado en la parte inferior de la ventana del “Import Data”.

Figura 10. Visualización de los datos importados al Ident.



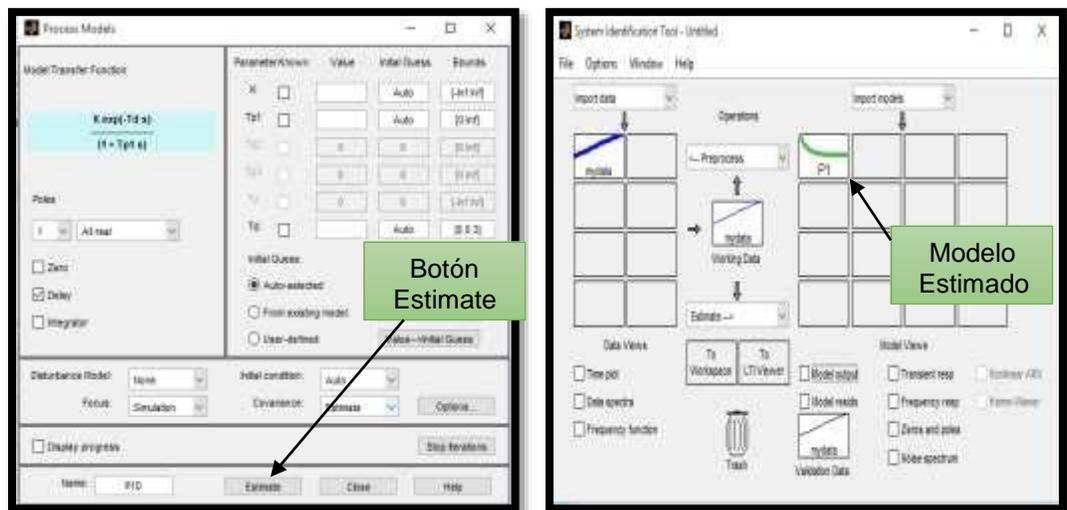
Una vez verificado que se hayan importado los datos figura 10, proceda hallar un modelo estimado del proceso abriendo la pestaña “Estimate” opción “Process Models”, ver figura 11.

Figura 11. Ubicación de la pestaña Estimate en la ventana del “Ident.”



Una vez generada la ventana “Process Models” reemplace parámetro de la pestaña “All Real” por la opción “Underdamped”, luego oprima el botón “Estimate” ubicado en la parte inferior de la ventana para generar el modelo estimado del proceso de, ver figura 12.

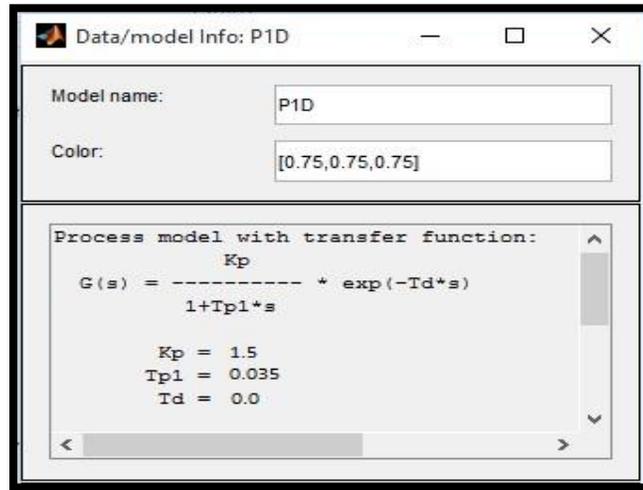
Figura 12. Izquierda Ventana del Process Models, derecha ventana del Ident con el Modelo estimado.



Haga doble clic sobre el modelo estimado del proceso para desplegar la ventana

“Data/model info” donde podrá visualizar la función de transferencia del modelo, ver ejemplo de la figura 13.

Figura 13. Ventana del Data model info.



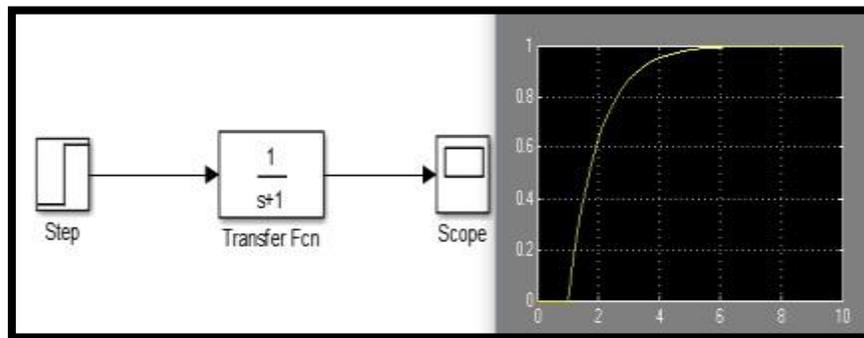
NOTA: Para más información utilice la guía de identificación de sistema dinámico método Ident que se encuentra en el siguiente link.

<http://www.mathworks.com/help/ident/gs/identifylinearmodelsusingthegui>.

PASO 6. VALIDACION DEL MODELO

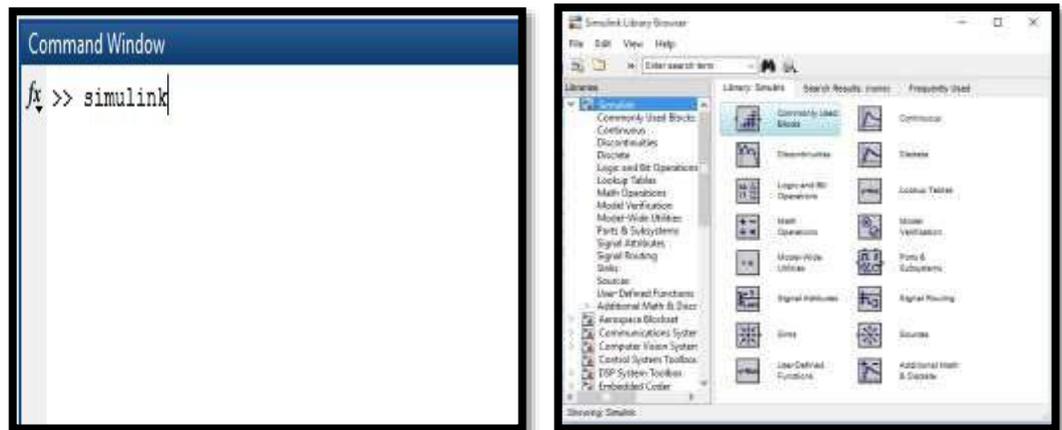
Simulink es un herramienta de gran utilidad para la simulación de sistemas dinámicos, principalmente se trata de un entorno de trabajo gráfico, en la que se especifican las partes de un sistema y su interconexión en forma de diagrama de bloques, ver figura 14.

Figura 14. Simulación de un sistema de primer orden en lazo abierto.



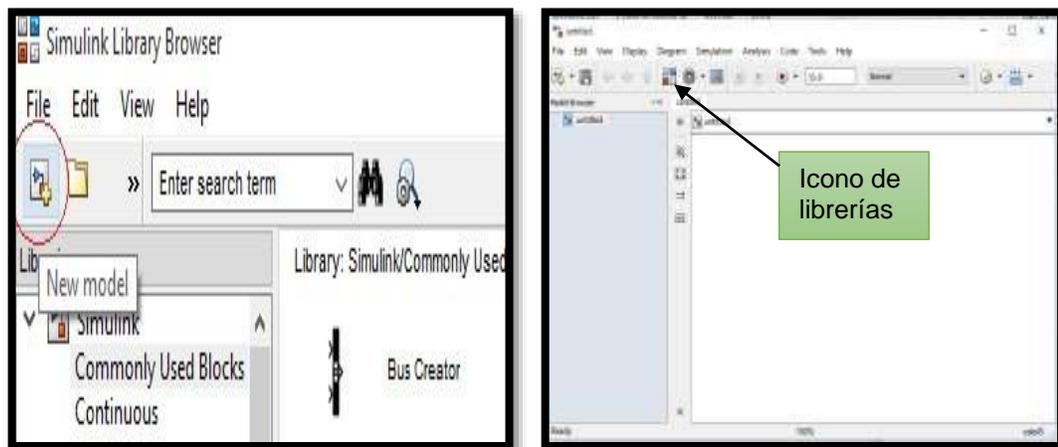
Para la validación de nuestro modelo en primer lugar, lanzaremos la aplicación escribiendo *Simulink* en la línea de comandos de Matlab (“*Command window*”) luego oprima la tecla “*Enter*” para visualizar la ventana de Simulink, figura 15.

Figura 15. Izquierda ventana de comandos. Derecha ventana “Simulink”.



Una vez tengamos la ventana del Simulink, procedemos a abrir la ventana del modelo para la simulación, esto se logra dando clic en el icono (*New Model*) ubicado en la parte superior izquierda de la barra de herramientas, figura 16.

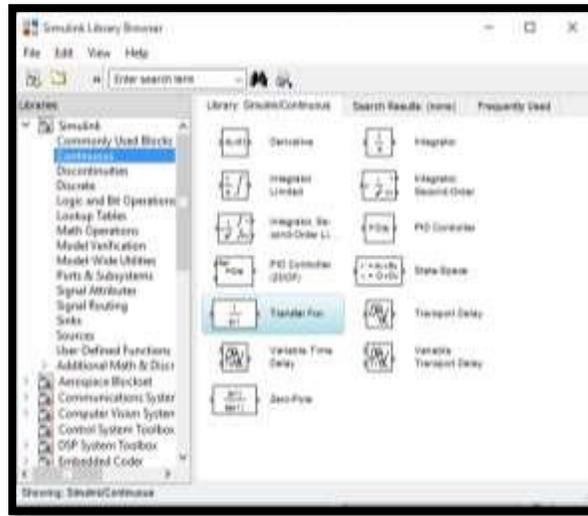
Figura 16. Izquierda ventana Simulink, derecha ventana de simulación.



Una vez abierta la ventana de simulación iniciaremos la creación de nuestro modelo.

Ingresamos los bloques “*Transfer Fcn*”, “*Step*” y “*Scope*” ubicados en el menú “*libraries*” ver figura 17.

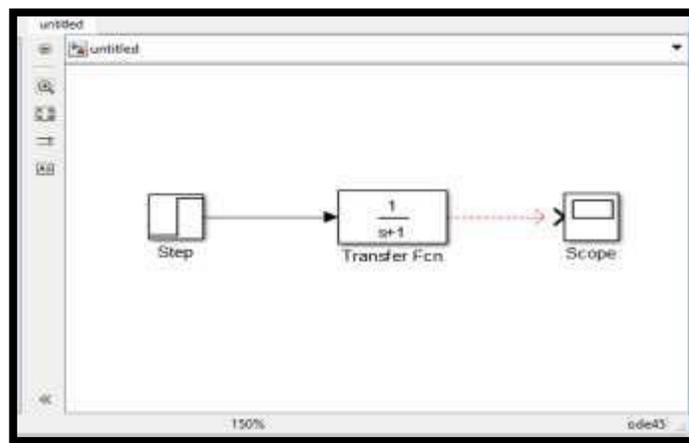
Figura 17. Ubicación del Bloque “Transfer Fnc” en el menú librerías.



NOTA: Para exportarlos bloques a la ventana de simulación, damos clic derecho sobre el bloque y en la opción “Add to untitled” hacemos clic.

Luego de haber insertados los bloque necesarios para la simulación del sistema, procedemos a interconectarlos entre sí. Para esto se deberá mantener oprimido el botón izquierdo del mouse sobre la saliente del bloque, luego arrástrelo hacia la entrada del siguiente bloque, ver figura 18.

Figura 18. Interconexión de los bloques.



Ahora se proceda a configurar cada uno de los bloques. Para esto deberá dar doble clic sobre cada bloque para desplegar la ventana de configuración de parámetros. La configuración del Bloque “Step” se hace a través de 3 parámetros principales como son:

Step time = Hace referencia al tiempo que tarda en generarse la señal paso, este valor por defecto es de 1 segundo.

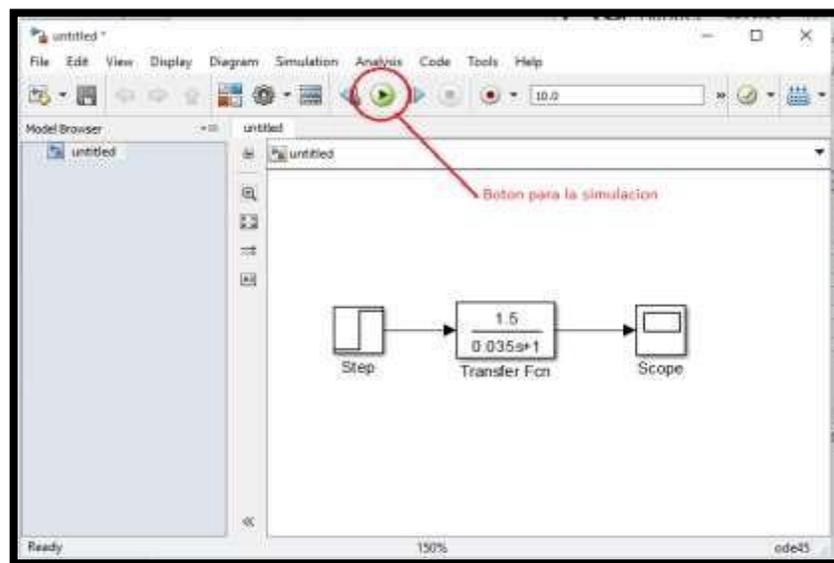
Initial Value = Valor inicial de la señal paso por defecto es 0.

Final Value = Valor final de la señal paso por defecto es 1.

El Bloque “*Transfer Fcn*” está compuesto por los parámetros “*Numerator coefficients*” y “*Denominator coefficients*”, los cuales hacen referencia a los datos del numerador y denominador de la función de transferencia del modelo identificado por el método “Ident”.

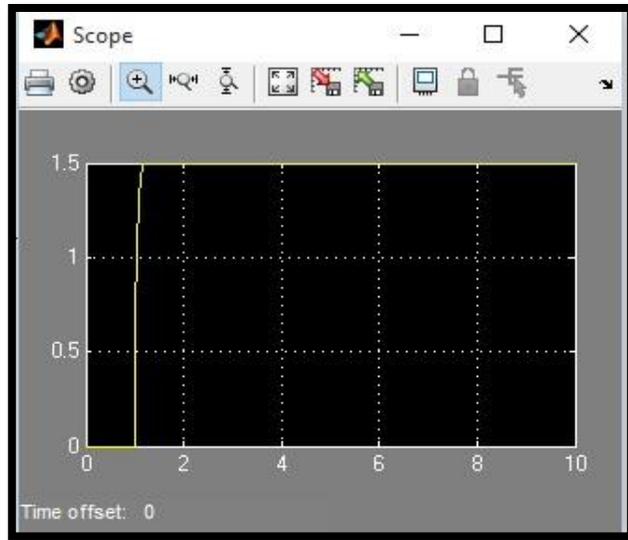
Una vez configurado los bloques se procede a realizar la simulación del sistema con el ámbito de representar gráficamente la respuesta temporal del modelo, esto se logra dando clic sobre el icono “Run” ubicado en la barra de herramientas de la ventana del simulador, figura 19.

Figura 19. Simulación en lazo abierto.



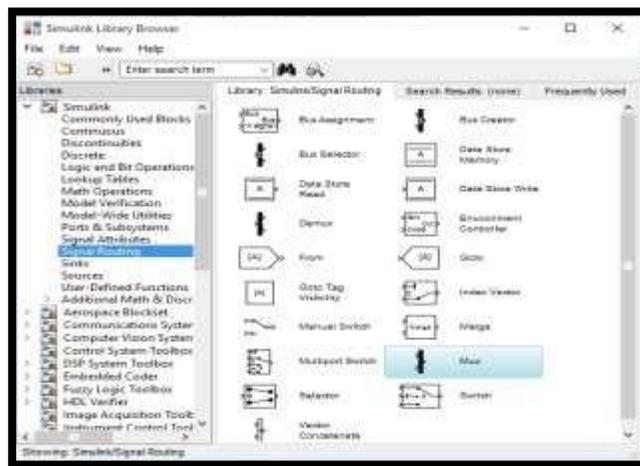
Haga doble clic sobre el bloque “Scope” para visualizar la respuesta del modelo una vez haya hecho la simulación, figura 20.

Figura 20. Respuesta del modelo.



Para visualizar la señal paso y la respuesta del modelo deberá ingresar el bloque “Mux” a la ventana de simulación, este bloque se halla localizado en “libraries>>Signal routing, figura 21.

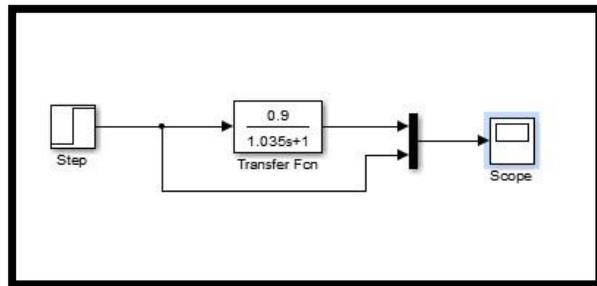
Figura 21. Ubicación del bloque Mux.



Una vez haya ingresado el bloque deberá borrar la conexión que hay entre el bloque “Transfer fcn” y el “Scope” , dando clic sobre la conexión seguido de la

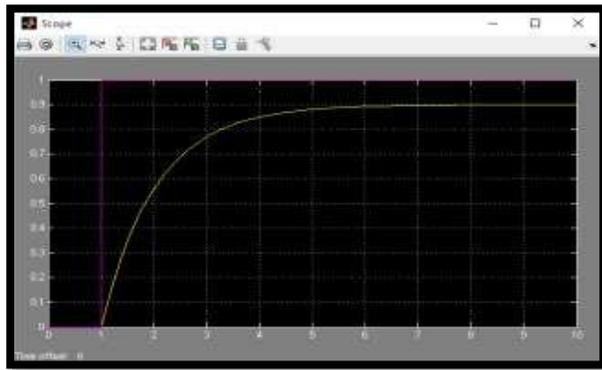
tecla Suprimir. Luego deberá configurar las conexiones del bloque “Transfer Fcn” y el “ Scope” a través del bloque “Mux”, figura 22.

Figura 22. Conexiones del bloque Mux.



Ejecute la simulación para visualizar a través del Scope el comportamiento de la señal paso y la respuesta temporal del modelo, figura 23.

Figura 23. Simulación de la respuesta del modelo ante una entrada escalón unitario. En color violeta la señal paso, en amarillo la respuesta del modelo.



Compare la respuesta temporal de la simulación con la respuesta del sistema real e ingrese los datos en la siguiente tabla.

Tabla 1. Características de la respuesta temporal del modelo real y el modelo Simulado.

| RESPUESTA TEMORAL | SIMULACION | REAL |
|--|-------------------|-------------|
| Orden del sistema (1, 2 o Superior) | | |
| Tiempo de retardo (Td) | | |
| Tiempo de subida (Ts) | | |
| Sobre Impulso (Os) | | |
| Tiempo de establecimiento(SS) | | |
| <i>Error en estado estacionario(e_{ss})</i> | | |

Ejercicio (para entregar próxima clase)

Considere la función de transferencia del modelo estimado hallado con la herramienta Ident de Matlab y diseñe un controlador PID con el método de lazo abierto de Ziegler Nichols primer método.

Simule el sistema de control en Simulink (con la función de transferencia) y evalúe el efecto de un cambio unitario en la referencia.

Ajuste las ganancias para obtener un Overshoot menor que 5% y un tiempo de establecimiento menor a 2 segundos.

GUIA No. 3
ACONDICIONAMIENTO DE SEÑAL E IDENTIFICACIÓN
DE MODELOS DINAMICOS

Instituto Tolimense de Formación Técnica profesional
Facultad de Ingeniería y Ciencias Agro-Industriales
Departamento de Electrónica Laboratorio de Control

INTRODUCCIÓN

Para el desarrollo de esta guía utilizaremos planta BMH (Balancín Motor Hélice), la cual integra la tarjeta de adquisición PTAD-S01. Y el instrumento virtual IVGARD para la generación y adquisición de señales. Aplicaremos un voltaje entre 0-5 voltios para hacer girar el motor y así mover el balancín en diferentes ángulos, luego por medio del bloque de acondicionamiento de señal representaremos la posición del balancín a partir de la adecuación de la señal que entrega el sensor, se analizará el comportamiento dinámico del sistema el cual es excitado por medio de una señal tipo escalón unitario y la curva de respuesta del sistema será registrada y almacenada para posteriormente ser procesada con la aplicación del toolbox "Ident" de Matlab que nos dará el modelo aproximado del sistema (función de transferencia).

El comportamiento de este modelo será simulado con la herramienta Simulink de Matlab para posteriormente ser comparada con el modelo real.

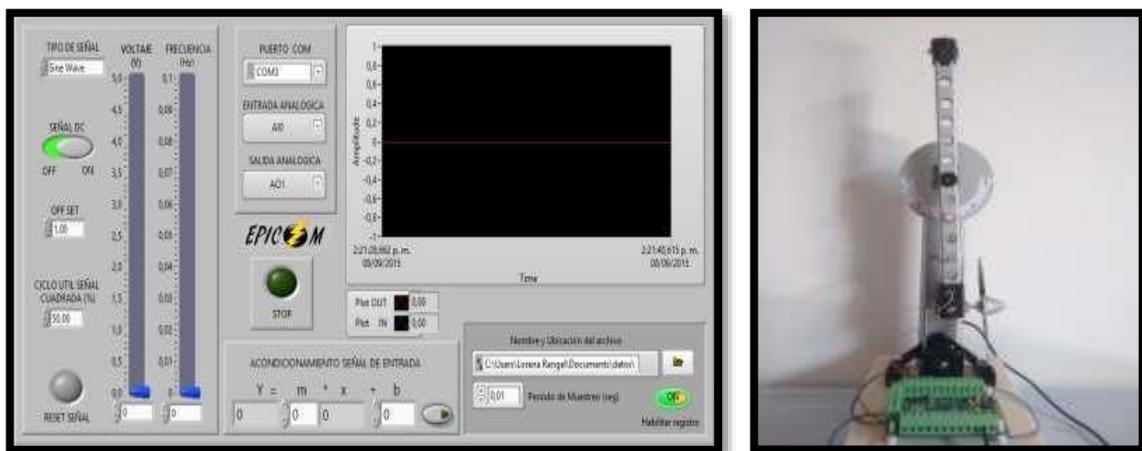
Objetivos

- Determinar la linealidad del sensor.
- Convertir del voltaje de salida del sensor en el ángulo de inclinación del balancín a través del bloque de acondicionamiento de señal.
- Aplicar los métodos de identificación no paramétricos
- Hallar la función de transferencia a través de datos experimentales.
- Verificar la validez del modelo físico con el modelo estimado

Equipo necesario

- Computadora con Windows 7, LabVIEW y Matlab 2013 o superior
- Tarjeta de adquisición PTAD-S01 + interfaz cable USB
- Planta BMH.
- Instrumento Virtual de generación de señales, adquisición y registro de datos IVGARD

Figura 1. Instrumento virtual (IVGARD) y Planta BMH.



DESCRIPCIÓN

PLANTA (BHM) La planta como se observa en la figura 1, está formada por un tubo rectangular en aluminio como balancín o barra móvil cuya longitud es de 30 cm y un bastidor cuya altura es de 25 cm, el balancín va unido a la bancada por medio del eje cursor del potenciómetro el cual permite el movimiento de giro. En uno de los extremos del balancín hay un soporte donde va colocado el motor y la hélice.

Acoplado al eje de giro mediante unos pequeños tornillos está fijado el cursor de un potenciómetro que actúa como sensor de posición. Es aquí donde comienza todo el flujo de datos, el sensor en este caso potenciómetro, es alimentado con la fuente de 5 Volts que proporciona la tarjeta de adquisición, de manera que a cada valor de tensión entre cursor y el común le corresponderá una posición determinada del balancín.

ACOPLE DE LA PLANTA A LA PC.

Conecte la planta a una toma de energía de 120 VAC a través de adaptador DC de la planta BMH.

EJECUCIÓN Y CONFIGURACIÓN DEL INSTRUMENTO VIRTUAL

Ejecute el programa para la Generación Adquisición y Registro de Datos (IVGARD), dando clic sobre el icono ubicado en el escritorio de Windows. Figura 2.

Figura 2. Icono



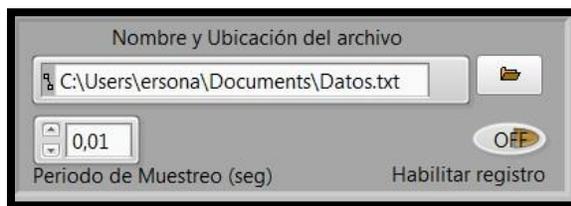
En el panel frontal del instrumento, configure en el bloque de periféricos el PUERTO COM, la ENTRADA ANÀLOGICA “AIO” y la SALIDA ANÁLOGA “A01”, figura 3.

Figura 3. Bloque periféricos.



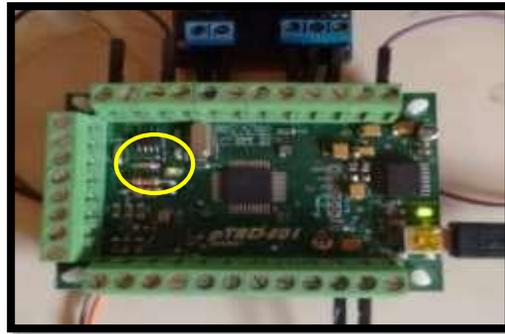
Configure el bloque registro de datos con su respectiva ruta y tiempo de muestreo, ejemplo figura. 4

Figura 4. Bloque registro de datos.



Conecte la planta BMH a la PC por medio del cable USB y verifique que el led de transmisión y recepción de datos de la tarjeta PTAD-S01 se halla encendido, figura 5.

Figura 5. Ubicación del led de transmisión y recepción de datos.



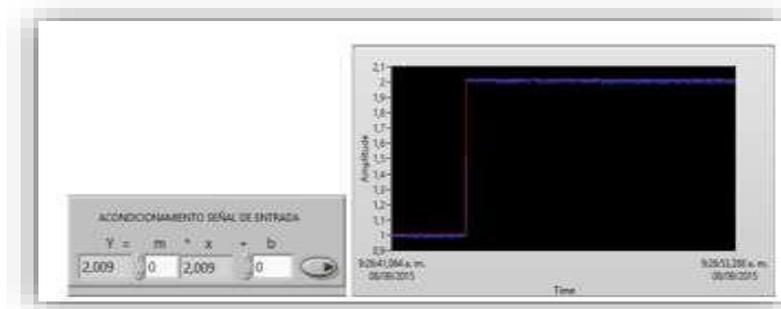
A través del bloque generador de señales se configuramos el sistema para generar una señal DC.

Prueba del sistema.

Realice una prueba del sistema aumentando gradualmente el voltaje por medio del slider identificado como Voltaje, este aumentara la velocidad del motor y por ende la posición del balancín. Nótese que el motor DC tiene una zona muerta, esto quiere decir que necesita de cierto valor de voltaje para romper su inercia.

Verifique por medio del indicador numérico identificado con la letra (x) ubicado en bloque de acondicionamiento de señal el voltaje que genera el sensor cada vez que cambia de posición el balancín, compare este valor con el valor que toma la señal en el bloque de visualización. Estos datos deberán ser iguales, ejemplo figura 6.

Figura 6. Representación gráfica y numérica del voltaje de la señal.



PASO 1. ACONDICIONAMIENTO DE SEÑAL .

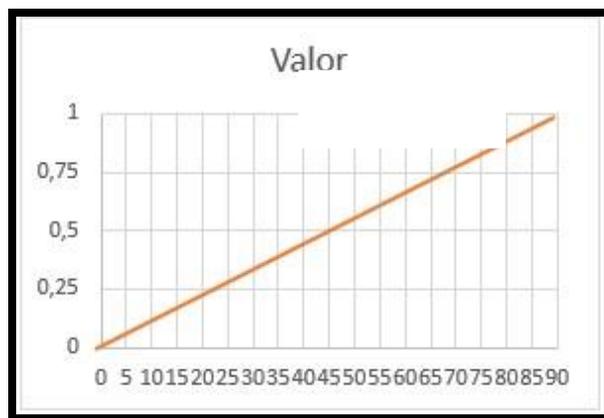
Ahora se procede a registrar los valores de voltaje que emite el sensor al ubicar el balancín de dentro de la escala ubicada en el bastidor. Tomar lectura moviendo el balancín cada 5 gradados hasta llegar a un Angulo de 90 grados, ejemplo figura 7.

Figura 7. Registro de datos. Voltaje del sensor vs Angulo de inclinación.



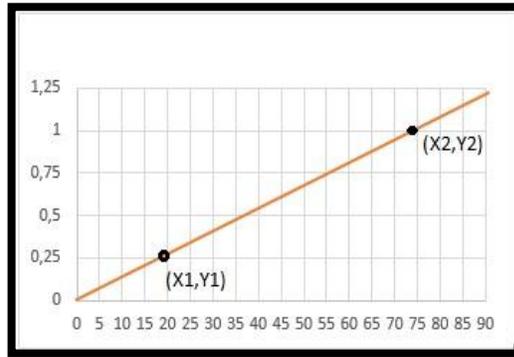
Con los datos ya registrados procedemos a determinar la linealidad del sensor graficando, por medio de un plano cartesiano la relación entre los vectores voltaje del sensor y el ángulo de inclinación, ejemplo figura 8.

Figura 8. Tabulación de los datos en el Plano cartesiano.



Luego de haber obtenido la recta se procede a hallar la pendiente, tomando como referencia los valores de (x , y) en dos puntos sobre la recta, ejemplo figura 9.

Figura 9. Vectores (X, Y).



Registre los valores que toma el punto (X1, Y1) y de (X2, Y2) en la siguiente tabla.

Tabla 1. Valores de los vectores.

| Vector | Valor |
|--------|-------|
| X1 | |
| X2 | |
| Y1 | |
| Y2 | |

Con los datos de la tabla 1, procedemos a hallar la pendiente de la recta reemplazando los valores de vectores (X1, Y1), (X2, Y2) en la siguiente ecuación.

$$m = \frac{Y_2 - Y_1}{X_2 - X_1}$$

Una vez obtenido el valor de la pendiente (m), procedemos a hallar la ecuación de la recta por medio de la siguiente ecuación.

$$y - y_1 = m(x - x_1)$$

Despejando nos queda que la ecuación de la recta es:

$$y = m * x + b$$

Donde:

Y = Es la señal del sensor acondicionada.

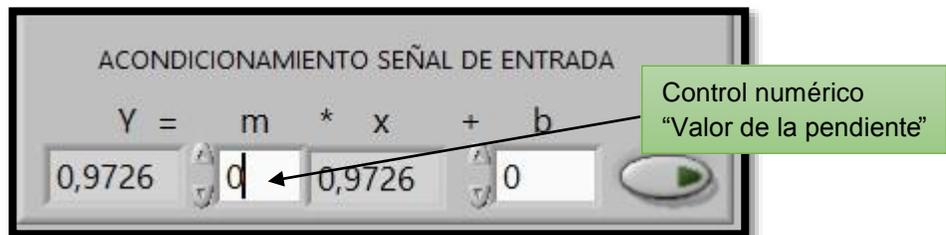
m = Es la pendiente de la recta.

X = Es la señal del sensor expresada en voltios.

b = Es el punto donde la recta se intersecta con el eje y.

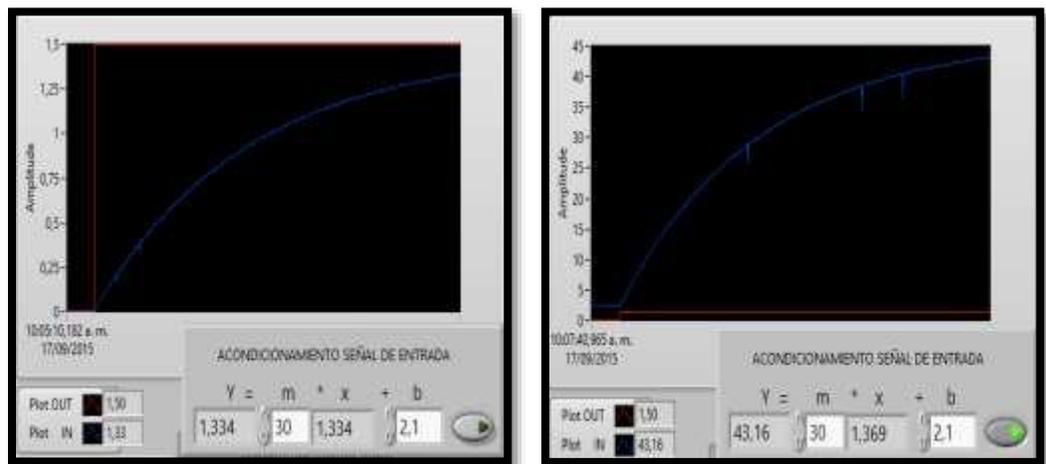
Luego ingresamos en el bloque de acondicionamiento de señal los valores de (m) y (b) obtenidos a partir de la ecuación de la recta, para esto debemos ubicar el cursor sobre en el control numérico identificado con la letra (m) y (b) y oprimir el clic izquierdo del mouse, luego con la ayuda del teclado numérico se ingresan los valores, ejemplo figura10.

Figura 10. Configuración del bloque de acondicionamiento.



Una vez se haya ingresado los datos de (m) y de (b) oprima el interruptor de acondicionamiento de señal para que se refleje el valor de la señal acondicionada (Y) en el bloque de visualización, ver ejemplo de la figura 11.

Figura 11. Ventana izquierda señal no acondicionada. Ventana derecha señal acondicionada.



Una vez configurado el bloque aumentamos el voltaje hasta ubicar el balancín en un ángulo de 39° , una vez posicionado verificamos que el dato mostrado por el instrumento coincida con la posición del balancín, ver figura 12. En caso contrario se deberá realizar nuevamente todos los pasos de acondicionamiento.

Figura 12. Visualización en el panel del instrumento la posición del balancín.

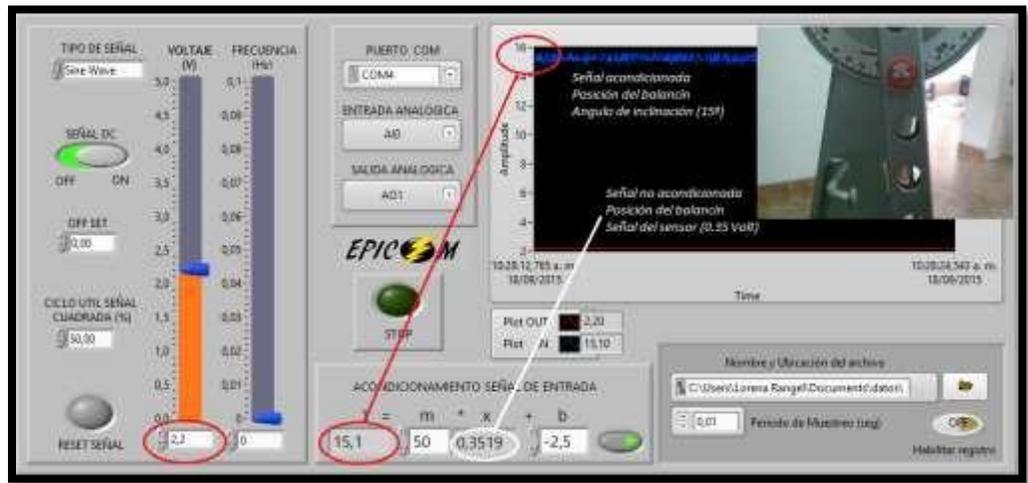


PASO 2. SEÑAL PASO Y REGISTRO DE DATOS.

Una vez que se haya configurado el bloque de acondicionamiento de señal y comprobado que los datos coinciden, se procede a realizar la identificación de nuestro modelo dinámico, registrando el comportamiento de la planta (BMH) cuando se excita con una señal escalón unitario, la cual es generada a través del instrumento.

Teniendo en cuenta que nuestro actuador (Motor DC) necesita de un voltaje mínimo para romper la inercia. Procedemos a aumentar el voltaje gradualmente hasta que el balancín se posicione en un ángulo de 15° .

Figura 13. Posición inicial del balancín.



Tome lectura del voltaje inicial a través del Display digital del slider de voltaje y del ángulo de inclinación por medio del indicador numérico (Y) ubicado en el bloque de acondicionamiento de señal y regístrelos en la tabla 2.

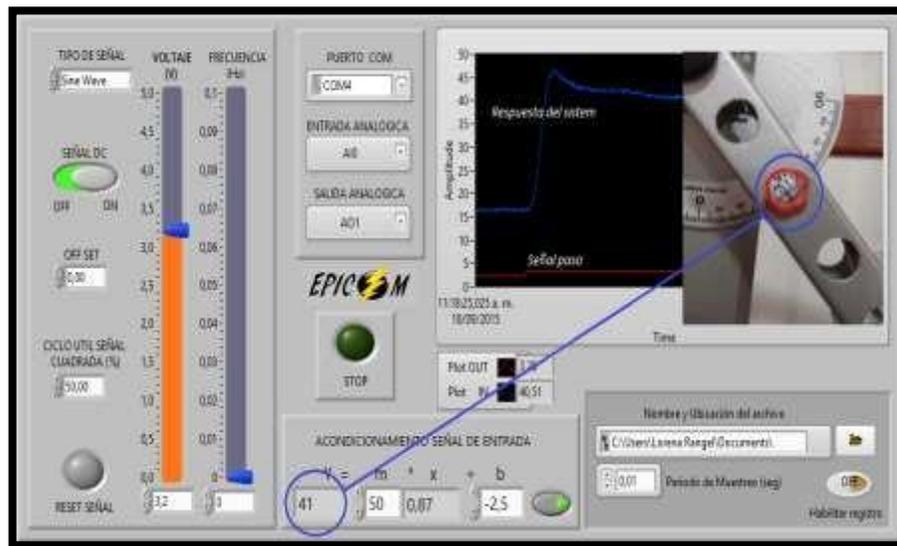
NOTA: Estos valores representan el offset de la señal tanto de entrada como de salida, ya que los valores iniciales para ambos casos no inician en cero (0) es decir, que el Offset para los datos de la señal paso es de 2.2, y para los datos de la respuesta del sistema es de 15, esto es debido a la zona muerta del motor.

Tabla 2. Valores del Offset.

| Voltaje inicial | Angulo inicial |
|-----------------|----------------|
| 2.2 | 15° |

Por medio del Display digital del slider de voltaje Generamos el escalón unitario cambiando el valor del voltaje inicial de 2.2 a 3.2, luego activamos el habilitador de registro esto hará que el balancín se mueva y cambien el Angulo de inclinación, figura 14.

Figura 14. Respuesta del sistema ante una entrada escalón unitario.



Una vez se establece el balancín, se deshabilita el registro de datos restableciendo la posición del interruptor “Habilitar registro”.

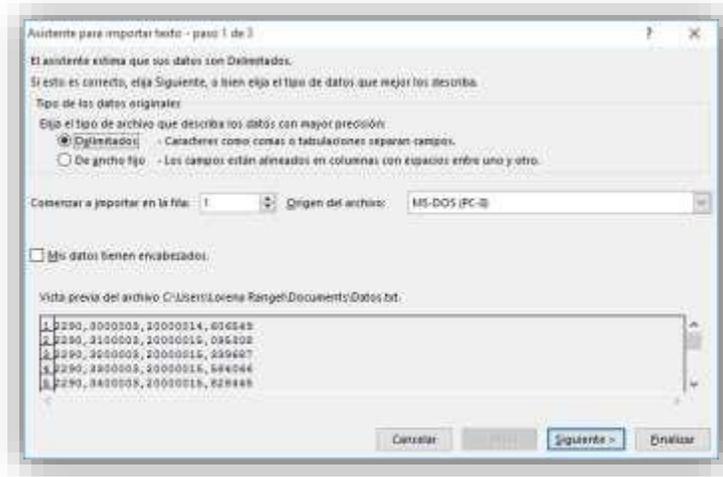
Constate que los datos se hallan registrado correctamente, abriendo el archivo “Datos.txt” ubicado en ruta especificada en el bloque de registró (Mis documentos/Datos.txt).

PASO 3. ADECUACIÓN DE LOS DATOS.

Para hacer de forma eficiente la identificación del sistema, debemos primero que todo remover el Offset a los datos de entrada (señal paso), como a los dantos de salida (Respuesta del sistema). Esto con el fin de que los datos tengan valores iniciales iguales a “0”.

Ejecutamos el programa Excel y en la barra de menú seleccionamos la opción abrir, la cual despliega una ventana donde nos pide el nombre del archivo que queremos importar. Seleccionamos el archivo ubicado en la carpeta mis documentos con el nombre Datos.txt. Luego en la ventana del “asistente para la importación de texto”, damos siguiente, siguiente y por ultimo finalizar, figura 15

Figura 15. Ventana del asistente para la importación de texto a Excel.



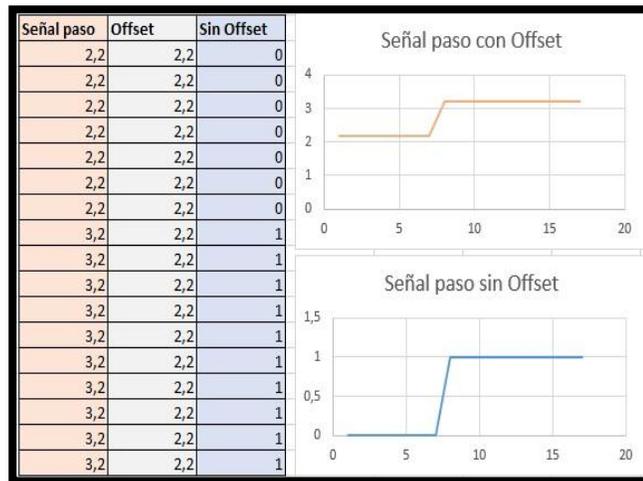
Una vez importados los datos procedemos a remover el Offset, para esto debemos crear 3 columnas las cuales las identificaremos con los nombres “Señal paso”, “Offset” y “Sin Offset”. Una vez creadas seleccionamos y copiamos la columna que contiene los datos de entrada que fueron importados y los pegamos en la columna “Señal paso”. Ahora la columna identificada como “Offset” la llenamos completamente con el valor de (2.2) que equivale al Offset de la señal paso. Luego para obtener la señal paso sin Offset debemos restarle a los valores de la columna “Señal paso” los valores de la columna “Offset”, para esto debemos escribir en la columna “Sin Offset” la siguiente instrucción: “Shift + = Señal paso – Offset” y oprimimos la tecla “Enter”, ver figura 16.

Figura 16. Aplicación de la instrucción a los datos de entrada.

| Señal paso | Offset | Sin Offset |
|------------|--------|------------|
| 2,2 | 2,2 | =B3-C3 |
| 2,2 | 2,2 | 0 |
| 2,2 | 2,2 | 0 |
| 2,2 | 2,2 | 0 |
| 2,2 | 2,2 | 0 |
| 2,2 | 2,2 | 0 |
| 3,2 | 2,2 | 1 |
| 3,2 | 2,2 | 1 |
| 3,2 | 2,2 | 1 |
| 3,2 | 2,2 | 1 |
| 3,2 | 2,2 | 1 |
| 3,2 | 2,2 | 1 |
| 3,2 | 2,2 | 1 |
| 3,2 | 2,2 | 1 |
| 3,2 | 2,2 | 1 |
| 3,2 | 2,2 | 1 |

En la figura 17, se muestra un ejemplo de la representación gráfica de la señal paso con Offset y sin Offset.

Figura 17. Representación gráfica de los datos.



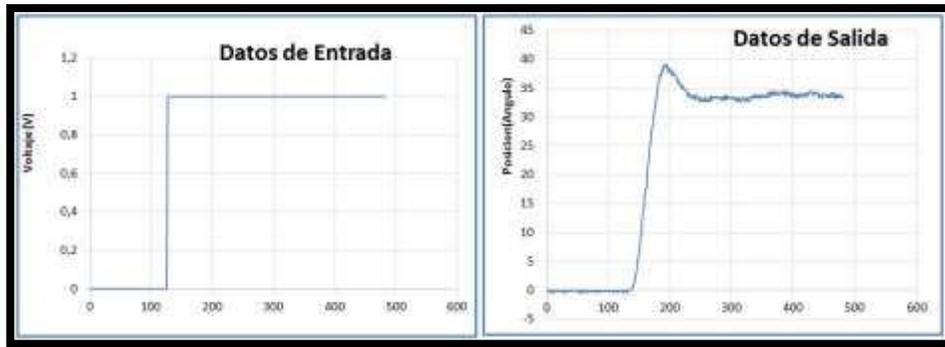
Una vez removido el Offset de la señal paso, se procede hacer el mismo procedimiento para los datos de salida (respuesta del sistema) teniendo en cuenta que el valor del Offset para estos datos de es de 15, figura 18.

Figura 18. Datos de salida.

| Señal de Salida | Offset | Salida sin Offset |
|-----------------|--------|-------------------|
| 15 | 15 | 0 |
| 15 | 15 | 0 |
| 15 | 15 | 0 |
| 15 | 15 | 0 |
| 15 | 15 | 0 |
| 15 | 15 | 0 |
| 15 | 15 | 0 |
| 15,05 | 15 | 0,05 |
| 15,07 | 15 | 0,07 |
| 15,08 | 15 | 0,08 |
| 15,09 | 15 | 0,09 |
| 15,11 | 15 | 0,11 |
| 15,12 | 15 | 0,12 |
| 15,14 | 15 | 0,14 |
| 15,15 | 15 | 0,15 |
| 15,17 | 15 | 0,17 |
| 15,18 | 15 | 0,18 |

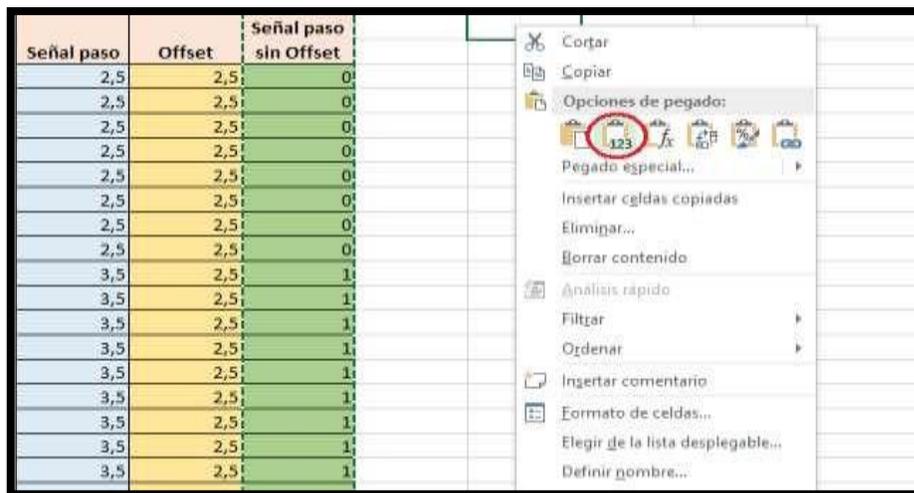
A continuación se da un ejemplo de los datos de entrada y salida a los cuales se les ha removido el Offset. Como se puede apreciar sus valores iniciales son iguales a "0", figura 19.

Figura 19. Representación de datos señal paso y la señal de respuesta sin offset.



Una vez removido el Offset, seleccionamos los datos dando clic sobre la columna "Señal paso sin Offset" y los copiamos oprimiendo (Ctrl + c), luego la pegamos en otra columna dando clic derecho sobre esta y en opciones de pegado seleccionamos 123, como se muestra en la figura 20.

Figura 20. Método para el pegado especial.



Realice los mismos pasos para los datos (Señal de salida). Al final deberá haber creado 2 columnas, la primera serán los datos de entrada (señal Paso) y la

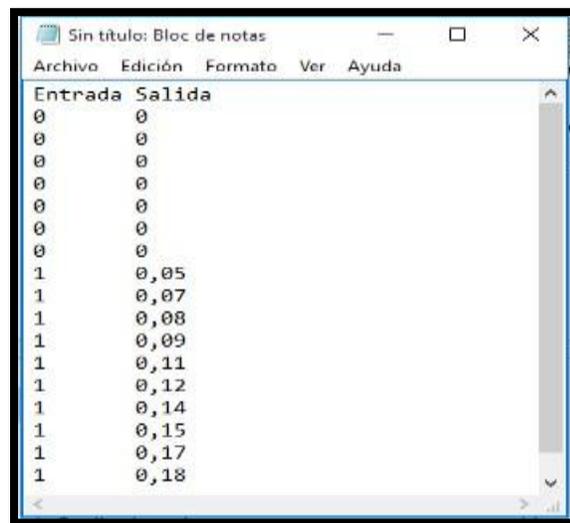
segunda los datos de salida (respuesta del sistema) sin Offset, ver ejemplo de la figura 21.

Figura 21. Datos de entrada y salida sin Offset.

| Entrada | Salida |
|---------|--------|
| 0 | 0 |
| 0 | 0 |
| 0 | 0 |
| 0 | 0 |
| 0 | 0 |
| 0 | 0 |
| 0 | 0 |
| 1 | 0,05 |
| 1 | 0,07 |
| 1 | 0,08 |
| 1 | 0,09 |
| 1 | 0,11 |
| 1 | 0,12 |
| 1 | 0,14 |
| 1 | 0,15 |
| 1 | 0,17 |
| 1 | 0,18 |

Seleccione las 2 columnas (Entrada, salida) y cópielas haciendo (Ctrl+c), luego abra el bloc de notas de Windows y péguelas en la ventana haciendo (Ctrl+v), ver figura 22.

Figura 22. Ventana del bloc de notas con los datos de entrada y salida sin Offset.



En la barra de menú seleccione archivo>>guardar cómo y en la carpeta mis documento lo guarda con el nombre Datos_BMH.txt.

PASO 4. PROCESAMIENTO DE DATOS Y ANÁLISIS DE LA RESPUESTA TEMPORAL .

Importe los datos de entrada y salida con la herramienta “Import data” de Matlab.

Exporte los datos al “WorkSpace” a través del “Import Selection” ubicada en la barra de herramientas de la ventana del “Import data”.

Verifique en el “WorkSpace” que se hayan creado las variables “Entrada”, “Salida”,

Figura 23. Variables de Entrada y salida el “Workspace”.

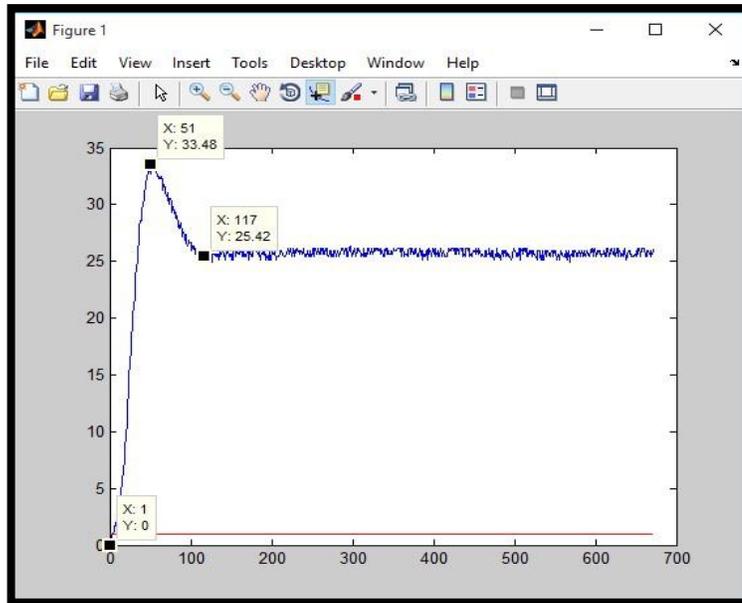


| Name ▲ | Value | M |
|---------|----------------|---|
| Entrada | <672x1 double> | 1 |
| Salida | <672x1 double> | 0 |

Para visualizar los datos de entrada y salida en una misma gráfica, Digite en el “Command window” la siguiente instrucción:

```
plot(Entrada,'r')  
hold on  
plot(Salida,'b')
```

Figura 24. Representación gráfica de los datos. En rojo la señal paso, en azul la respuesta del sistema.

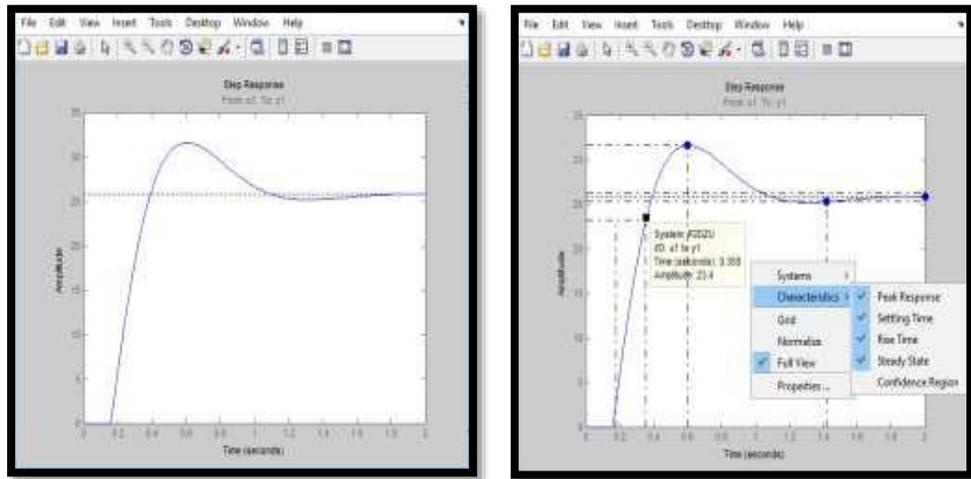


NOTA: En la GUIA N° 1, se explica detalladamente la forma de llevar a cabo el paso 4.

Análisis de la respuesta temporal.

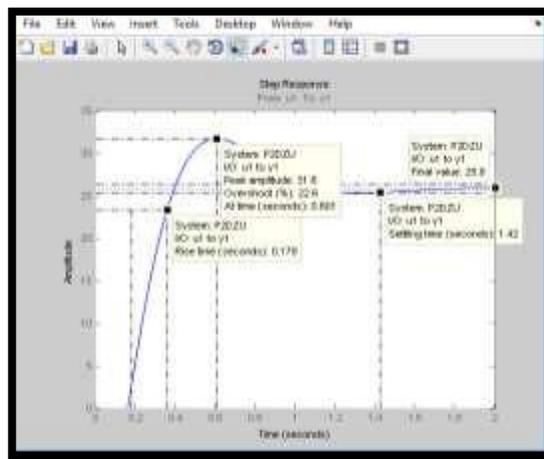
Una vez hallada la función de transferencia del modelo por el método Ident y exportada al "Workspace" se procede a graficar la respuesta del modelo ante una entrada escalón unitario para su análisis. Para esto digitamos en el "Command Window" de Matlab la siguiente instrucción "step(P2DU)", la cual graficará la respuesta del modelo. Para visualizar las características de la respuesta temporal en la gráfica, ubicamos el cursor sobre esta y damos clic derecho, en características damos clic en todas las opciones, figura 25.

Figura 25. Plot de la respuesta del modelo.



Ahora por medio del “Data cursor” ubicado en la barra de herramientas del plot, colocamos los “Datatips” (etiqueta) sobre los puntos, con el fin de visualizar sus valores, ver figura 26.

Figura 26. Características de la respuesta temporal.



PASO 5. IDENTIFICACIÓN DEL SISTEMA MÉTODO “IDENT”.

Una vez procesados los datos, se procede a identificar la función de transferencia del modelo, utilizando el Toolbox “Ident” de Matlab.

Para generar la ventana de la herramienta “Ident” Ubíquese en el “Command window” de Matlab y digite la siguiente instrucción: >> Ident, seguido de la tecla Enter.

NOTA: En la GUIA N° 2, paso 5 se explica detalladamente la forma de llevar la identificación del sistema aplicando el método “Ident” de Matlab.

PASO 7. SIMULACIÓN.

Con la ayuda de Simulink construya el bloque de funciones en lazo abierto con el modelo estimado, aplique un escalón unitario y compare la respuesta del sistema simulado con los datos del sistema real y regístrelos en la siguiente tabla.

Tabla 3. Características de la respuesta temporal del modelo real y el modelo simulado.

| RESPUESTA TEMORAL | Simulación | Real |
|--|------------|------|
| Tiempo de retardo (Td) | | |
| Tiempo de subida (Ts) | | |
| Sobre Impulso (Os) | | |
| Tiempo de establecimiento(SS) | | |
| Error en estado estacionario(e_{ss}) | | |

NOTA: En la GUIA 2, ejercicio 6 puntos del 1 al 15 se explica detalladamente la forma de llevar a cabo la simulación.

GUIA No. 4

SINTONIZACIÓN DEL CONTROLADOR PID DISCRETO

Instituto Tolimense de Formación Técnica profesional

Facultad de Ingeniería y Ciencias Agro-Industriales

Departamento de Electrónica Laboratorio de Control

INTRODUCCIÓN

En esta práctica se diseñara el controlador PID para controlar el ángulo de inclinación del sistema BMH (Balancín Motor hélice) se analizara su función dentro del lazo de control, se comprobara su comportamiento con señales típicas de prueba como es la señal paso y se hará la validación del modelo electromecánico Balancín Motor Hélice por medio de la interfaz gráfica IGPMC.

OBJETIVOS

- Diseñar un regulador en lazo cerrado clásico tipo PID y analizar la estabilidad del sistema completo en lazo cerrado.
- Realizar la implementación completa de un sistema de control discreto, incluyendo la interfaz analógica-digital.
- Simular el sistema de control en lazo cerrado con Simulink y comparar su comportamiento con el sistema real.

EQUIPO NECESARIO

- PC
- Planta BMH
- interfaz gráfica para el monitoreo y control (IGPMC)
- Matlab con Simulink

Figura 1. Interfaz gráfica de control IGPMC y la planta BMH.



DESCRIPCIÓN DEL SISTEMA.

INTERFAZ GRÁFICA PARA EL MONITOREO Y CONTROL (IGPMC) es la encargada de controlar y suministrar la información necesaria al usuario del comportamiento del proceso, aquí se asignan los datos necesarios para llevar a cabo el control de la variable. Está compuesto principalmente por indicadores gráficos en el que se pueden visualizar el error en estado estacionario (ess), el Set Point vs la posición del balancín (SET POIN-VARIABLE) y la acción de control (ACCION DE CONTROL), además cuenta con controles tipo slider, en él que ajustan las ganancias del controlador (KP, KI, KD), el tiempo de muestreo (ST), asignar el SET POINT (SP) además cuenta con controles tipo texto, donde se configuran los periféricos como son: el canal de entrada análoga (AI0-AI7), salida análoga (AO0-AO1) y el puerto de comunicación (COM).

PROCEDIMIENTO.

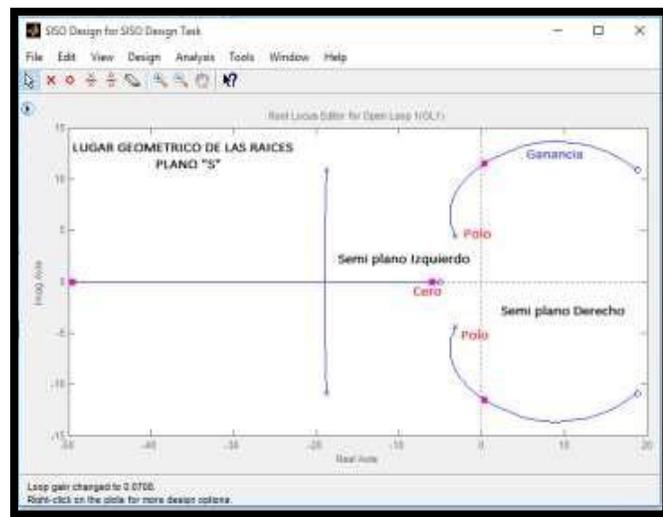
PASO 1. SINTONIZACIÓN.

Para la sintonización del controlador, se procede a aplicar el método 2 de Ziegler-Nichols para señales oscilantes. La técnica determina que debemos aplicar una ganancia hasta que el sistema tenga oscilaciones permanentes, esto con el fin de determinar los valores de la ganancia crítica y el periodo crítico.

NOTA: Si la salida no presenta oscilaciones sostenidas para cualquier valor que pueda tomar K_p , entonces este método no se puede aplicar. Siendo Así, la ganancia crítica (K_{cr}) y el periodo (P_{cr}) correspondiente se determinan de forma experimental.

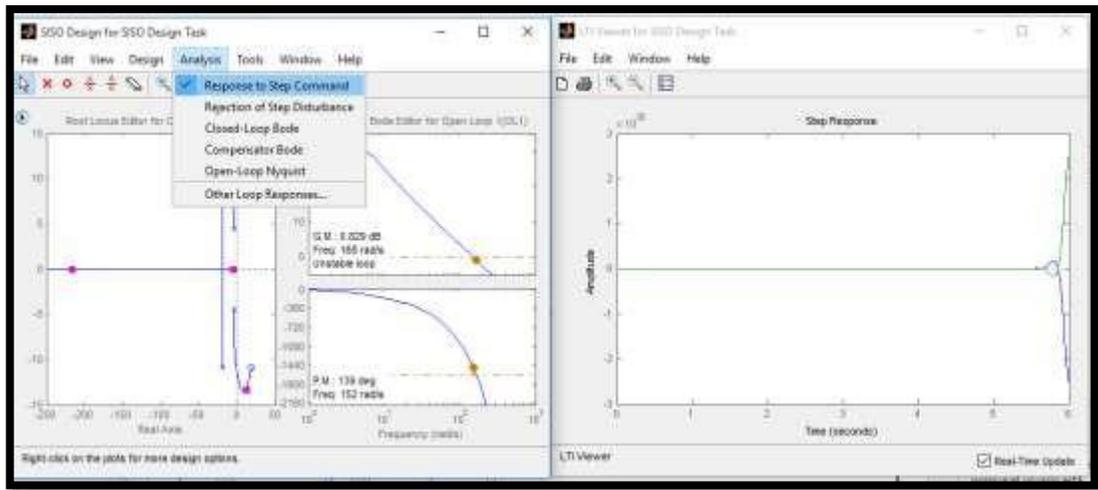
Para hallar la ganancia crítica (K_{cr}) y el periodo crítico (P_{cr}) abrimos la ventana del SISOTOOL escribiendo en el "Command Window" la siguiente instrucción: `>>sisotool("variable")` el cual nos arroja la ventana del Toolbox con la ubicación de los polos y Ceros de nuestro modelo, ver figura 2.

Figura 2. Ventana del toolbox SISOTOOL y la ubicación de polos y ceros en el plano "S".



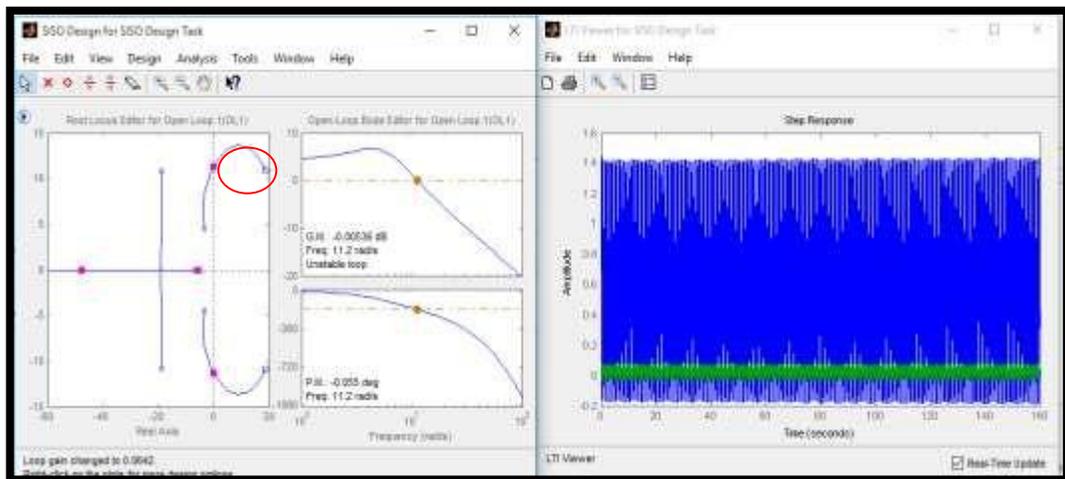
Para visualizar el comportamiento de la señal al momento de ajustar la ganancia, abrimos la ventana de la respuesta paso. Para esto nos ubicamos en la barra de menús de la ventana del Toolbox y damos clic en "Analysis" opción "Response to Step Command", ver figura 3.

Figura 3. Ventana de la respuesta paso.



Para ajustar la ganancia ubicamos el cursor sobre el punto de color magenta localizado en el semiplano derecho del plano “S” y con la ayuda del mouse lo arrastramos hacia el semiplano izquierdo hasta obtener oscilaciones permanentes, ver figura 4.

Figura 4. Ajuste de la ganancia K_p para obtener oscilaciones permanentes.



Luego de haber obtenido las oscilaciones permanentes se toma nota del valor de ganancia crítica (K_{cr}) ubicada en la parte inferior izquierda de la ventana del Root locus y de la frecuencia crítica (F_{cr}) en la ventana del bode, esto con el fin de aplicar la regla de sintonías de ZN método 2 y obtener las ganancias del controlador PID, ver tabla 1.

Tabla 1. Reglas de sintonía de Ziegler-Nichols método 2.

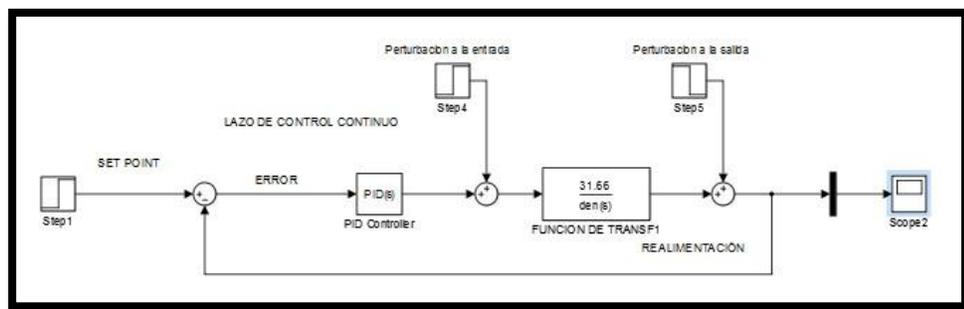
| Tipo de controlador | K_p | T_i | T_d |
|---------------------|--------------|-----------------------|---------------|
| P | $0.5K_{cr}$ | ∞ | 0 |
| PI | $0.45K_{cr}$ | $\frac{1}{1.2}P_{cr}$ | 0 |
| PID | $0.6K_{cr}$ | $0.5P_{cr}$ | $0.125P_{cr}$ |

NOTA: Para poder hallar el periodo crítico (P_{cr}), la frecuencia crítica (F_{cr}) deberá ser expresada en Hertz

PASO 2. SIMULACIÓN

Con ayuda de Simulink implemente el lazo de control. Configure los parámetros del bloque de control PID con las ganancias obtenidas por método 2 de ZN y del bloque "Transfer fcn", figura 5.

Figura 5. Lazo de control con perturbaciones a la entrada y salida en Simulink.



Evalué el efecto de un cambio unitario en la referencia y un cambio tipo escalón en la señal de perturbación a la entrada y salida. Evalué el comportamiento de la respuesta temporal del sistema.

Convierta las ganancias del controlador continuo (s) a discreto (z) y regístrelos en la tabla 2.

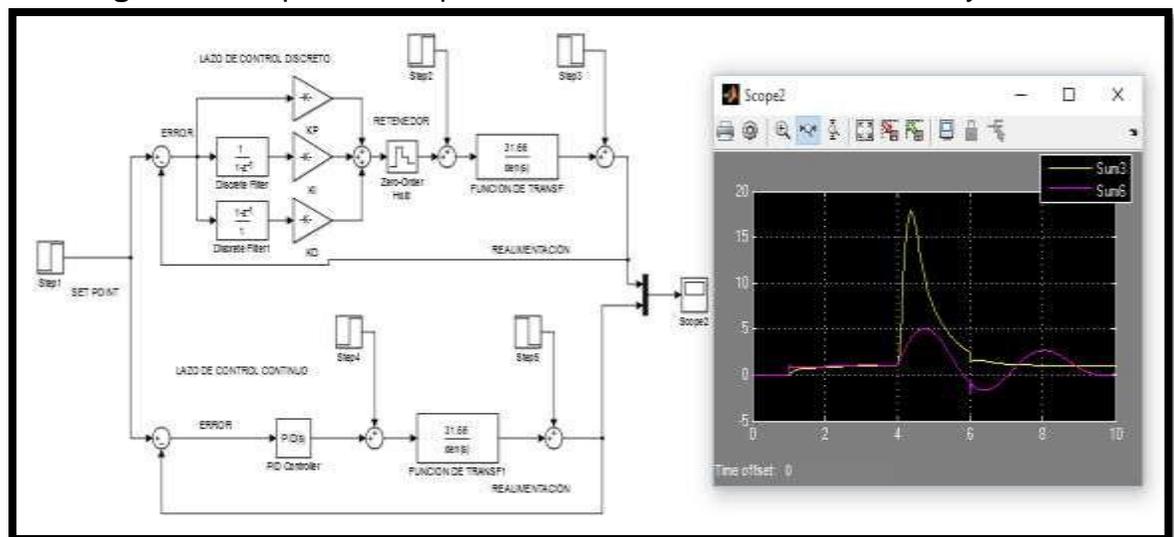
Tabla 2. Ganancias de controlador PID.

| Ganancias del controlador | Continuo (Kp, Ti, Td) | Discreto (KP, KI, KD) |
|---------------------------|-----------------------|-----------------------|
| Ganancias proporcional | | |
| Ganancia integral | | |
| Ganancia derivativa | | |

NOTA: tenga en cuenta que para hacer la conversión del controlador en tiempo continuo a tiempo discreto deberá tener el valor del tiempo de muestreo, este valor se obtiene dividiendo por 10 el “Rise Time” (tiempo de subida), el cual fue hallado en el paso 6 de la GUIA 3.

Con la ayuda de Simulink construya el sistema de control discreto en lazo cerrado y compare la respuesta de los dos sistemas ver el ejemplo de la figura 6.

Figura 6. Respuesta temporal de los lazos de control discreto y continuo.



PASO 3. CONTROL MODELO REAL

Conecte la planta a una toma de energía de 120 VAC.

Conecte la planta a la PC a través del cable USB.

Ejecute el programa dando doble clic sobre el icono ubicado en escritorio con el nombre de “IGPMC”

Figura 7. Interfaz de monitoreo y control planta BHM.



PASO 4. CONFIGURACIÓN DE LA INTERFAZ GRÁFICA.

- Entrada Análoga (AI0) y salida Análoga (AO1)
- Puerto de comunicaciones (COM) depende de la PC
- Tiempo de muestreo (ST)
- Ganancias del controlador discreto (KP, KI, KD) los hallado en el paso 2. Set Point (SP) este iniciara en 0.

Ingrese valores en la referencia (SP) y visualice el comportamiento del sistema a través de los indicadores grafico de la interfaz.

Compare la respuesta temporal del sistema real con el sistema simulado, para ver si tienen similitud

Ajuste las ganancias del controlador para obtener mejores resultados.

Haga un cuadro comparativo entre el sistema real y el simulado

Tabla 3. Respuesta temporal del modelo real y el modelo simulado.

| Ganancias PID discreto | RESPUESTA TEMORAL | Simulación | Real |
|------------------------|--|------------|------|
| | Tiempo de retardo (Td) | | |
| | Tiempo de subida (Ts) | | |
| | Sobre Impulso (Os) | | |
| | Tiempo de establecimiento(SS) | | |
| | Error en estado estacionario(e_{ss}) | | |

¿Cuál es comportamiento del sistema al aumentar y disminuir el tiempo de muestreo (ST)?

Paso 8. Presente un informe detallado de la práctica con sus conclusiones, del comportamiento del sistema real y el simulado en formato IEEE.

GUIA No. 5

IDENTIFICACION, SINTONIZACIÓN Y CONTROL PLANTA BMH SIMULINK-ARDUINO

Instituto Tolimense de Formación Técnica profesional
Facultad de Ingeniería y Ciencias Agro-Industriales
Departamento de Electrónica Laboratorio de Control

INTRODUCCIÓN

A través de esta guía el estudiante hará la identificación de un sistema a partir de datos experimentales. Se excitará la planta BMH con una señal paso generada a través del pin 2 de la salida PWM de la tarjeta Arduino mega y la curva de respuesta del sistema será registrada por medio de la entrada análoga 0 de la placa Arduino y enviada al "Workspace" a través del bloque "Scope". Luego los datos son procesados con la herramienta Ident de Matlab que nos dará el modelo estimado del proceso. Una vez obtenida la función de transferencia se procede a realizar la sintonización del controlador aplicando el método 2 de ZN que nos dará las ganancias del controlador PID. Estas ganancias se ingresan a la interfaz de monitoreo y control para llevar a cabo el control de la posición del balancín. Al final se hará una comparación entre el modelo simulado y el real.

OBJETIVOS

- Generar y adquirir señales a través de la tarjeta Arduino mega 2560.
- Almacenar y manipular los datos de señales adquiridas por el sistema Simulink - Arduino
- Aplicar los métodos de identificación no paramétricos
- Hallar la función de transferencia a través de datos experimentales.
- Verificar la validez del modelo físico con el modelo estimado

EQUIPO NECESARIO

- Computadora con Windows, Matlab / Simulink support package for arduino.
- Planta BMH
- Tarjeta de adquisición Arduino mega 2560
- Cable USB
- Interfaz de monitoreo y control Simulink-Arduino
- Instrumento para de generación de señales, Adquisición y registro de datos Simulink- Arduino.

Figura 1. Interfaz gráfica de monitoreo y control y planta BMH.



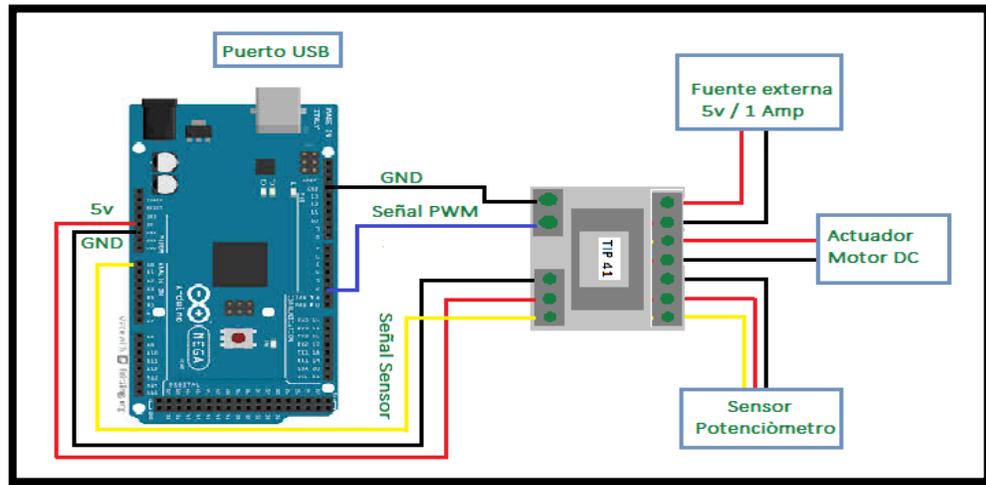
DESCRIPCION DEL SISTEMA.

INTERFAZ GRÁFICA PARA EL MONITOREO Y CONTROL es la encargada de controlar y suministrar la información necesaria al usuario del comportamiento del proceso, aquí se asignan los datos necesarios para llevar a cabo el control de la variable. Está compuesto principalmente por indicadores gráficos en el que se pueden visualizar el error en estado estacionario (Ess), la referencia y la señal del proceso, además cuenta con controles tipo slider, en el que se pueden ingresar los valores de referencia (Set point) y ajustar las ganancias del controlador (Kp, Ti, Td) en tiempo real.

PLANTA (BHM) La planta como se observa en la figura 1, está formada por un tubo rectangular en aluminio como balancín o barra móvil cuya longitud es de 30 cm y un bastidor cuya altura es de 25 cm, el balancín va unido a la bancada por medio del eje cursor del potenciómetro el cual permite el movimiento de giro. En uno de los extremos del balancín hay un soporte donde va colocado el motor y la hélice.

El potenciómetro es alimentado con la fuente de 5 Volts que proporciona la placa Arduino mega 2560 y su salida esta acoplada a la entrada analógica 0 de la tarjeta. La salida PWM pin 2 de la tarjeta Arduino esta acoplada al transistor de potencia (TIP41) encargada de conmutar la señal PWM y generar el empuje necesario para hacer girar el motor.

Figura 2. Diagrama eléctrico de la tarjeta Arduino mega 2560 y planta BMH.



PASO 1. ACOPLA DEL SISTEMA.

Conecta la planta a la toma de 110 v AC

Por medio del cable USB conecte la tarjeta a la PC

PASO 2. EJECUCION DEL PROGRAMA PARA LA ADQUICION DE DATOS

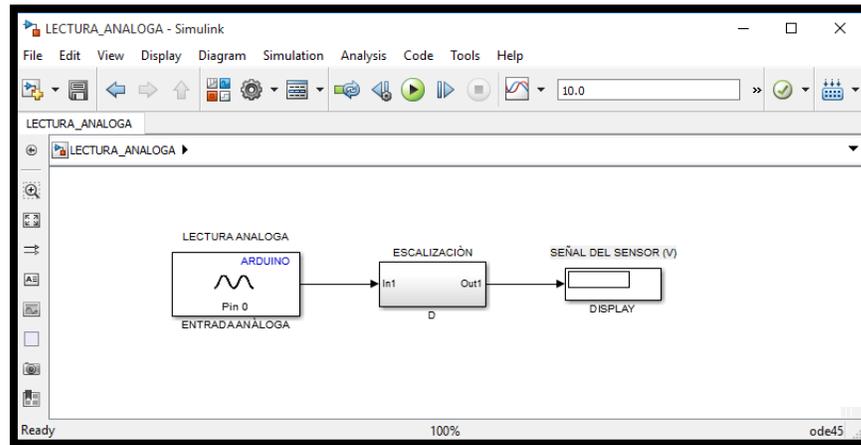
Ejecute el programa MATLAB dando doble clic sobre el icono ubicado en el escritorio, figura 3.

Figura 3. Icono de Matlab.



Una vez abierta la ventana de Matlab nos ubicamos sobre la barra de herramientas y en la opción "Open file" damos doble clic. Esta acción nos lanza la ventana del "Open" donde debemos seleccionar el archivo "LECTURA_ANALOGA.xls" ubicado en el escritorio dentro de la carpeta ARDUINO-SIMULINK, figura 4.

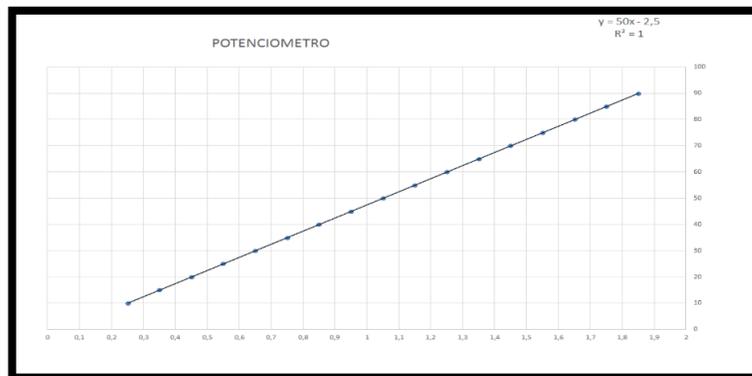
Figura 4. Diagrama de bloques para la toma de datos.



Al correr el programa el sistema se enlaza automáticamente con la tarjeta Arduino, la cual empezará a leer los datos generados por el sensor, estos datos son visualizados a través de bloque Display.

Tome lecturas del sensor con respecto al Angulo de inclinación cada 5 grados, una vez tabulados los datos gráfíquelos en un plano cartesiano o por medio de la herramienta dispersión en Excel y obtenga la ecuación de la recta, ver ejemplo de la figura 5.

Figura 5. Representación de la recta en el plano cartesiano a través de la herramienta dispersión de Excel.



Para el ejemplo de la figura 5, la ecuación de la recta queda: $y = m \cdot x + b$

Donde:

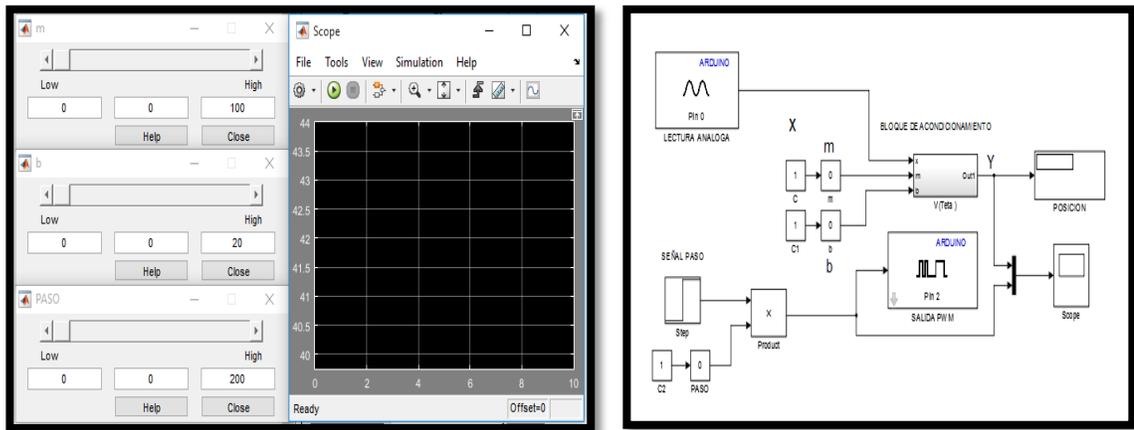
$m = 50$

$b = -2.5$

PASO 3. ACONDICIONAMIENTO DE SEÑAL SIMULINK

Una vez obtenida la ecuación de la recta proceda a lanzar la aplicación “ADQ_ARDUINO” ubicado en el escritorio dentro de la carpeta SIMULINK-ARDUINO. Con los datos obtenidos de la ecuación de la recta, se configuran los valores del bloque de acondicionamiento (m) y (b) ubicados en el panel frontal, figura 6.

Figura 6. Panel frontal y diagrama de bloques del sistema DAQ_ARDUINO.



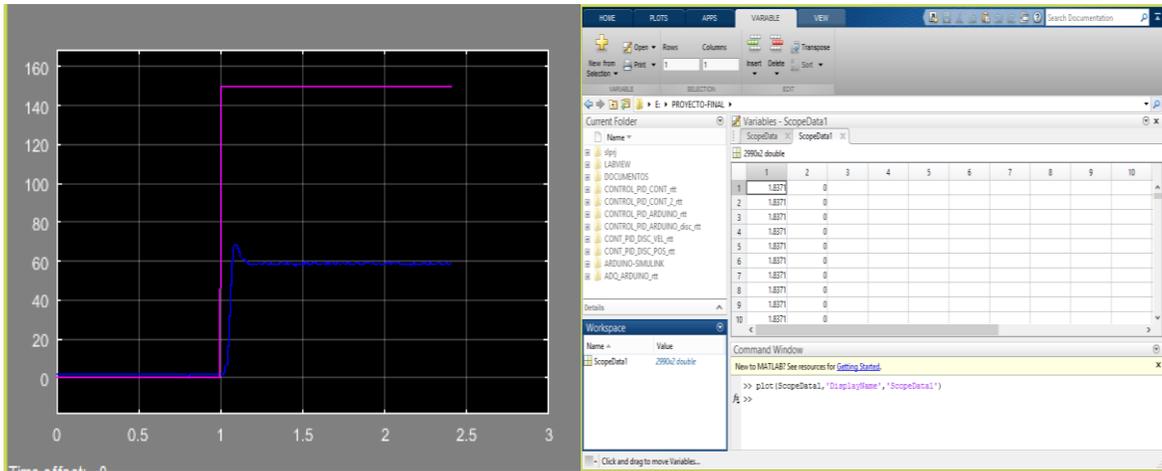
Una vez configurado el bloque corra el programa. Ubique el balancín en un ángulo específico, luego compare esta posición con el dato mostrado en el Display, estos datos deben coincidir. En caso contrario deberá realizar de nuevo el ejercicio.

PASO 4. ADQUISICIÓN Y REGISTRO DE DATOS SIMULINK

Luego de haber configurado el bloque de acondicionamiento de señal (bloques m y b) Configure el sistema para generar una señal escalón de 150 equivalente al 58.5 % del ciclo útil (Duty Cycle) de la señal PWM, a través del bloque paso ubicado en el panel frontal mostrado en la figura 6, luego ejecute el programa para generar la señal paso.

El registro de los datos se hace por medio del bloque Scope que además de visualizar las señales de entrada y salida se configura para que los datos sean exportados automáticamente al “Workspace” al parar el programa, figura 7.

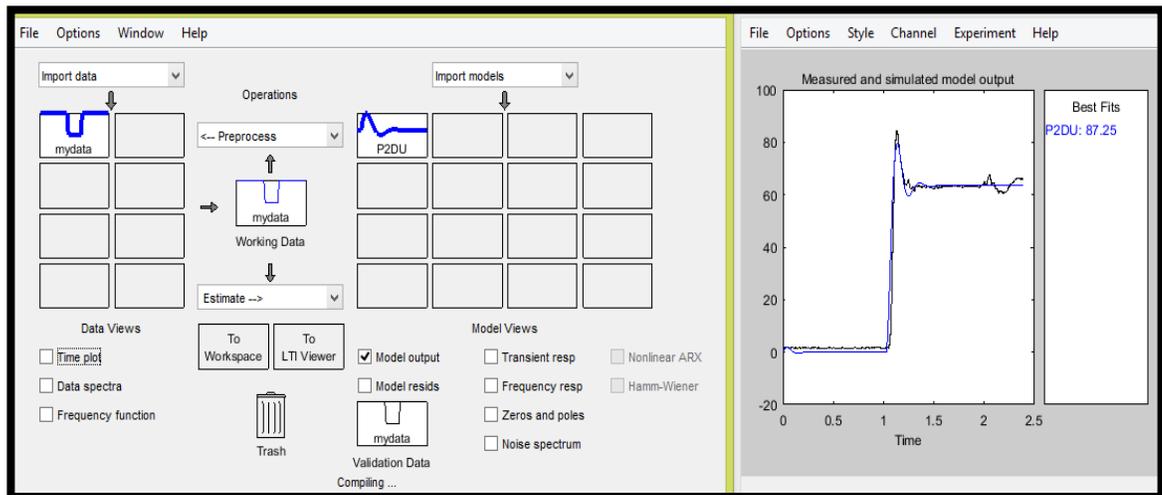
Figura 7. Datos importados al Workspace a través de bloque “Scope” de Simulink.



PASO 5. IDENTIFICACIÓN DEL MODELO (FUNCION DE TRANSFERENCIA) IDENT/MATLAB.

Una vez se hayan registrado los datos procede a realizar la identificación de nuestro modelo por medio de la herramienta Ident de Matlab, figura 8.

Figura 8. Ventana del Ident. Modelo estimado.



Si la estimación supera el 76% de efectividad de como identificado y validado el modelo, de lo contrario realice de nuevo la identificación. Para obtener la función de transferencia del modelo envíe la variable que contiene los datos (P2DU) al “WorkSpace” de Matlab, luego digite la instrucción `>> tf (P2DU)` en la ventana del Command Window, figura 9.

Figura 9. Ventana del Command Window con la Función de transferencia del modelo.

```
>> tf(P2DU)

ans =

From input "u1" to output "y1":|
                                0.423
exp(-0.03*s) * -----
                   0.001039 s^2 + 0.02597 s + 1

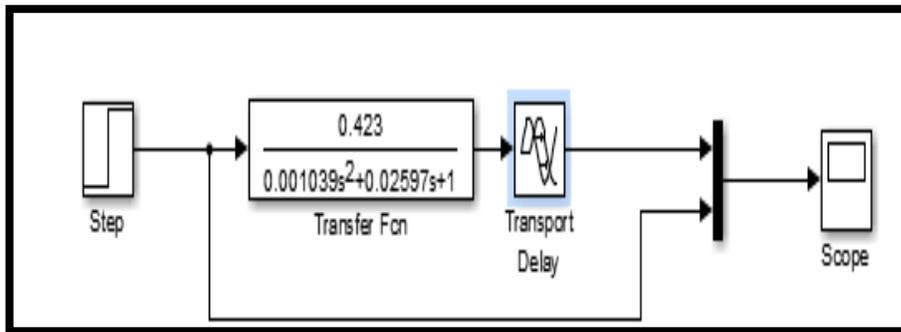
Name: P2DU
Continuous-time transfer function.

>> step(P2DU)
>> |
```

PASO 6. SIMULACION EN LAZO ABIERTO.

Una vez hallada la función de transferencia proceda a simular el comportamiento nuestro modelo implementado un diagrama de bloques en Simulink, donde la planta es excitada con una entrada escalón de 150, ver figura10.

Figura 10. Respuesta modelo visualización de la señal de entrada y salida.

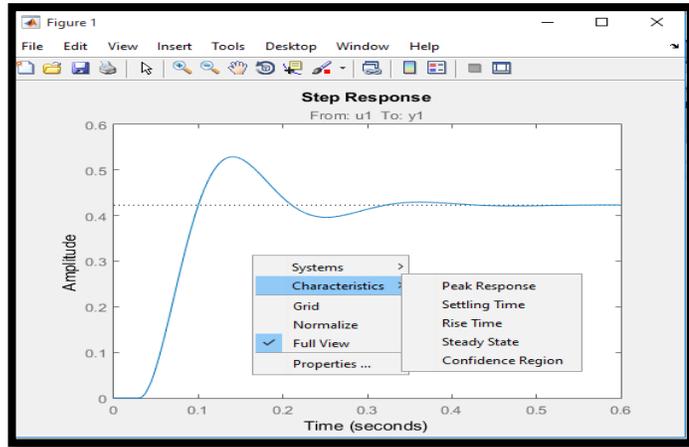


Verifique que la respuesta temporal de la simulación tenga las mismas características del sistema real. Realice el mismo procedimiento pero con una entrada escalón unitario.

PASO 7. CARACTERÍSTICAS DE LA RESPUESTA TEMPORAL DEL MODELO.

Para visualizar las características de la respuesta temporal del sistema digite el "Command Window" la siguiente instrucción, >> step (P2DU), que nos mostrara en una gráfica el comportamiento del modelo ante una entrada escalón unitario, ver figura 11.

Figura 11. Respuesta del modelo ante una entrada escalón unitario.



Por medio de la gráfica determine

Tiempo de retardo (Delay time)= _____

Tiempo de subida (Rise time) = _____

Sobrepaso (Overshoot)= _____

Tiempo de establecimiento (Settling time)= _____

Error en estado estacionario (ess) = _____

Tiempo de muestreo (Sampling time) = _____

PASO 8. SINTONIZACION METODO ZN.

Teniendo en cuenta que la respuesta del sistema tiene oscilaciones sostenidas se emplea el segundo método de Ziegler-Nichols, donde se aplican la regla de sintonización de la tabla 1.

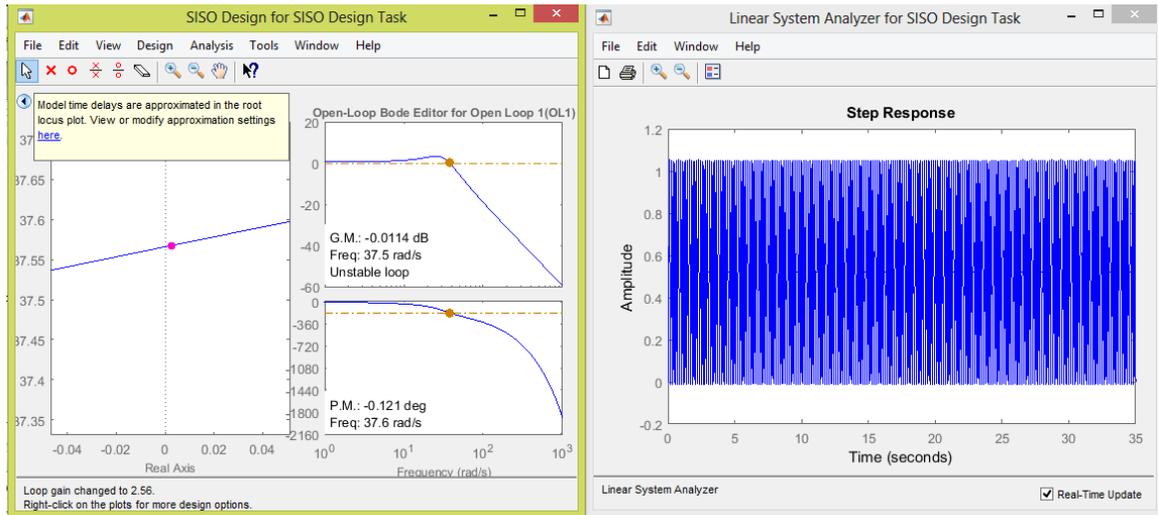
Tabla 1. Reglas de sintonización segundo método ZN.

| Tipo de controlador | K_p | T_i | T_d |
|---------------------|--------------|-----------------------|---------------|
| P | $0.5K_{cr}$ | ∞ | 0 |
| PI | $0.45K_{cr}$ | $\frac{1}{1.2}P_{cr}$ | 0 |
| PID | $0.6K_{cr}$ | $0.5P_{cr}$ | $0.125P_{cr}$ |

Como se puede apreciar en la tabla 1, la regla demanda conocer la ganancia crítica (Kcr) y el periodo crítico (Pcr).

Par hallar la gananciacia critica Kcr y el periodo critico Pcr se utiliza el toolbox SISO de Matlab, para esto digtamos en el Command Window el siguiente instrucción >> sisotool (P2DU). Una vez generada la ventana del SISO se procede a umentar la ganancia del sistema hasta obtener oscilaciones permanentes, ver ejemplo de la figura 12.

Figura 12. Ventana de la herramienta SISO de Matlab.



NOTA: En la GUIA Nª 4, PASO 1, se explica con más detalle la forma de obtener la ganancia crítica (Kcr) y el periodo crítico (Pcr) a través de toolbox SISO.

Luego de obtener las ganancias crítica Kcr y el periodo crítico Pcr se aplican las reglas de sintonización de la tabla 1, donde:

$$K_c = 0.6(K_{cr})$$

$$T_c =$$

$$K_i = 0.5(K_{cr})$$

$$T_i =$$

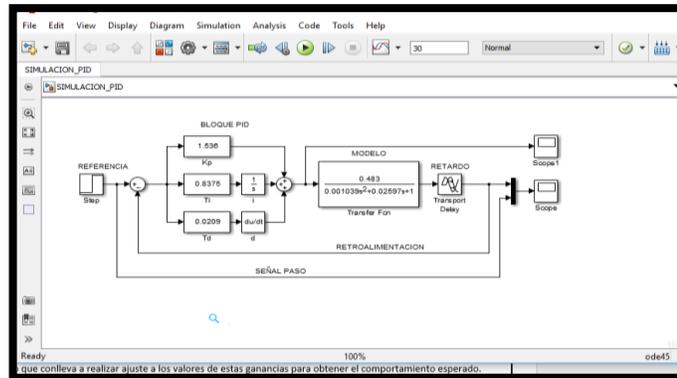
$$K_d = 0.125(K_{cr})$$

$$T_d =$$

PASO 9. SIMULACIÓN DEL LAZO DE CONTROL DEL MODELO.

Una vez obtenida las ganancias del controlador proceda a lanzar la aplicación “CONTROL_PID” ubicado en el escritorio dentro de la carpeta SIMULINK- ARDUINO, ver figura 13.

Figura 13. Diagrama de bloques del lazo de control en Simulink.

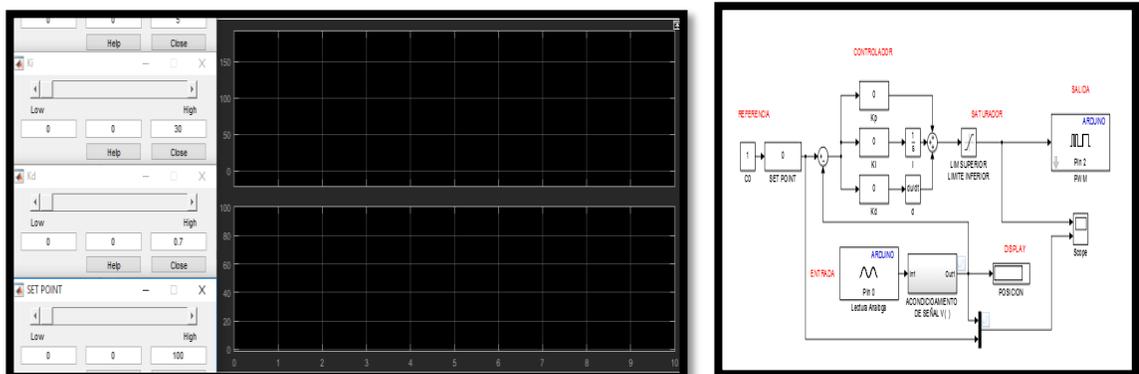


Ubíquese sobre los bloques de ganancias del controlador PID e ingrese los valores obtenidos en el paso anterior y ejecute la simulación. Analice el comportamiento del sistema a través del bloque “Scope”, si la respuesta del sistema no es la esperada deberá realizar ajuste a los valores de estas ganancias para obtener mejores resultados.

PASO 10. INTERFAZ DE MONITOREO Y CONTROL.

Una vez obtenido el comportamiento deseado en la simulación se proceda a lanzar la aplicación “CONTROL_PID_ARDUINO” para el monitoreo y control de la planta BMH, ubicado en el escritorio dentro de la carpeta SIMULINK- ARDUINO, ver figura 14.

Figura 14. Interfaz gráfica de monitoreo y control Simulink.



A través de la interfaz ajuste las ganancias K_p , T_i , T_d con los valores obtenidos en la simulación, asigne el valor inicial de 20 en la referencia (set point) y ponga a correr el programa, esto hace que el balancín se posicione en un ángulo de inclinación de 20 grados, una vez estabilizado en esa posición, cambie la referencia la cual pasara de 20 a 60, analice el comportamiento de la señal del sistema y compárela con el comportamiento de la señal en la simulación, luego determine:

Tiempo de retardo (Delay time)= _____

Tiempo de subida (Rise time) = _____

Sobrepaso (Overshoot)= _____

Tiempo de establecimiento (Settling time)= _____

Error en estado estacionario (ess) = _____

Tiempo de muestreo (Sampling time) = _____

Tabla 2. Respuesta temporal del modelo real y el modelo simulado.

| Ganancias PID | RESPUESTA TEMPORAL | Simulación | Real |
|---------------|--|------------|------|
| | Tiempo de retardo (T_d) | | |
| | Tiempo de subida (T_s) | | |
| | Sobre Impulso (O_s) | | |
| | Tiempo de establecimiento(SS) | | |
| | Error en estado estacionario(e_{ss}) | | |

PASO 11. INFORME

Presente un informe detallado de la práctica con sus conclusiones, del comportamiento del sistema real y el simulado en formato IEEE.