UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TECNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA EN ELECTRÓNICA EN CONTROL Y AUTOMATISMO

TEMA:

ANALISIS DE SISTEMAS DE CONTROL DE PRIMER Y SEGUNDO ORDEN USANDO LENGUAJES DE PROGRAMACIÓN MATLAB-SIMULINK

Previa la obtención del Título

INGENIERO EN ELECTRÓNICA EN CONTROL Y AUTOMATISMO

ELABORADO POR:

CARLOS ANDRES VIVANCO LOAIZA

GUAYAQUIL, ENERO DE 2013

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. Carlos Andres Vivanco Loaiza como requerimiento parcial para la obtención del título de INGENIERO EN ELECTRÓNICA EN CONTROL Y AUTOMATISMO.

Guayaquil, Enero de 2013

Ing. Efrén Herrera Muentes
Director de Tesis

Ing. Manuel Romero Paz
Decano de la Facultad Técnica

Ing. Armando Heras Sánchez
Director de Carrera

Ing. Luis Vallejo Samaniego
Coordinador Académico

Eco. Gladys Contreras Molina
Coordinadora Administrativa

Ing. Judith Gálvez Soto
Revisor de Tesis

Ing. Eduardo Zambrano Robayo
Revisor de Tesis

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

INGENIERÍA EN ELECTRÓNICA EN CONTROL Y AUTOMATISMO

DECLARACIÓN DE RESPONSABILIDAD

CARLOS ANDRES VIVANCO LOAIZA

DECLARO QUE:

El proyecto de grado denominado "Análisis de sistemas de control de primer y segundo orden usando lenguajes de programación Matlab-Simulink" ha sido desarrollada con base a una investigación exhaustiva, respetando derechos intelectuales de terceros conforme las citas que constan en las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.
Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Guayaquil, Enero del 2013

EL AUTOR

VIVANCO LOAIZA CARLOS ANDRES

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

INGENIERÍA EN ELECTRÓNICA EN CONTROL Y AUTOMATISMO

AUTORIZACIÓN

Yo, VIVANCO LOAIZA CARLOS ANDRES

Autorizo a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del proyecto titulado: “Análisis de sistemas de control de primer y segundo orden usando lenguajes de programación Matlab-Simulink”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Guayaquil, Enero del 2013

EL AUTOR

VIVANCO LOAIZA CARLOS ANDRES

Agradecimientos

Dedico este proyecto de Tesis a Dios, por ser fuente suprema de toda sabiduría y ser la luz y guía de nuestros propósitos así como la fuerza que inspira nuestro camino.

A mi familia, quienes me han brindado su apoyo incondicional en mi vida, impulsándome siempre a seguir a delante y luchar por mis metas, además por ser el pilar fundamental para alcanzar tan anhelado triunfo que representa el final de una de las etapas más importantes de mi vida y el inicio de otras que serán aún más enriquecedores.

Al cuerpo docente de la Universidad Católica, al Ing. Efrén Herrera, Director de tesis, por el apoyo y orientación brindada para la culminación del presente proyecto.

INDICE GENERAL

INDICE DE FIGURAS

INDICE DE TABLAS

INTRODUCCIÓN

El control automático en la actualidad desempeña una función importante y vital en el adelanto de la ingeniería y la ciencia, su importancia en desarrollo de los sistemas de automóviles espaciales, como guía de misiles tanto de aviones y barcos y de robótica, este se ha convertido en una parte transcendental en los procesos modernos de la industria y la elaboración o fabricación, como por ejemplo el control automático es esencial en el control o modelo matemático o numérico de máquinas industriales de manufactura en el diseño de sistemas aeronáuticos no piloteados, en los sistemas aeroespacial y en el diseño de automóviles en la industria automotriz, adicionalmente también es importante en operaciones de la industria y su control en temperatura, presión, viscosidad, humedad, vibración, flujo, iluminación etc.

El objetivo de esta tesis es profundizar y fortalecer el estudio, la comprensión y aplicación del comportamiento óptimo de los controles automáticos dinámicos que en la actualidad el Ingeniero en Electrónica control y Automatismo requiere conocer para ofrecer sus servicios profesionales en la industria con el propósito de mejorar la productividad, simplificar el trabajo de muchas operaciones manuales repetitivas y rutinarias, así como de otras actividades, empezando con el conocimiento de un modelado matemático, haciendo lasos de control y poder simularlos y graficarlos a través de una herramienta de programación o sistema como el Matlab y el Simulink.

Para facilitar el estudio de los controles automáticos, se utilizó el laboratorio de Electrónica control y Automatismo de la Facultad de Educación Técnica para el Desarrollo de la Universidad Católica Santiago de Guayaquil, el software Matlab y Simulink y se realizaron algunas prácticas experimentales, para comprobar los diferentes sucesos en los circuitos aplicados.

Las prácticas serán plasmadas como un aporte al estudio de los controles automáticos el cual ayudará a la compresión más simple tanto para alumnos, profesores y profesionales, realizando mediciones, cálculos y observaciones que pueden reforzar los conocimientos de estos procesos.

CAPÍTULO 1 – EL PROBLEMA

1.1 Planteamiento del Problema

Actualmente la materia de control es impartida de forma teórica, por cuya razón existe una eminente necesidad por parte del estudiante de aprender interactuando la parte teórica dictada en clases con la parte práctica simulada en el mismo momento que se está explicando o dictando la teoría, reforzar con demostraciones prácticas de laboratorio el funcionamiento de los sistemas de control de primer y segundo orden usando herramientas de cálculo y de simulación como el Matlab y el Simulink, fundamentando los conocimientos con experimentos, le dará un plus adicional al aprendizaje y proceso cognitivo de los profesionales en esta carrera.

1.1.1 Delimitación del Problema

De acuerdo a lo indicado anteriormente esta tesis es delimitada de la siguiente manera:
Analizar los sistemas de control de primer y segundo orden usando herramientas de programación como el Matlab-Simulink, para demostrar a través de prácticas simuladas los fundamentos básicos, el aprendizaje y funcionamiento de los sistemas de control.

1.1.2 Causas y Consecuencias

La subutilización en el laboratorio de Electrónica de herramientas digitales de programación que ayudan al cálculo, simulación y graficación de modelos y controladores avanzados.

Poco entendimiento del estudiante de la parte teórica con la práctica al no poder relacionar y comparar los sistemas de primer orden y segundo orden en forma interactiva.

El estudiante no conoce el funcionamiento práctico de las herramientas digitales de programación y simulación como son el Matlab y el Simulink.

Bajo nivel de aprendizaje práctico del estudiante al recibir estudios teóricos de una asignatura, en este caso podría ser el de la Materia de Teoría de control 1.

1.1.3 Justificación

Se considera importante el trabajo de esta tesis debido a que se propone un método de estudio teórico y práctico acerca de los sistemas de control de primer y segundo orden usando herramientas de cálculo y de simulación como el Matlab y el Simulink, temas que son tratados en la materia de Teoría de Control uno de la carrera de Ingeniería en Electrónica en Control y Automatismo impartida en el sexto semestre de la Facultad de Educación Técnica para el Desarrollo de la Universidad Católica de Santiago de Guayaquil, si bien es cierto esta materia es netamente teórica es de suma ayuda el que el estudiante tenga una guía que le facilite el entendimiento de los supuestos y cálculos de los fenómenos estudiados en la teoría, por cuya razón esta tesis está dirigida a reforzar el estudio escrito impartido por el profesor a uno que experimenta y ejercita en la práctica de laboratorio.

1.2 OBJETIVOS

Los objetivos planteados para este proyecto de investigación son los siguientes:

1.2.1 Objetivo General

Fortalecer la metodología de estudio, la comprensión y aplicación del comportamiento de los sistemas de control automáticos de primer y segundo orden usando lenguajes de programación de cálculo y de simulación como el *Matlab* y el *Simulink.*

1.2.2 Objetivos Específicos

- Aprender los diferentes modelos matemáticos de sistemas dinámicos.
- Aprender a representar en diagramas de bloques los diferentes sistemas de control.
- Aprender los diferentes diseños de sistemas de control.
- Estudiar el manejo y las funciones básicas del Matlab y aplicar los modelos matemáticos de los sistemas dinámicos.
- Estudiar el manejo y las funciones básicas del Simulink y aplicar los diagramas y diseños de los sistemas de control.
- Determinar las prácticas que serán realizadas.
- Analizar los resultados de las prácticas desarrolladas.

1.3 HIPOTESIS

Reunir la información y comparar la teoría con la práctica, en este caso realizar algunos ejercicios de laboratorio y comprobar si efectivamente se cumple el fenómeno de los diferentes circuitos simulados realizados con los supuestos expresados en conceptos y definiciones, una vez plasmado las comparaciones y hallazgos esta tesis debería ser utilizada como un manual y guía de ayuda, apoyo y fortalecimiento del conocimiento y aprendizaje no solo para los estudiantes sino que también para profesores y profesionales en la carrera.

CAPÍTULO 2 MARCO TEORICO

EXPOSICION FUNDAMENTADA

2. ANTECEDENTES

En la década de los cincuenta se estudiaron algunos sistemas de control y automatización, estos sistemas fueron aceptables en su momento pero no eran óptimos, así que a finales de esta década se realizó mucho énfasis en un sistema de automatización que funcionaba pero que era susceptible de mejorarlo con respecto de algún criterio.

A medida que las industrias se iban modernizando se volvían más complejas en el control y automatización de sus plantas, ya no solo manejaban una entrada y una salida si no que ya comenzaron a ser producción múltiple o sea que comenzaron a manejar varias entradas y varias salidas en sus cadenas de producción el cual se requería de análisis más profundos y la implementación de modelos matemáticos más complejos en donde se requería varias ecuaciones.

Luego en los años sesenta con la llegada y la disponibilidad de las computadoras digitales el análisis en el dominio del tiempo de sistemas complejos, la teoría del control moderno basada en el análisis en el dominio del tiempo y la síntesis a partir de variables de estado, se han desarrollado para manejar la complejidad de las plantas modernas y los requisitos cada vez más exigentes sobre precisión, peso y coste en aplicaciones tanto como militares, espaciales e industriales.

Desde los años ochenta hasta la actualidad los sistemas de control se han centrado en sistemas de control robustos y temas relacionados, ahora en la actualidad que los computadoras digitales son más baratas y más compactas se usa como parte integral de los sistemas de control, sin dejar a un lado toda la gama de programas y sistemas que no solo están creados para el ámbito de la ingeniería sino que también para los sistemas biológicos, biomédicos, económicos y socioeconómicos.

2.1. CONCEPTO BÁSICO

El autor tomo la explicación textual del tema de: Ogata, K. (2003).
Un sistema de control está definido como un conjunto de componentes que pueden regular su propia conducta o la de otro sistema con el fin de lograr un funcionamiento predeterminado.

De esta forma podríamos considerar en un sentido lo más amplio posible un sistema de control como aquel sistema que ante unos objetivos determinados responde con una serie de actuaciones.

2.2. Componentes básicos de un sistema de control.

El autor tomo la explicación textual del tema de: Ogata, K. (2003).
Los componentes básicos de un sistema de control se puede describir mediante:

- Objetivos de control
- Componentes del sistema de control
- Resultados o salidas

La relación básica entre estos tres componentes se ilustra en la figura2.1. En términos más técnicos, los objetivos se pueden identificar como entradas, o señales actuantes u , y los resultados también se llaman salidas, o variables controladas, y en general el objetivo de un sistema de control es controlar las salidas en alguna forma prescrita mediante las entradas a través de los elementos del sistema de control.

Figura2.1: Componentes básicos de un Sistema de Control

Ogata, K. (2003). *Ingeniería de Control Moderna.* Madrid: Pearson Educación.

2.3. Elementos en un sistema de control.

El autor tomo la explicación textual del tema de: R.P. Ñeco, O. Reinoso, N.García, R. Arcil. (2003).

Los sistemas de control se clasifican en los siguientes elementos:

2.3.1. Variable a controlar.

Generalmente se le conoce como señal de salida. Constituye la señal que deseamos que adquiera unos valores determinados.

2.3.2. Planta o Sistema.

La planta o sistema constituye el conjunto de elementos que realizan una determinada función.

2.3.3. Sensor.

El sensor es el elemento que permite captar el valor de la variable a controlar en determinados instantes de tiempo.

2.3.4. Señal de referencia.

Es la señal consigna o valor que deseamos que adquiera la señal de salida (objetivo de control).

2.3.5. Actuador.

El actuador es el elemento que actúa sobre el sistema modificando de esta forma la señal de salida.

2.3.6. Controlador.

El controlador o regulador es el elemento que comanda al actuador en función del objetivo de control.

Todos estos elementos aparecen de alguna u otra forma en casi todo sistema de control. Identificar y estudiar cada uno de ellos de una forma correcta resulta esencial para poder diseñar un controlador que permita alcanzar el objetivo de control deseado en todo instante.

2.4. Sistemas de control en lazo abierto y lazo cerrado

Los sistemas de control son aquellos sistemas que tienden a mantener una relación pre establecida entre la variable de salida (variable controlada) y la referencia. En función del efecto de la retroalimentación pueden clasificarse en sistemas en lazo abierto y sistemas en lazo cerrado.

Cuando se desea mantener un objetivo de control determinado en un sistema dos son los esquemas de control que se pueden considerar: sistemas de control en lazo abierto y sistemas de control en lazo cerrado.

2.4.1. Sistema de control en lazo abierto

El autor tomo la explicación textual del tema de: R.P. Ñeco, O. Reinoso, N.García, R. Arcil. (2003).

Es aquel en el que la señal de salida no influye sobre la acción de control. De esta forma el controlador o regulador no tiene en cuenta el valor de la señal de salida, ni se compara esta con la señal de referencia para decidir la actuación en todo instante sobre el sistema. El caso más típico de un sistema de control en lazo abierto lo constituye la lavadora eléctrica donde el sistema de control va modificando el tiempo, la temperatura de lavado, etc. en función de la indicación del usuario y no en función del nivel de lavado de la ropa (que constituiría el objetivo de control). De esta forma el usuario decide el programa que desea realizar (señal d referencia), y el controlador actúa sobre los diferentes mecanismos del sistema (lavadora) de forma que realiza una serie de actuaciones sin tener en cuenta la señal de salida. En la figura2.2 se pueden observar las señales involucradas en un control en lazo abierto.

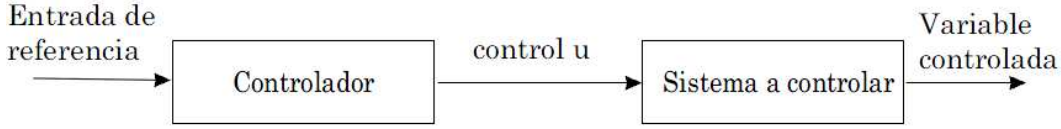


Figura # 2.2: Sistema de Control de lazo abierto
Fuente: García, R. P. (2003). Apuntes de Sistemas de Control. Elche: Editorial Club Universitario.

2.4.2. Sistema de control en lazo cerrado

En estos sistemas, la salida se mide y se retroalimenta, de manera que la salida tiene efecto sobre la acción de control. La diferencia entre la referencia y la variable de salida o realimentación se conoce como señal de error actuante y se utiliza en el controlador para llevar la variable controlada al valor deseado. Estos sistemas son sistemas de control realimentados en los que la acción de realimentación se utiliza para reducir el error del sistema. Tal como se muestra en la figura 2.3.

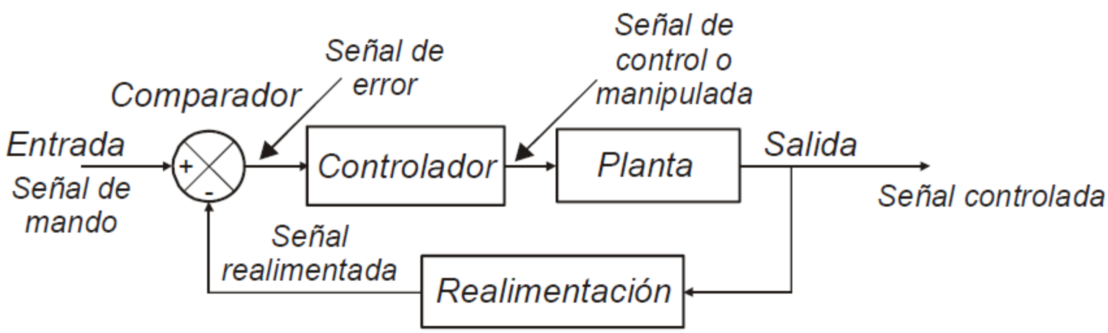


Figura # 2.3: Diagrama de Bloques de un sistema de control de lazos cerrado
Fuente: García, R. P. (2003). Apuntes de Sistemas de Control. Elche: Editorial Club Universitario.

2.5. Realimentación de sistemas

El autor tomo la explicación textual del tema de: R.P. Ñeco, O. Reinoso, N.García, R. Arcil. (2003).

En el apartado previo se ha establecido la diferencia entre los sistemas de control en lazo abierto y los sistemas de control en lazo cerrado. Como quedó establecido, en estos últimos se compara la señal de salida (o variable que se desea controlar) obtenida en la mayor parte de las ocasiones a través de un conjunto de sensores, con una señal de referencia. Este efecto se conoce como realimentación. El efecto inmediato que persigue esta realimentación es reducir el error entre la señal de salida y la señal de referencia actuando en consecuencia.

Pero no sólo la realimentación tiene por cometido reducir el error entre la señal de salida y la señal de referencia de un sistema. La realimentación también produce efectos sobre la ganancia global del sistema (puede tanto aumentar como disminuir en función de la realimentación), la estabilidad (un sistema inicialmente estable puede pasar a ser inestable o a la inversa en función de la realimentación), así como sobre las perturbaciones posibles que se presenten sobre el mismo (puede reducir el efecto de las perturbaciones que se originan sobre el sistema). Por lo tanto, la realimentación es un elemento clave muy a tener en cuenta en el estudio de los sistemas de control ya que puede modificar considerablemente los resultados producidos por estos.

2.6. Tipos de sistemas de control realimentados

Los sistemas de control realimentados se pueden clasificar de diversas formas, dependiendo del propósito de clasificación, por ejemplo, de acuerdo con el método de análisis y diseño, los sistemas de control se clasifican en **Lineales y no Lineales**, **Variantes con el tiempo e Invariantes con el tiempo**, de acuerdo con los tipos de señales usados en el sistema, se hace referencia a **sistemas en tiempo continuo y tiempo discreto, o sistemas modulados o no modulados,** a menudo los sistemas de control se clasifican de acuerdo con su propósito principal, por ejemplo un sistema de control de posición y un sistema de control de velocidad controlan las variables de salida de acuerdo con la forma como su nombre lo indica.

El tipo de un sistema de control se define de acuerdo con la forma de la función de transferencia en lazo abierto. En general, existen muchas formas de identificar un sistema de control de acuerdo con alguna función especial del sistema. Es importante que algunas de estas formas comunes de clasificar a los sistemas de control sean conocidas para obtener una perspectiva propia antes de embarcarse en su análisis y diseño.

2.6.1 Sistemas de control lineales vs no lineales.

Esta clasificación está hecha de acuerdo con los métodos de análisis y diseño estrictamente hablando, los sistemas lineales no existen en la práctica, ya que todos los sistemas físicos son no lineales en algún grado. Los sistemas de control realimentados son modelos ideales fabricados por el analista para simplificar el análisis y seño. Cuando las magnitudes de las señales en un sistema de control están limitadas en intervalos en los cuales los componentes del sistema exhiben una característica lineal (se aplica el principio de superposición), el sistema es esencialmente lineal. Pero cuando las magnitudes de las señales se extienden más allá del intervalo de porción lineal, dependiendo de la severidad de la no linealidad, el sistema no se debe seguir considerando lineal. Por ejemplo los amplificadores usados en los sistemas de control a

menudo exhiben un efecto de saturación cuando la señal de entrada es muy grande; el campo magnético de un motor normalmente tiene propiedades de saturación.
Muy a menudo las características no lineales son introducidas en forma intencional en un sistema de control para mejorar su desempeño o proveer un control más efectivo. Por ejemplo para alcanzar un control de tiempo mínimo, un tipo de controlador prendido-apagado (relevador) se emplea en muchos misiles o sistemas de control de naves espaciales. Típicamente en estos sistemas, los motores de reacción están a los lados del vehículo para producir un par de reacción para control de altitud, estos motores son controlados en una forma o totalmente prendidos o apagados, por lo que una cantidad fija de aire es aplicada desde un motor de reacción dado durante cierto tiempo para controlar la altitud del vehículo espacial.

Para sistemas lineales, existen una gran cantidad de técnicas analíticas y gráficas para fines de diseñó y análisis, los sistemas no lineales son difíciles de tratar en forma matemática, y no existen métodos generales disponibles para resolver una gran variedad de clases de sistemas no lineales. En el diseño de sistemas de control, es práctico, primero diseñar el controlador en base a un modelo de un sistema lineal despreciando las no linealidades del sistema. Entonces el controlador diseñado se aplica al modelo del sistema no lineal para su evaluación o rediseño mediante simulación en computadora

2.6.2 Sistemas invariantes con el tiempo vs variantes con el tiempo

Cuando los parámetros del sistema de control son estacionarios con respecto al tiempo durante la operación del sistema, el sistema se denomina **sistema invariante con el tiempo.** En la práctica la mayoría de los sistemas físicos contienen elementos que derivan o varían con el tiempo. Por ejemplo la resistencia de la bobina de un motor eléctrico viajará cuando es excitado por primera vez y su temperatura está aumentando.

Un sistema en tiempo continuo es aquel en las que las señales en varias partes del sistema son todas funciones de la variable continua tiempo *t*. Entre todos los sistemas de control en tiempo continuo, las señales se pueden clasificar posteriormente como de va a cd, a diferencia de la definición general de señales de va a cd utilizadas en ingeniería eléctrica, los sistemas de control va y cd tienen un significado especial en la terminología de sistemas de control.

Cuando se hace referencia a un sistema de control va, usualmente significa que las señales en el sistema están moduladas según algún esquema de modulación. Por otro lado, cuando se hace referencia a un sistema de control cd, no significa que todas las señales en el sistema sean unidireccionales; entonces no habría movimientos de control correctivo. Un control cd simplemente implica que las señales no son moduladas, pero aún son señales de va de acuerdo a la definición anterior. El esquema de un sistema de control de lazo cerrado se representa en la fig. 2.4. Las formas de ondas típicas de las señales en respuesta a una función escalón de entrada se muestran en la figura. Los componentes típicos de un sistema de control de cd son potenciómetros, amplificadores de cd, tacómetros de cd, etcétera.

El esquema de un sistema de control va que desempeña la misma tarea que el de la fig. 2.4 se representa en la fig. 2.5. En este caso las señales en el mismo están moduladas; esto es, la información se transmite mediante una señal portadora de va. Observe que la variable controlada- salida permanece aún similar a la del sistema cd. En este caso, las señales moduladas son desmoduladas por la característica de paso bajo del motor de va. Los sistemas de control de va se utilizan en una forma extensa en aeronaves y sistemas de control de misiles, en los que el ruido y las perturbaciones a menudo crean problemas. Al utilizar sistemas de control de va modulados con una portadora de 400mhz o mayor, el sistema sea menos susceptible a ruido de baja frecuencia. Los componentes típicos de un sistema de control de va son: sinceros, amplificadores de va, motores de va, giroscopios, acelerómetros, etcétera

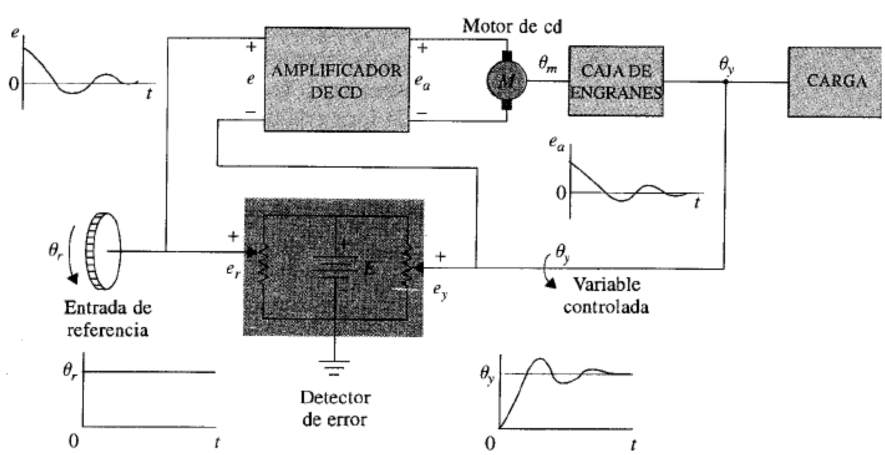


Figura # 2.4: Diagrama de un sistema de control típico de cd en lazo cerrado
Kuo, B. C. (1996). *Sistemas de Control Automatico*. Mexico: Pearson Educación.

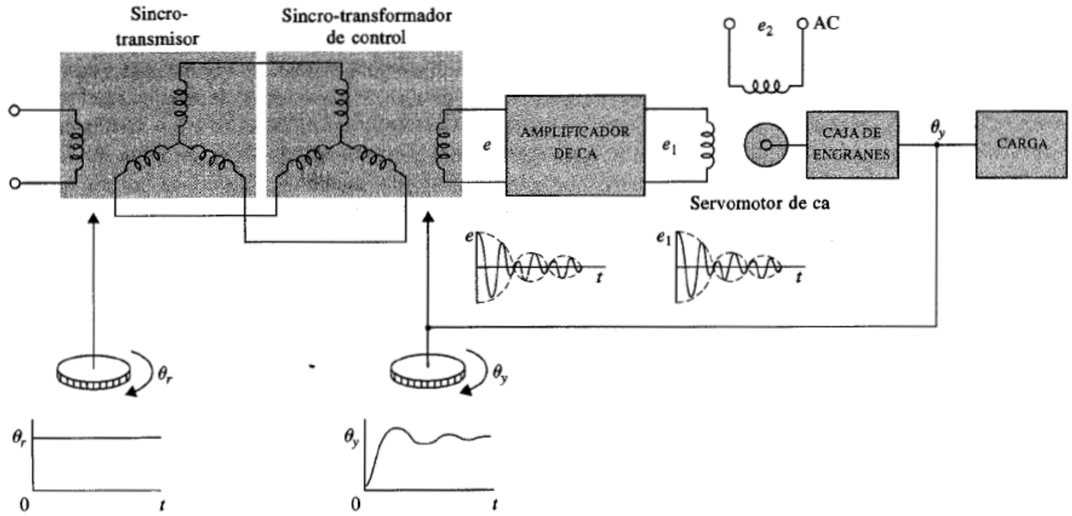


Figura # 2.5: Diagrama de un sistema de control típico de va típico en lazo cerrado
Kuo, B. C. (2001). *Sistemas de Control Automatico*. Mexico: Pearson Educación.

2.6.3. Sistemas de control   en tiempo discreto.

Los sistemas de control en tiempo discreto difieren de los sistemas de control en tiempo continuo en que las señales en uno o más puntos del sistema son, ya sea en la forma de pulsos o un código digital. Normalmente los sistemas de tiempo discreto se subdividen en sistemas de control de datos muestreados y sistemas de control digital. Los sistemas de control muestreados se refiere a una clase más general de sistemas en tiempo discretos en los que las señales están en la forma de pulsos de datos. Un sistema de control digital se refiere al uso de una computadora o controlador digital en el sistema, de tal forma que las señales están códigos digitales, tal como un código binario.

En general un sistema de control muestreado recibe datos o información sólo en forma intermitente en instantes específicos. Por ejemplo la señal de error en un sistema de control se puede proporcionar en la forma de pulsos, en cuyo caso el sistema de control no recibe información acerca del error durante los periodos entre dos pulsos consecutivos. Estrictamente un sistema de datos muestreados también se puede clasificar como un sistema de va, ya que la señal de sistema esta modulada por pulsos.

2.7. Acciones básicas de control

La forma en que el control automático produce la señal de control se denomina acción de control. En los controles automáticos industriales son muy comunes los siguientes tipos de acciones de control:

- Acción de dos posiciones (ON-OFF).
- Acción proporcional (P).
- Acción integral (I).
- Acción proporcional-integral (PI).
- Acción proporcional-integral-derivativo (PID).

2.7.1. Acción de dos posiciones

El autor tomo la explicación textual del tema de**:** Gómez Sarda, J. R. (2008).

Este tipo de acción también se conoce con el nombre de ON-OFF o de todo o nada. Es un control relativamente simple y por eso muy empleado tanto en aplicaciones industriales como domésticas. En los sistemas de control con acción de dos posiciones, el elemento actuante solo tiene dos posiciones fijas, que casi siempre son: conectado y desconectado. La señal de control *u(t)* en este tipo de control permanece en un valor máximo o mínimo en dependencia del error actuante *e(t),* o sea:

$u(t) = M_1$ para $e(t) > 0$

$u(t) = M_2$ para $e(t) < 0$

Los controles de dos posiciones son normalmente dispositivos eléctricos, por lo general una válvula con un solenoide como actuador. En la Figura 2.6 se muestra el diagrama de bloques de un control de dos posiciones.

Como la conmutación se realiza en el = 0, un pequeño cambio en cualquier sentido lo haría conmutar, por lo que esta es muy frecuente. Esto haría que se deteriorara con rapidez el elemento de acción final o actuador. Para reducir este efecto, se provee al control deliberadamente de una brecha diferencial (Figura 2.7). Esta brecha diferencial hace que la salida del control u(t) mantenga su valor hasta que la señal de error actuante haya pasado levemente del valor de cero.

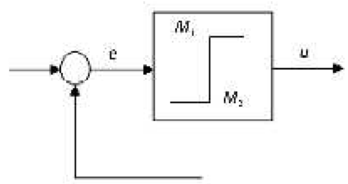


Figura # 2.6: Diagrama de bloque de un control de dos posiciones
Fuente: Gómez Sarda, J. R. (2008). Temas especiales de instrumentación y control. Cuba: Editorial Félix Varela.

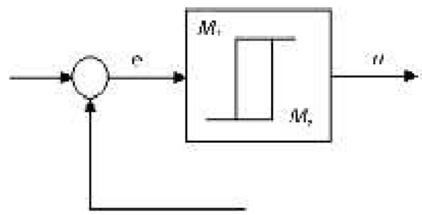


Figura # 2.7: Diagrama de bloque de un control de dos posiciones con brecha diferencial
Fuente: Gómez Sarduy, J. R. (2008). Temas especiales de instrumentación y control. Cuba: Editorial Félix Varela.

Un ejemplo de este tipo de control lo constituye el sistema de control de nivel de líquido mostrado en la Figura 2.8 Con este control, la válvula está abierta o cerrada y el flujo de entrada es una constante positiva o cero. La señal de salida se mueve continuamente entre los dos límites requeridos mostrando una oscilación de la salida entre dos valores, como se observa en la Figura 2.8. Esta respuesta es típica de un control de dos posiciones. Otros ejemplos de control de dos posiciones son los equipos controlados por termostatos (refrigerador, horno eléctrico, aire acondicionado, plancha, etc.).

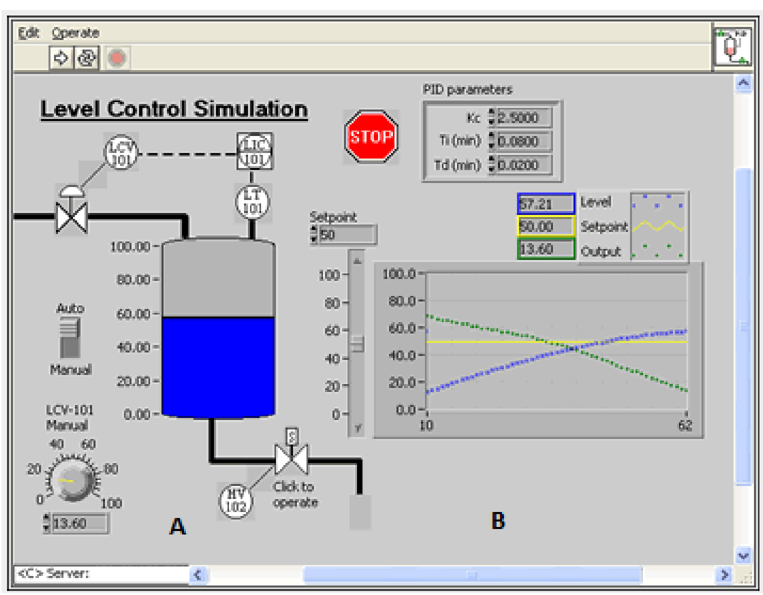


Figura # 2.8: Sistema de control de nivel de líquido.
Fuente: http://www.xarosemena.com/data/control/tank_level.htm

2.7.2. Acción proporcional (P)

El autor tomo la explicación textual del tema de**:** Gómez Sarda, J. R. (2008).

Como el propio nombre lo indica, la relación que une la variable manipulada m con el error e, es una relación lineal, expresada por la siguiente ley matemática: u(t) = Kp . e(t) + Uo Donde:

Kp: ganancia proporcional.

Uo : valor de salida del controlador cuando e(t) = 0 y se suele seleccionar en el medio de la gama de salida del controlador . En los controladores comerciales se recomienda situar la condición inicial de la salida de la manera siguiente:

Señales eléctricas

4-20 mA   Uo= 12ma

1-5 V Uo = 3 V

Señales neumáticas

3-15 psi   Uo=9psi

0,2-1 kgf /cm 2 Uo=0,6 kgf/cm

20-100 kPa Uo = 60 kPa

La ganancia proporcional es igual a la variación de la señal de control u cuando existe una variación unitaria del error e:

$$KP \frac{\Delta u(t)}{\Delta e(t)}$$

En magnitudes transformadas de Laplace:

$$KP \frac{U(s)}{E(s)}$$

El diagrama de bloques de la acción proporcional se muestra en la Figura2.9. El control proporcional es esencialmente un amplificador de ganancia ajustable. Es muy común que en los controladores comerciales este parámetro no se dé como ganancia, sino como el inverso de la ganancia, conocido como banda proporcional.

% Banda proporcional = $\dfrac{100}{KP}$

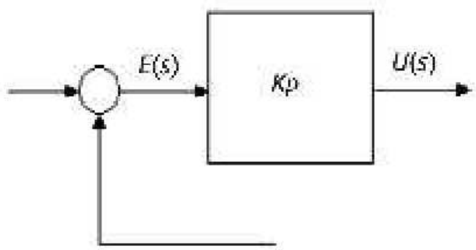


Figura # 2.9: Diagrama en bloque de un control proporcional
Fuente: Gómez Sarduy, J. R. (2008). Temas especiales de instrumentación y control. Cuba: Editorial Félix Varela.

2.7.3. Acción proporcional Integral

El modo de control Integral tiene como propósito disminuir y eliminar el error en estado estacionario, provocado por el modo proporcional. El control integral actúa cuando hay una desviación entre la variable y el punto de consigna, integrando esta desviación en el tiempo y sumándola a la acción proporcional. El *error* es integrado, lo cual tiene la función de promediarlo o sumarlo por un período determinado; Luego es multiplicado por una constante **I**. Posteriormente, la respuesta integral es adicionada al modo Proporcional para formar el control P + I con el propósito de obtener una respuesta estable del sistema sin error estacionario.
El modo integral presenta un desfasamiento en la respuesta de 90º que sumados a los 180º de la retroalimentación (negativa) acercan al proceso a tener un retraso de 270º, luego entonces solo será necesario que el tiempo muerto contribuya con 90º de retardo para provocar la oscilación del proceso. La ganancia total del lazo de control debe ser menor a 1, y así inducir una atenuación en la salida del controlador para conducir el proceso a estabilidad del mismo. Se caracteriza por el tiempo de acción integral en minutos por repetición. Es el tiempo en que delante una señal en escalón, el elemento final de control repite el mismo movimiento correspondiente a la acción proporcional.

El control integral se utiliza para obviar el inconveniente del offset (desviación permanente de la variable con respecto al punto de consigna) de la banda proporcional.
La fórmula del integral está dada por:

$$I_{\text{sal}} = K_i \int_0^t e(\tau)\, d\tau$$

2.7.4. Acción Proporcional Derivativa

La acción derivativa se manifiesta cuando hay un cambio en el valor absoluto del error; (si el error es constante, solamente actúan los modos proporcional e integral).
El *error* es la desviación existente entre el punto de medida y el valor consigna, o "*Set Point*".
La función de la acción derivativa es mantener el error al mínimo corrigiéndolo proporcionalmente con la misma velocidad que se produce; de esta manera evita que el error se incremente.

Se deriva con respecto al tiempo y se multiplica por una constante **D** y luego se suma a las señales anteriores (P+I). Es importante adaptar la respuesta de control a los cambios en el sistema ya que una mayor derivativa corresponde a un cambio más rápido y el controlador puede responder acordemente.
La fórmula del derivativo está dada por:

$$D_{\text{sal}} = K_d \frac{de}{dt}$$

El control derivativo se caracteriza por el tiempo de acción derivada en minutos de anticipo. La acción derivada es adecuada cuando hay retraso entre el movimiento de la válvula de control y su repercusión a la variable controlada.

Cuando el tiempo de acción derivada es grande, hay inestabilidad en el proceso. Cuando el tiempo de acción derivada es pequeño la variable oscila demasiado con relación al punto de consigna. Suele ser poco utilizada debido a la sensibilidad al ruido que manifiesta y a las complicaciones que ello conlleva.

El tiempo óptimo de acción derivativa es el que retorna la variable al punto de consigna con las mínimas oscilaciones

Ejemplo: Corrige la posición de la válvula (elemento final de control) proporcionalmente a la velocidad de cambio de la variable controlada.

La acción derivada puede ayudar a disminuir el rebasamiento de la variable durante el arranque del proceso. Puede emplearse en sistemas con tiempo de retardo considerables, porque permite una repercusión rápida de la variable después de presentarse una perturbación en el proceso.

2.7.5. Acción proporcional-integral-derivativo (PID)

Un PID (Proporcional Integral Derivativo) es un mecanismo de control por realimentación que calcula la desviación o error entre un valor medido y el valor que se quiere obtener, para aplicar una acción correctora que ajuste el proceso. El algoritmo de cálculo del control PID se da en tres parámetros distintos: el proporcional, el integral, y el derivativo.

El valor Proporcional determina la reacción del error actual.
El Integral genera una corrección proporcional a la integral del error, esto nos asegura que aplicando un esfuerzo de control suficiente, el error de seguimiento se reduce a cero.

El Derivativo determina la reacción del tiempo en el que el error se produce.

La suma de estas tres acciones es usada para ajustar al proceso vía un elemento de control como la posición de una válvula de control o la energía suministrada a un calentador, por ejemplo. Ajustando estas tres variables en el algoritmo de control del PID, el controlador puede proveer un control diseñado para lo que requiera el proceso a realizar. La respuesta del controlador puede ser descrita en términos de respuesta del control ante un error, el grado el cual el controlador llega al "set point", y el grado de oscilación del sistema.

El uso del PID para control no garantiza control óptimo del sistema o la estabilidad del mismo. Algunas aplicaciones pueden solo requerir de uno o dos modos de los que provee este sistema de control. Un controlador PID puede ser llamado también PI, PD, P o I en la ausencia de las acciones de control respectivas. Los controladores PI son particularmente comunes, ya que la acción derivativa es muy sensible al ruido, y la ausencia del proceso integral puede evitar que se alcance al valor deseado debido a la acción de control , tal como se muestra en la figura 2.10

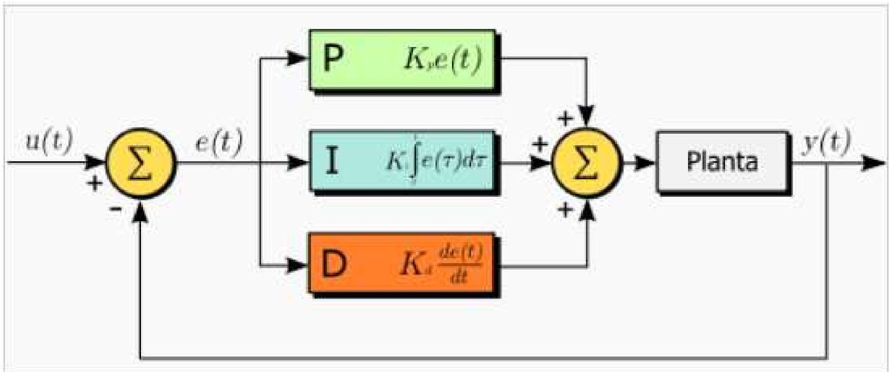


Figura # 2.10 : Diagrama en bloque de un control PID
Fuente: Por Arturo Urquizo (Trabajo propio) [CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0) undefined GFDL (http://www.gnu.org/copyleft/fdl.html)], undefined

2.8. Transformada de Laplace

Control de Procesos

El campo de aplicación de los sistemas de control es muy amplio. Y una herramienta que se utiliza en el diseño de control clásico es precisamente: La transformada de Laplace

En el estudio de los procesos es necesario considerar modelos dinámicos, es decir, modelos de comportamiento variable respecto al tiempo. Esto trae como consecuencia el uso de ecuaciones diferenciales respecto al tiempo para representar matemáticamente el comportamiento de un proceso.

El comportamiento dinámico de los procesos en la naturaleza puede representarse de manera aproximada por el siguiente modelo general de comportamiento dinámico lineal:

$$a_n \frac{d^n y(t)}{dt^n} + a_n - 1 \frac{d^{n-1} y(t)}{dt^{n-1}} + a_{n-2} \frac{d^{n-2} y(t)}{dt^{n-2}} + \ldots\ldots + a_0 y(t) = x(t)$$

La transformada de Laplace es un método operacional que puede usarse para resolver ecuaciones diferenciales lineales. Este método presenta las siguientes ventajas frente al método clásico para la resolución de ecuaciones diferenciales:

- La transformada de Laplace permite convertir una ecuación integro-diferencial en una ecuación algebraica. Esta ecuación algebraica se manipula para obtener la solución en el dominio s.
- La respuesta transitoria y en estado estable, se obtienen al obtener la solución final, que se encuentra tomando la transformada inversa de Laplace (esta es obtenida por tablas o utilizando el método de expansión por fracciones parciales).
- Permite utilizar técnicas gráficas para predecir el funcionamiento del sistema sin obtener la solución del mismo.

Se puede decir que es la segunda transformación más utilizada para resolver problemas físicos, después de la transformación de Fourier. La transformada de Laplace unilateral se define como:

$$F(s) = \mathcal{L}\left\{f(t)\right\} = \int_{0}^{\infty} f(t)e^{-st}dt$$

Donde:

$f(t)$	Es una función en el tiempo
$F(s)$	Es la transformada de Laplace de $f(t)$
s	Es una variable compleja
$\mathcal{L}$	Es el operador lineal de Laplace

Aplicando la transformada de Laplace a una ecuación diferencial, se tiene una ecuación algebraica cuya solución se obtiene a partir de operaciones básicas del álgebra. Esta solución está en función de s y para transformarla a una función en el tiempo se necesita de La Transformada inversa de Laplace.

2.9. La transformada inversa de Laplace

La transformada inversa de Laplace formalmente se define por la siguiente integral de inversión:

$$f(t) = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} F(s) e^{st} ds$$

Donde c es una constante mayor que cualquier punto singular de $F(s)$

Esta integral de inversión rara vez se usa, ya que existen otros métodos más directos y simples. Como por ejemplo tablas de transformadas o fracciones parciales. Tal como se muestra en la figura 2.11

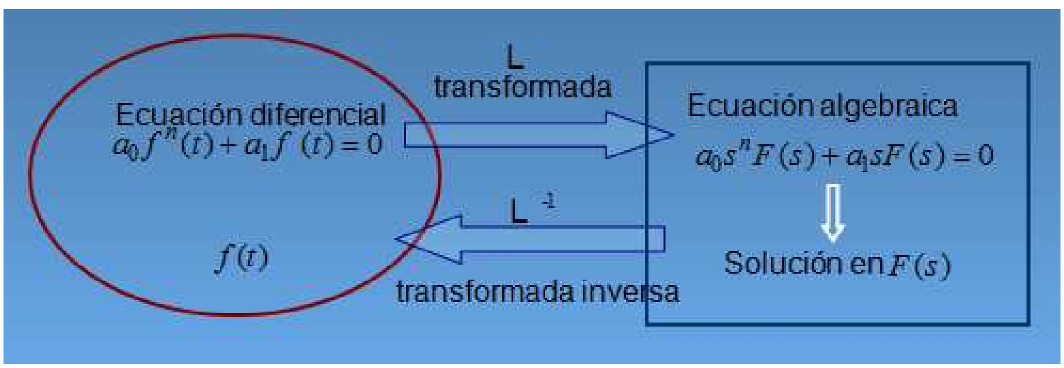


Figura # 2.11: Transformada de Laplace
Fuente: ***Departamento de Control, División de Ingeniería Eléctrica Facultad de Ingeniería UNAM México D.F. a 16 de Agosto de 2006***

PROPIEDADES DE LA T. LAPLACE		TRANSFORMADAS MÁS COMUNES	
Linealidad	$L[af(t)+bg(t)]=aF(s)+bG(s)$	**f(t)**	**F(s)**
Diferenciación en el dominio del tiempo	$L\left[\frac{df(t)}{dt}\right]=sF(s)-f(0)$ $L\left[\frac{d^2f(t)}{dt^2}\right]=s^2F(s)-sf(0)-\frac{df(0)}{dt}$ $L\left[\frac{d^nf(t)}{dt^n}\right]=s^nF(s)-s^{n-1}f(0)-$ $-s^{n-2}f'(0)-...-s^0f^{n-1}(0)$	*Impulso unitario*	1
		1	$\frac{1}{s}$
Integración en el dominio del tiempo	$L\left[\int_0^t f(t)dt\right]=\frac{F(s)}{s}-\frac{f^{(-1)}(0^+)}{s}$ $L[f^{(-n)}(t)]=\frac{F(s)}{s^n}-\sum_{i=1}^{n}\frac{1}{s^{n-i+1}}f^{(-i)}(0^+)$	t^n	$\frac{n!}{s^{n+1}}$
Desplazamiento en el tiempo	$L[f(t-d)]=e^{-sd}F(s)$	e^{-at}	$\frac{1}{s+a}$
Teorema del valor inicial	$\lim_{t\to 0}f(t)=\lim_{s\to\infty}sF(s)$	t^ne^{-at}	$\frac{n!}{(s+a)^{n+1}}$
Teorema del valor final	$\lim_{t\to\infty}f(t)=\lim_{s\to 0}sF(s)$	Asenwt	$A\frac{w}{s^2+w^2}$
Teorema de convolución	$L\left[\int_0^\infty f(t)g(t-\tau)d\tau\right]=F(s)G(s)$	Acoswt	$A\frac{s}{s^2+w^2}$
Transformación de variables. Cambio de escala	$L[f(t/\alpha)]=\alpha F(\alpha s)$ $L[f(\alpha t)]=\frac{1}{\alpha}F(s/\alpha)$ α= constante positiva	$Ae^{-at}senwt$	$A\frac{w}{(s+a)^2+w^2}$
Traslación en el campo complejo	Si $L[f_1(t)]=F(s)$ y $L[f_2(t)]=F(s\pm\alpha)$ siendo α= constante, entonces $f_2(t)=e^{\mp\alpha t}f_1(t)$	$Ae^{-at}\cos wt$	$A\frac{s+a}{(s+a)^2+w^2}$
Diferenciación en el campo complejo	$L[tf(t)]=-\frac{dF(s)}{ds}$		

Tabla 1. Funciones comunes y su transformada de Laplace.
Fuente: http://www.disa.bi.ehu.es/spanish/asignaturas/15212/TEMA_4_TransformadaDeLaplace.pdf

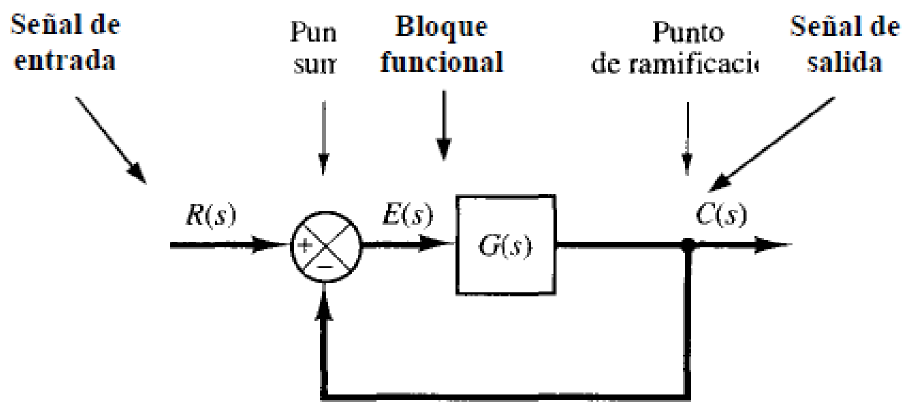
2.10. Diagramas en Bloques

Un sistema de control puede tener varios componentes. Para mostrar las funciones que lleva a cabo cada componente en la ingeniería de control, por lo general se usa una representación denominada diagrama de bloques.

Un diagrama de bloques de un sistema es una representación gráfica de las funciones que lleva a cabo cada componente. Tal diagrama muestra las relaciones existentes entre los diversos componentes.

En un diagrama de bloques se enlazan una con otra todas las variables del sistema, mediante bloques funcionales. El bloque funcional o simplemente bloque es un símbolo para representar la operación matemática que sobre la señal de entrada hace el bloque para producir la salida.

La figura muestra un elemento del diagrama de bloques. La punta de flecha que señala el bloque indica la entrada, y la punta de flecha que se aleja del bloque representa la salida. Tales flechas se conocen como señales.



Observe que las dimensiones de la señal de salida del bloque son las dimensiones de la señal de entrada multiplicadas por las dimensiones de la función de transferencia en el bloque.

Un diagrama de bloques contiene información relacionada con el comportamiento dinámico, pero no incluye información de la construcción física del sistema. En consecuencia, muchos sistemas diferentes y no relacionados pueden representarse mediante el mismo diagrama de bloques.

2.11. Reducción de un diagrama de bloques

El autor tomo la explicación textual del tema de**:** NUÑEZ, M. J. (12 de 09 de 2012).

Es importante señalar que los bloques pueden conectarse en serie, sólo si la entrada de un bloque no se ve afectada por el bloque siguiente. Si hay efectos de carga entre los componentes, es necesario combinarlos en un bloque único.

Un diagrama de bloques complicado que contenga muchos lazos de realimentación se simplifica mediante un reordenamiento paso a paso mediante las reglas del álgebra de los diagramas de bloques. Algunas de estas reglas importantes aparecen en la tabla2 y se obtienen escribiendo la misma ecuación en formas distintas.

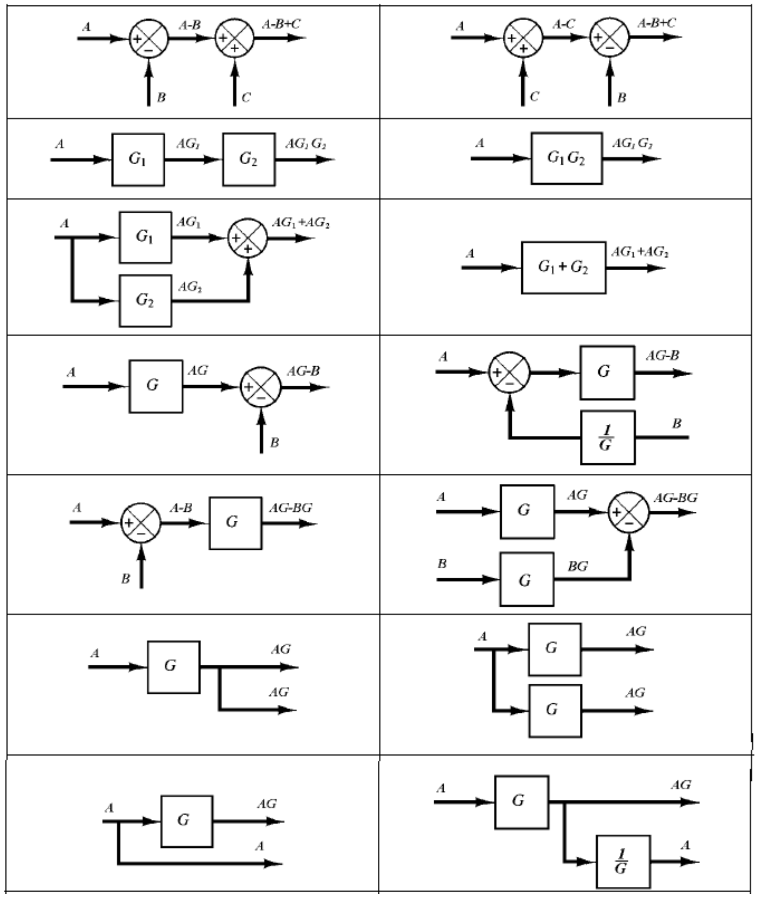


Tabla 2 : Reglas del Algebra de bloques
Fuente: NUÑEZ, M. J. (12 de 09 de 2012). *http://gama.fime.uanl.mx.* Recuperado el 12 de 09 de 2012, de http://gama.fime.uanl.mx/~agarcia/materias/ingco/apclas/03%20-%20Diagramas%20de%20Bloques.pdf

La simplificación de un diagrama de bloques mediante reordenamientos y sustituciones reduce de manera considerable la labor necesaria para el análisis matemático subsecuente. Sin embargo, debe señalarse que, conforme se simplifica el diagrama de bloques, las funciones de transferencia de los bloques nuevos se vuelven más complejas, debido a que se generan polos y ceros nuevos.

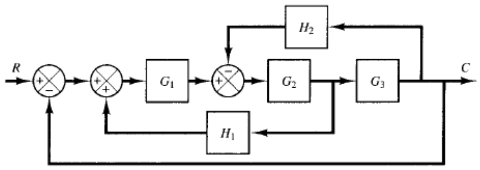
Al simplificar un diagrama de bloques, hay que recordar lo siguiente:

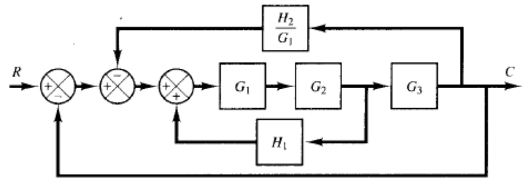
a) El producto de las funciones de transferencia en la dirección de la trayectoria directa debe ser el mismo.

b) El producto de las funciones de transferencia alrededor del lazo debe ser el mismo.

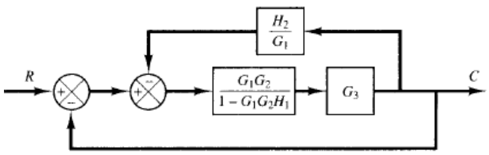
Ejemplo:

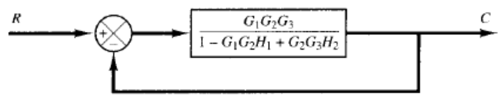
Simplificar el siguiente diagrama de bloques



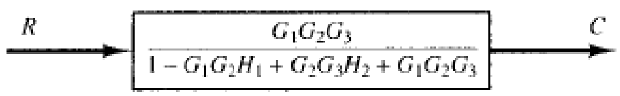


A **B**





C **D**

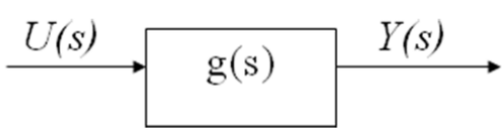


E

2.12. Función de transferencia – sistemas de primer orden

Introducción

Trabajar en el dominio de Laplace no solamente es útil para la resolución matemática de ecuaciones sino que se presta especialmente para ser utilizado con el concepto de función de transferencia. En general un proceso recibe una entrada u(t) y genera una salida y(t). Si llevamos estas señales al dominio de Laplace tendremos una entrada U(s) que genera una salida Y(s). La función que relaciona salida con entrada se denomina función de transferencia g(s).



De modo que $Y(s) = g(s) \times U(s)$.

2.12.1. Sistemas de Primer Orden

Se denominan sistemas de primer orden a aquellos en los que en la ecuación general aparece solamente la derivada primera del lado izquierdo (el de la variable de estado). O sea que se reducen al formato siguiente:

$$\tau \frac{dy}{dt} + y = k\ u$$

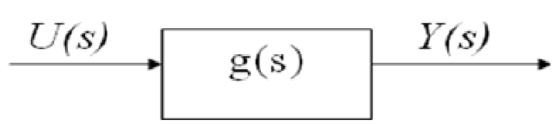
Donde k se denomina ganancia del proceso y τ es la constante de tiempo del sistema. En general encontraremos que la ecuación está escrita en función de las variables "desviación" respecto al valor de estado estacionario. Por lo tanto en general $y(0) = 0$, $u(0) = 0$. Tomando transformadas de Laplace

$$\begin{aligned}
\tau\left[sY(s) - y(0)\right] + Y(s) &= kU(s) \\
\tau s Y(s) + Y(s) &= kU(s) \\
(\tau s + 1) Y(s) &= kU(s)
\end{aligned}$$

$$\begin{aligned}
Y(s) &= \frac{k}{\tau s + 1} U(s) \\
Y(s) &= g(s) U(s) \\
g(s) &= \frac{k}{\tau s + 1}
\end{aligned}$$

2.12.2. Respuestas de sistemas de primer orden a diferentes entradas

Seguimos manejándonos con el esquema



Dónde: $$g(s) = \frac{k}{\tau\ s + 1}$$

Escalón de magnitud ΔU a tiempo $t = 0$

Sabemos que $$\mathcal{L}[\Delta U] = \frac{\Delta U}{s}$$

Por lo tanto $$Y(s) = \frac{k\ \Delta U}{s(\tau s + 1)}$$

Tomando anti transformadas $$\mathcal{L}^{-1}\left[\frac{1}{s(\tau s + 1)}\right] = 1 - e^{-t/\tau}$$

O bien

$$\boxed{y(t) = k\Delta U\left[1 - e^{-t/\tau}\right]}$$

Que escrito en forma adimensional es: $$\frac{y(t)}{k\Delta U} = \left[1 - e^{-t/\tau}\right]$$

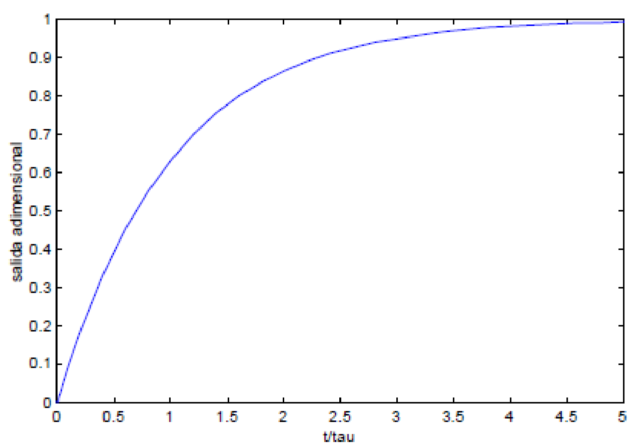


Figura # 2.12 : Respuesta paso de un Sistema Lineal de Primer Orden

2.12.3. Respuesta rampa de un sistema de primer orden

Al considerar que en la ecuación diferencial $\tau\frac{dy}{dt}+y=k\ u$, la variable de entrada es perturbada con un cambio rampa, es decir que X (t) = rt, entonces se puede escribir que:

$$\tau\frac{dY(t)}{dt}+Y(t)=Krt \qquad \textbf{(a)}$$

Al resolver esta ecuación se obtiene como solución la siguiente respuesta para Y(t):

$$Y(t)=Kr\left[\tau\exp\left(-\frac{t}{\tau}\right)+t-\tau\right] \qquad \textbf{(b)}$$

La ecuación (b) se obtiene aplicando el factor integrante a (a) y una integración indefinida da como solución general

$$Y(t)=Kr(t-\tau)+A_1\exp\left(-\frac{t}{\tau}\right) \qquad \textbf{(c)}$$

Evaluando la ecuación (c) para la condición inicial $Y(0) = 0$, se obtiene que el valor de la constante de integración es $A = Kr\ \tau$ y, con ello, la solución dada por (b).

La figura2.13 muestra, gráficamente, el perfil de la respuesta rampa de un sistema lineal de primer orden. Se puede observar un comportamiento lineal y paralelo a la rampa de entrada después de un determinado tiempo, que aproximadamente es cinco veces la constante de tiempo.

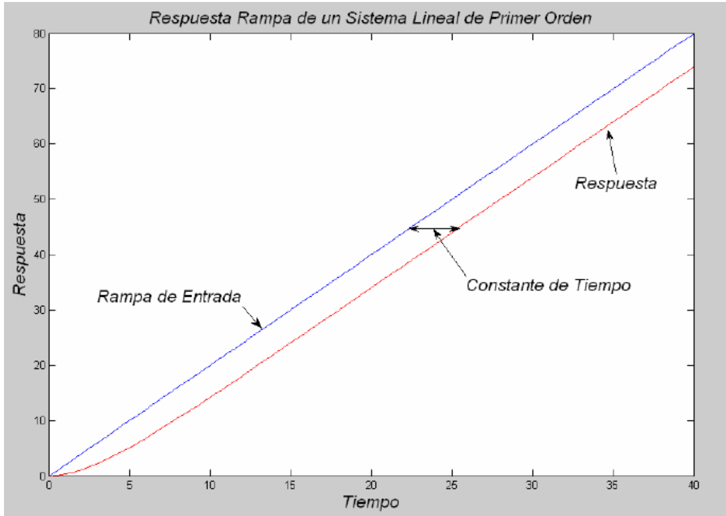


Figura # 2.13: Respuesta Rampa de un Sistema de Primer Orden (*K* = *3*, τ = *3*, *r* = *2*)

Se resalta en la Figura 2.13 el atraso de la respuesta con respecto a la rampa de entrada y se demuestra con la ecuación (b) que dicho atraso es igual al tiempo correspondiente a la constante de tiempo.

2.12.4. Respuesta seno de un sistema de primer orden

Al considerar que en la ecuación diferencial $\tau\frac{dy}{dt}+y=k\,u$, la variable de entrada es perturbada con un cambio seno, es decir que X (t) = ASen(wt) , entonces se puede escribir que:

$$\tau\frac{dY(t)}{dt}+Y(t)=KASen(wt) \quad \textbf{(a)}$$

Al resolver la ecuación (a) se obtiene como solución la siguiente respuesta para *Y(t):*

$$Y(t)=\frac{KAw\tau}{1+(w\tau)^2}\exp\left(-\frac{t}{\tau}\right)+\frac{KA}{\sqrt{1+(w\tau)^2}}Sen(wt+\theta) \quad \textbf{(b)}$$

Siendo, $\theta = \tan^{-1}(-w\tau)$

La ecuación (b) se obtiene aplicando el factor integrante a (a) y una integración indefinida da como solución general.

$$Y(t) = \frac{KA}{1+(w\tau)^2}\left[Sen(wt) - (w\tau)Cos(wt)\right] + A_1 \exp\left(-\frac{t}{\tau}\right) \quad \textbf{(c)}$$

Evaluando la ecuación (c) para la condición inicial $Y(0) = 0$, se obtiene que el valor de la constante de integración es $A_1 = \frac{KAw\tau}{1+(w\tau)^2}$ y, con ello, la solución dada por (b)

La figura2.14 se muestra el perfil gráfico de la respuesta seno de un sistema lineal de primer orden. Se observa una corta región inicial con una ligera inflexión que se explica por la influencia del término exponencial en la expresión (b) que corresponde a la respuesta del sistema. Cuando este primer término exponencial es de un valor despreciable, la respuesta muestra un perfil definidamente sinusoidal que se distingue por las siguientes características:

- Su frecuencia es igual a la del seno de entrada.
- Su amplitud es el coeficiente del término sinusoidal y es dependiente de la frecuencia del seno de entrada, además de los otros parámetros incluidos en el mismo, es decir que:

$$A_{respuesta} = \frac{KA}{\sqrt{1+(w\tau)^2}}$$

Es atrasada con respecto al seno de entrada, lo que se mide mediante un ángulo fase que también es un valor que depende de la frecuencia del seno de entrada

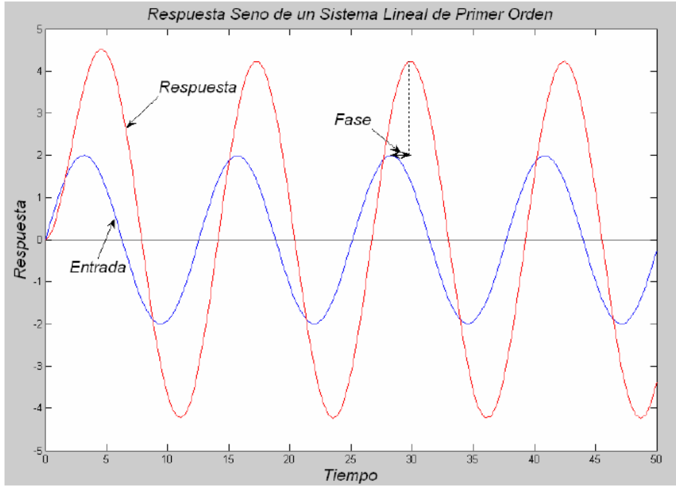


Figura # 2.14 Respuesta Seno de un Sistema de Primer Orden (*K = 3,* τ *= 2, A = 2, w= 0.5*)

Cada una de estas características es importante porque constituyen los fundamentos para analizar la dinámica de un sistema cualquiera en el dominio de la frecuencia que a su vez se utiliza para el diseño de sistemas de control.

2.13. Función de transferencia – sistemas de Segundo orden **Introducción**

Existe en la actualidad un sinnúmero de métodos de diseño de distinto origen que permiten desarrollar sistemas de control dentro de una amplia gama de características y/o posibilidades.

Pero poco, proporcionalmente, se ha trabajado sobre el problema del reconocimiento matemático de un sistema pre armado, que sea eminentemente práctico. Es decir, que una vez que se ha podido implementar el subsistema o planta a controlar, a veces no se conocen con certeza sus principales parámetros, cuál es su función de transferencia equivalente, etc. En otras circunstancias, puede conocerse su estructura, pero no los valores de los parámetros que la componen. Todo ello genera a veces dificultades insalvables para realizar un control efectivo y confiable.

Por consiguiente, poder determinar, aunque sea en forma aproximada los parámetros más importantes de una función de transferencia de un sistema se convierte en una necesidad insoslayable.

Lo que sigue muestra una forma sencilla de identificación de sistemas de segundo orden o de aquellos que puedan representarse en forma aproximada como tal, utilizando la respuesta en el tiempo a un salto escalón de entrada, a partir de una idea de Draper, Mc Kay y Lees publicada en el año 1953.

2.13.1. Característica

El autor tomo la explicación textual del tema de**:** Rivero, R. A. (15 de 09 de 2012).

La identificación de un sistema consiste en la determinación de la función de transferencia de la misma o de sus parámetros fundamentales, a partir de mediciones experimentales.

En particular, este trabajo desarrolla una metodología de identificación de sistemas de segundo orden, basada en las características de la respuesta en el tiempo *Y(t)* a la excitación de un salto escalón de entrada *X(t)*, tal como lo indica la figura 2.15

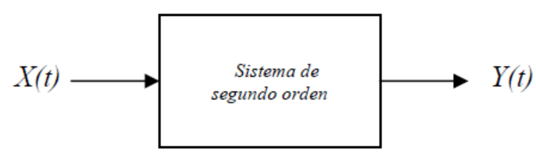


Figura # 2.15: Sistema de segundo orden
Fuente: Rivero, R. A. (15 de 09 de 2012). *www.edutecne.utn.edu.ar*. Recuperado el 17 de 09 de 2012, de http://www.edutecne.utn.edu.ar/tutoriales/identificacion-sistemas-segundo-orden.pdf

Se ha elegido este método de identificación porque presenta las siguientes características:

- Obtención de rápidos resultados aproximados.
- Simplicidad para analizar y entender.
- La respuesta en el tiempo a un salto escalón es posiblemente una de las más fáciles de obtener en un sistema cualquiera.
- Muchas veces en una planta, a pesar de que su función de transferencia tenga más de dos polos, la respuesta al salto escalón de entrada puede ser representada en forma aproximada por la respuesta de un sistema de segundo orden. Ello es posible porque frecuentemente los otros polos adicionales están ubicados más lejos del eje imaginario que estos polos dominantes y por ende la influencia de los mismos en la respuesta en el tiempo resulta menor. (En particular ello se

cumple significativamente cuando la relación entre las partes reales de los otros polos y las partes reales de los polos dominantes es mayor de 5 y no hay cero cercanos) Por ello, se la puede representar aproximadamente por un sistema de segundo orden conocido como de **modos dominantes**.

2.13.2. Metodología e implementación

El autor tomo la explicación textual del tema de**:** Rivero, R. A. (15 de 09 de 2012).

Se parte suponiendo:

a) que un sistema de segundo orden está representado por la ecuación clásica. (Rivero, R. A. 15 de 09 de 2012).

$$T(s) = \frac{y(s)}{x(s)} = \frac{\omega_n^2}{s^2 + 2\varsigma\omega_n s + \omega_n^2} \quad (1)$$

b) que la entrada es un salto escalón de amplitud ***Y0*** de la forma

$$x(s) = \frac{Y_0}{s} \quad (2)$$

c) y que las condiciones iniciales son nulas,

Entonces, la respuesta del sistema a un salto escalón de amplitud ***Y0*** está dada por:

$$Y(s) = \frac{Y_0}{s} \times \frac{\omega_n^2}{s^2 + 2\varsigma\omega_n s + \omega_n^2} \quad (3)$$

Se reconoce en estos casos tres formas posibles de respuestas distintas a saber:

a) Sub amortiguada u oscilante (para ζ menor que uno)

b) Critica (para ζ igual a uno)

c) Sobre amortiguada o no oscilante (para ζ mayor que uno)

Cada una de ellas presenta características distintas que la diferencian entre sí. Sin embargo, para el reconocimiento de los parámetros del sistema, se utiliza otra división basada en las siguientes situaciones prácticas:

1. Respuestas de sistemas oscilantes con sobre picos significativos (ζ menor que 0,5)

2. Respuestas de sistemas sin sobre picos o con sobre picos de poco valor (ζ entre 0,5 y

2)

3. Respuestas de sistemas sobre amortiguados (ζ mayor que 2)

Se presenta a continuación, para cada una de ellas, el método de reconocimiento de los parámetros y su justificación. (Rivero, R. A. 15 de 09 de 2012).

En todos los casos se supone que las condiciones iniciales del sistema son nulas.

2.13.3. Respuesta oscilante (para ζ igual o menor que 0,5)

El autor tomo la explicación textual del tema de**:** Rivero, R. A. (15 de 09 de 2012).

La respuesta de un sistema oscilante a una entrada salto escalón de amplitud Y0 es de la forma tal como indica en la figura2.16

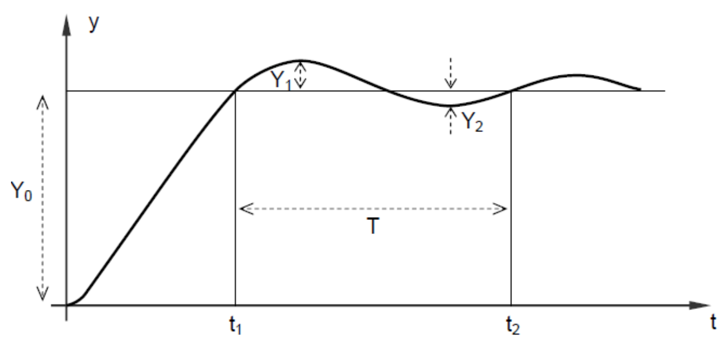


Figura # 2.16: Respuesta oscilante del sistema de segundo orden a la excitación de un salto escalón de amplitud Y0

Fuente: Rivero, R. A. (15 de 09 de 2012). *www.edutecne.utn.edu.ar.* Recuperado el 17 de 09 de 2012, de http://www.edutecne.utn.edu.ar/tutoriales/identificacion-sistemas-segundo-orden.pdf

En la práctica, es fácil distinguir los sobre picos primero (Y1) y segundo (Y2) cuando el valor de ζ es igual o menor que 0,5. Cuando ζ es mayor que 0,5, pero menor que 1, si bien matemáticamente es posible determinar la oscilación de la respuesta, a veces los valores de los sobre picos Y1 e Y2 no resultan lo suficientemente notorios en los métodos de medición, por lo que una estimación de los parámetros utilizando estos valores pueden llevar a errores poco aceptables.

En consecuencia, en esta primera parte, el objetivo es encontrar valores aproximados para ωn y ζ, que representen de la mejor forma posible a los parámetros fundamentales del sistema de segundo orden, cuando ζ sea igual o menor que 0,5.

Para ello, se utilizan los valores de Y0, Y1 y Y2 (que son las amplitudes del salto escalón de entrada, del primer sobre pico y segundo sobre pico de la respuesta en el tiempo respectivamente) y de T (que es el período de una oscilación de la respuesta en el tiempo).

Entonces, la respuesta en el tiempo y(t) cumple la ecuación:

$$y(t) = Y_0 - Y_0 \times e^{-\zeta\omega_n t}\left[\cos(\omega_n\sqrt{1-\zeta^2}t) + \frac{\zeta}{\sqrt{1-\zeta^2}}sen(\omega_n\sqrt{1-\zeta^2}t)\right] \quad (4)$$

Puede demostrarse a partir de la ecuación (4), que:

$$\frac{Y_1}{Y_0} = e^{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \quad \text{e} \quad \frac{Y_2}{Y_0} = e^{-\frac{2\zeta\pi}{\sqrt{1-\zeta^2}}} \quad (5)$$

De donde se obtienen dos posibles valores para ζ, utilizando las expresiones:

$$\zeta = \frac{\ln(Y_1/Y_0)}{\sqrt{\pi^2 + (\ln(Y_1/Y_0))^2}} \quad \text{y} \quad \zeta = \frac{\ln(Y_2/Y_0)}{\sqrt{4\pi^2 + (\ln(Y_2/Y_0))^2}} \quad (6)$$

En la práctica se adopta el valor promedio de ambos.

Los valores de las ecuaciones (6) también pueden obtenerse utilizando la tabla 3 o el grafico 2.17

El valor de ωn se obtiene a partir de la expresión:

$$T = \frac{2\pi}{\omega_n\sqrt{1-\zeta^2}} \quad \text{o sea} \quad \omega_n = \frac{2\pi}{T\sqrt{1-\zeta^2}} \quad (7)$$

Valores porcentual de sobrepico	Valor de ζ calculado con el primer sobrepico	Valor de ζ calculado con el segundo sobrepico
0,05	0,690	0,431
0,10	0,591	0,344
0,15	0,517	0,289
0,20	0,456	0,248
0,25	0,404	0,216
0,30	0,358	0,188
0,35	0,317	0,165
0,40	0,280	0,144
0,45	0,246	0,126
0,50	0,216	0,110
0,55	0,187	0,095
0,60	0,161	0,081
0,65	0,136	0,068
0,70	0,113	0,057
0,75	0,091	0,046
0,80	0,071	0,036
0,85	0,052	0,026
0,90	0,034	0,017

Tabla 3 : Tabla de valores de las ecuaciones número 6
Fuente: Rivero, R. A. (15 de 09 de 2012). *www.edutecne.utn.edu.ar*. Recuperado el 17 de 09 de 2012, de http://www.edutecne.utn.edu.ar/tutoriales/identificacion-sistemas-segundo-orden.pdf

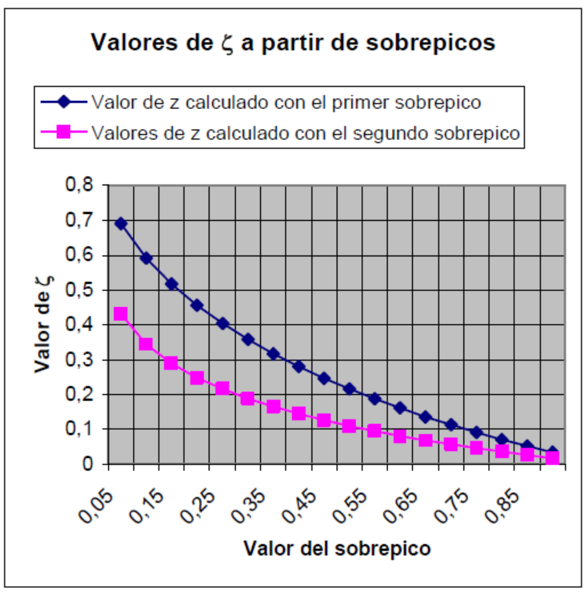


Figura # 2.17: Gráfico de valores de las ecuaciones número 6
Fuente: Rivero, R. A. (15 de 09 de 2012). *www.edutecne.utn.edu.ar*. Recuperado el 17 de 09 de 2012, de http://www.edutecne.utn.edu.ar/tutoriales/identificacion-sistemas-segundo-orden.pdf

2.13.4. Respuesta para amortiguamientos próximos al crítico (ζ entre 0,5 y 2)

El autor tomo la explicación textual del tema de**:** Rivero, R. A. (15 de 09 de 2012).

La forma de la respuesta de un sistema con amortiguamiento crítico o próximo al crítico es:

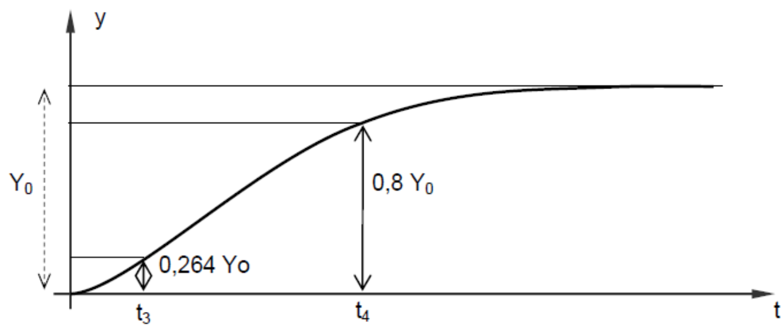


Figura # 2.18: Forma aproximada de la respuesta del sistema de segundo orden a la excitación de un salto escalón de amplitud Y0 cuando el valor de ζ está comprendido entre 0,5 y 2.
Fuente: Rivero, R. A. (15 de 09 de 2012). *www.edutecne.utn.edu.ar*. Recuperado el 17 de 09 de 2012, de http://www.edutecne.utn.edu.ar/tutoriales/identificacion-sistemas-segundo-orden.pdf

Se puede demostrar que una característica importante de la respuesta en estos casos es que si se toma como referencia a los valores de salida indicados en la figura 2.18, a saber:

$$Y_4 = Y_0 \times (1 - 4 \times e^{-3}) = Y_0 \times 0{,}80079$$
$$Y_3 = Y_0 \times (1 - 2 \times e^{-1}) = Y_0 \times 0{,}26416 \quad (8)$$

Y si se conocen los tiempos t3 y t4 respectivos, se cumple que:

$$\frac{t_4}{t_3} = 3 \quad (9)$$

Cuando ζ **es igual a uno**.

Por ello, si la relación t4/t3 es menor que tres, significa que ζ es menor que uno y si resulta mayor que tres, el valor de ζ es mayor que uno. Esta característica permite en consecuencia determinar los parámetros ζ y ω ***n*** del sistema de segundo orden, a partir de las dos posibles respuestas en el tiempo a una excitación escalón.

En efecto, la respuesta en el tiempo de un sistema de segundo orden a un salto escalón está dada por la ecuación:

$$y(t) = Y_0 - Y_0 \times e^{-\zeta\omega_n t}\left[\cos(\omega_n\sqrt{1-\zeta^2}\,t) + \frac{\zeta}{\sqrt{1-\zeta^2}}\,sen(\omega_n\sqrt{1-\zeta^2}\,t)\right] \quad (10)$$

Cuando ζ es menor que uno, y por

$$y(t) = Y_0 + Y_0 \times \frac{1}{2\sqrt{\zeta^2-1}\times(\zeta+\sqrt{\zeta^2-1})}e^{-(\zeta+\sqrt{\zeta^2-1})\omega_n t} - \\ - Y_0 \times \frac{1}{2\sqrt{\zeta^2-1}\times(\zeta-\sqrt{\zeta^2-1})}e^{-(\zeta-\sqrt{\zeta^2-1})\omega_n t} \quad (11)$$

Cuando ζ es mayor que uno.

A partir de estas ecuaciones, se puede determinar el valor de ζ utilizando la relación *t4/t3* en la tabla 2.

El valor de ω***n*** queda dado entonces, a partir de la tabla 4, utilizando la tercera columna, la relación *t4/t3* y el valor del tiempo *t4* tal que:

$$\omega_n = \frac{\omega_n t_4}{t_4} \quad (12)$$

t_4/t_3	ζ	$\omega_n t_4$
2,21	0,5	1,89
2,33	0,6	2,05
2,46	0,7	2,24
2,62	0,8	2,46
2,80	0,9	2,71
2,90	0,95	2,85
3,00	1	3,00
3,13	1,05	3,19
3,20	1,1	3,31
3,37	1,2	3,63
3,57	1,3	3,96
3,72	1,4	4,30
3,90	1,5	4,64
4,04	1,6	4,97
4,15	1,7	5,30
4,25	1,8	5,63
4,35	1,9	5,97
4,43	2	6,3

Tabla 4: Tabla de valor de ωn

Fuente: Rivero, R. A. (15 de 09 de 2012). *www.edutecne.utn.edu.ar.* Recuperado el 17 de 09 de 2012, de http://www.edutecne.utn.edu.ar/tutoriales/identificacion-sistemas-segundo-orden.pdf

Los valores anteriores también pueden obtenerse del gráfico 2.19, donde se presentan las curvas de t4/t3, ω n t3, y ω nt4 en función de ζ .

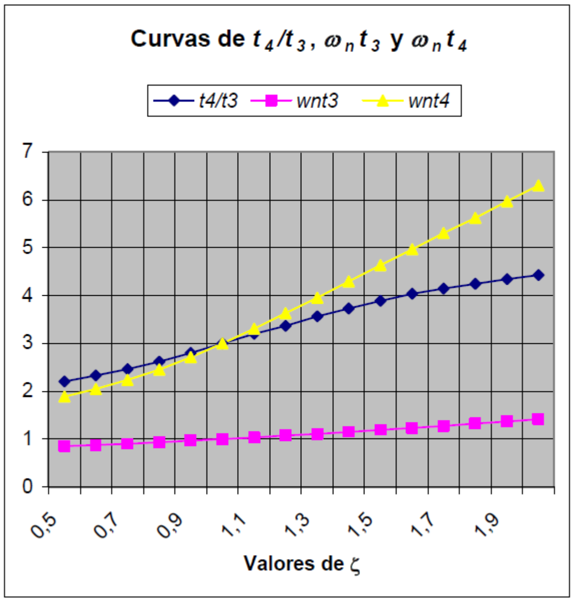


Figura # 2.19: Curvas de valores de ωn

Fuente: Rivero, R. A. (15 de 09 de 2012). *www.edutecne.utn.edu.ar.* Recuperado el 17 de 09 de 2012, de http://www.edutecne.utn.edu.ar/tutoriales/identificacion-sistemas-segundo-orden.pdf

2.13.5. Respuesta para sistemas sobre amortiguamientos (ζ mayor que 2)

El autor tomo la explicación textual del tema de**:** Rivero, R. A. (15 de 09 de 2012).

La respuesta de un sistema sobre amortiguado a una entrada salto escalón de amplitud Y0 a la entrada, es de la forma como indica la figura 2.20

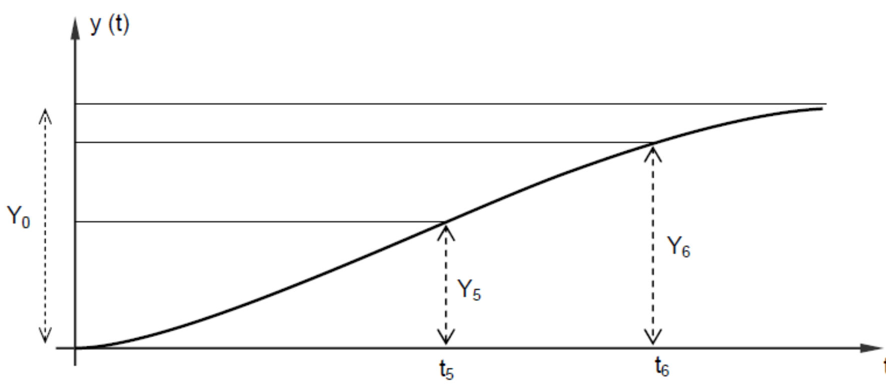


Figura # 2.20: Forma aproximada de la respuesta del sistema de segundo orden a la excitación de un salto escalón de amplitud Y0 cuando el valor de ζ es mayor que 2.

Fuente: Rivero, R. A. (15 de 09 de 2012). *www.edutecne.utn.edu.ar.* Recuperado el 17 de 09 de 2012, de http://www.edutecne.utn.edu.ar/tutoriales/identificacion-sistemas-segundo-orden.pdf

Su ecuación está dada por:

$$y(t) = Y_0 + Y_0 \times \frac{1}{2\sqrt{\zeta^2 - 1} \times (\zeta + \sqrt{\zeta^2 - 1})} e^{-(\zeta + \sqrt{\zeta^2 - 1})\omega_n t}$$
$$- Y_0 \times \frac{1}{2\sqrt{\zeta^2 - 1} \times (\zeta - \sqrt{\zeta^2 - 1})} e^{-(\zeta - \sqrt{\zeta^2 - 1})\omega_n t} \quad (13)$$

En estos casos, al ser el coeficiente de amortiguamiento ζ mayor que 2, el sistema posee polos reales y negativos, bastantes distanciados entre sí, por lo menos 3,7 veces el uno del otro.

Ellos son:

$$p_1 = \zeta + \sqrt{\zeta^2 - 1}$$
$$p_2 = \zeta - \sqrt{\zeta^2 - 1} \quad (14)$$

Debido a esta particularidad, resulta más accesible, como objetivo de identificación en este caso, encontrar dos valores aproximados para los polos p1 y p2 del sistema, tal que representen de la mejor forma posible a los parámetros fundamentales del sistema de segundo orden.
Por lo tanto la ecuación representativa del sistema de segundo orden a encontrar es:

$$T(s) = \frac{y(s)}{x(s)} = \frac{p_1 \times p_2}{s^2 + (p_1 + p_2)s + p_1 \times p_2} \quad (15)$$

Para ello, conociendo que la respuesta en el tiempo al salto escalón está prácticamente determinado por el polo de menor valor absoluto *p2*, cuando la amplitud de la salida es superior a la mitad del valor total, el cálculo de los coeficientes se facilita determinando primero el polo *p2,* como si fuera el único existente. Para ello se determinan dos valores de ordenadas de la curva de respuesta en frecuencia, *Y5(t5)* e *Y6(t6)*, tal que ambos sean mayores que el 50 % de la amplitud final de la respuesta, según indica la figura .

Entonces, el polo *p2* está dado por:

$$p_2 = \frac{\ln(1 - (Y_5 / Y_0)) - \ln(1 - (Y_6 / Y_0))}{t_5 - t_6} \quad (16)$$

Y luego el valor del polo *p1* como:

$$p_1 = p_2 \times \frac{(1-(Y_5/Y_0))\times e^{p_2 t_5}}{(1-(Y_5/Y_0))\times e^{p_2 t_5} - 1} \quad (17)$$

O

$$p_1 = p_2 \times \frac{(1-(Y_6/Y_0))\times e^{p_2 t_6}}{(1-(Y_6/Y_0))\times e^{p_2 t_6} - 1} \quad (18)$$

Tomándose en la práctica el valor promedio entre ambas expresiones (17) y (18).

Finalmente, de las ecuaciones (15), (16), (17) y (18), si se desea, pueden encontrarse los valores de ζ y ωn , según:

$$\omega_n = \sqrt{p_1 \times p_2} \quad (19)$$

y

$$\zeta = \frac{p_1 + p_2}{2\sqrt{p_1 \times p_2}} \quad (20)$$

2.14. El programa Matlab

El autor tomo la explicación textual del tema de**:** Javier García de Jalón, J. I. (2005).

MATLAB es el nombre abreviado de “MATrix LABoratory”. MATLAB es un programa para realizar cálculos numéricos con vectores y matrices. Como caso particular puede también trabajar con números escalares −tanto reales como complejos−, con cadenas de caracteres y con otras estructuras de información más complejas. Una de las capacidades más atractivas es la de realizar una amplia variedad de gráficos en dos y tres dimensiones. MATLAB tiene también un lenguaje de programación propio.

MATLAB es un gran programa de cálculo técnico y científico. Para ciertas operaciones es muy rápido, cuando puede ejecutar sus funciones en código nativo con los tamaños más adecuados para aprovechar sus capacidades de vectorización. En otras aplicaciones resulta bastante más lento que el código equivalente desarrollado en C/C++ o Fortran.

En la versión 6.5, MATLAB incorporó un acelerador JIT (Just In Time), que mejoraba significativamente la velocidad de ejecución de los ficheros *.m en ciertas circunstancias, por ejemplo cuando no se hacen llamadas a otros ficheros *.m, no se utilizan estructuras y clases, etc. Aunque limitado en ese momento, cuando era aplicable mejoraba sensiblemente la velocidad, haciendo innecesarias ciertas técnicas utilizadas en versiones anteriores como la vectorización de los algoritmos. En cualquier caso, el lenguaje de programación de MATLAB siempre es una magnífica herramienta de alto nivel para desarrollar aplicaciones técnicas, fácil de utilizar y que, como ya se ha dicho, aumenta significativamente la productividad de los programadores respecto a otros entornos de desarrollo.

MATLAB dispone de un código básico y de varias librerías especializadas (toolboxes) tal como se muestra en la figura 2.23. En estos apuntes se hará referencia exclusiva al código básico. MATLAB se puede arrancar como cualquier otra aplicación de Windows, clicando dos veces en el icono correspondiente en el escritorio o por medio del menú Inicio. Al arrancar MATLAB se abre una ventana similar a la mostrada en la Figura 2.21. Ésta es la vista que se obtiene eligiendo la opción Desktop Layout/Default, en el menú View. Como esta configuración puede ser cambiada fácilmente por el usuario, es posible que en muchos casos concretos lo que aparezca sea muy diferente. En cualquier caso, una vista similar se puede conseguir con el citado comando View/Desktop Layout/ Default.

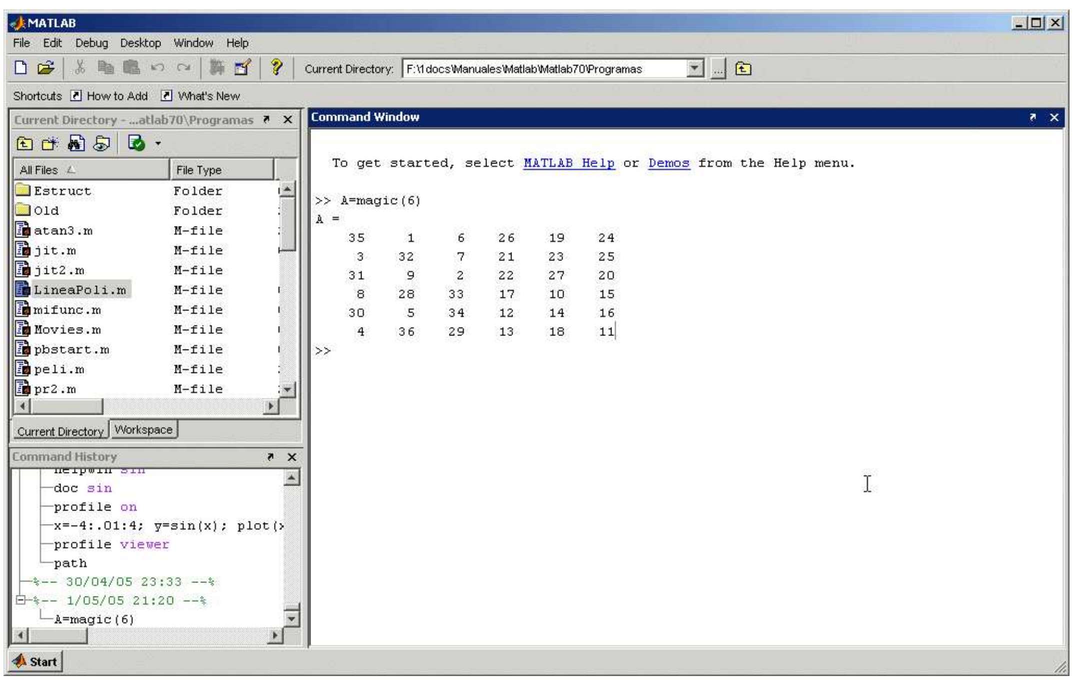


Figura # 2.21: Ventana inicial de MATLAB
Fuente: Javier García de Jalón, J. I. (2005). *Aprenda Matlab 7.0 como si estuviera en primero*. Madrid: Escuela Técnica Superior de Ingenieros Industriales Universidad Politécnica de Madrid.

La parte más importante de la ventana inicial es la *Command Window*, que aparece en la parte derecha superior.

En esta sub-ventana es donde se ejecutan los comandos de MATLAB, a continuación del *prompt* (aviso) característico (>>), que indica que el programa está preparado para recibir instrucciones.

En la pantalla mostrada en la Figura 2.21 se ha ejecutado el comando A=magic(6), mostrándose a continuación el resultado proporcionado por MATLAB. (Javier García de Jalón, J. I. 2005)

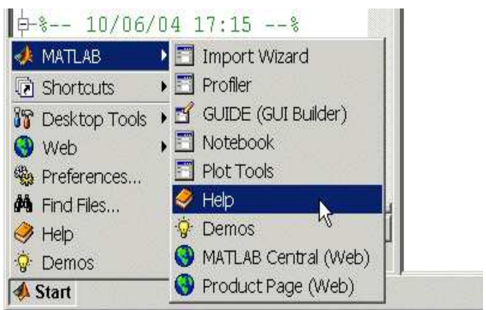


Figura # 2.22: Menú *Start/ Matlab*

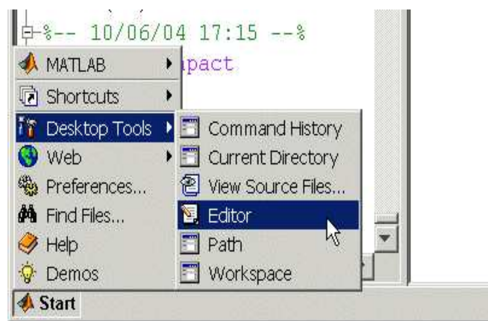


Figura # 2.23: Menú *Start /Desktop Tools*

Fuente: Javier García de Jalón, J. I. (2005). *Aprenda Matlab 7.0 como si estuviera en primero*. Madrid: Escuela Técnica Superior de Ingenieros Industriales Universidad Politécnica de Madrid.

En la parte superior izquierda de la pantalla aparecen dos ventanas también muy útiles: en la parte superior aparece la ventana *Current Directory*, que se puede alternar con *Workspace* clicando en la pestaña correspondiente. La ventana *Current Directory* muestra los ficheros del directorio activo o actual. El directorio activo se puede cambiar desde la *Command Window*, o desde la propia ventana (o desde la barra de herramientas, debajo de la barra de menús) con los métodos de navegación de directorios propios de Windows. Clicando dos veces sobre alguno de los ficheros *.m del directorio activo se abre el editor de ficheros de MATLAB, herramienta fundamental para la programación sobre la que se volverá en las próximas páginas. El *Workspace* contiene información sobre todas las variables que se hayan definido en esta sesión y permite ver y modificar las matrices con las que se esté trabajando.

En la parte inferior derecha aparece la ventana *Command History* que muestra los últimos comandos ejecutados en la *Command Window*. Estos comandos se pueden volver a ejecutar haciendo doble clic sobre ellos. Clicando sobre un comando con el botón derecho del ratón se muestra un menú contextual con las posibilidades disponibles en ese momento. Para editar uno de estos comandos hay que copiarlo antes a la *Command Window.*

En la parte inferior izquierda de la pantalla aparece el botón Start, con una función análoga a la del botón Inicio de Windows. Start da acceso inmediato a ciertas capacidades del programa. La Figura 2.22 muestra las posibilidades de Start/MATLAB, mientras que la Figura 2.23 muestra las opciones de Start/Desktop Tools, que permiten el acceso a las principales componentes o módulos de MATLAB.

El menú Desktop realiza un papel análogo al botón Start, dando acceso a los módulos o componentes de MATLAB que se tengan instalados. Puede hacerse que al arrancar MATLAB se ejecute automáticamente un fichero, de modo que aparezca por ejemplo un saludo inicial personalizado. Esto se hace mediante un fichero de comandos que se ejecuta de modo automático cada vez que se entra en el programa (el fichero startup.m, que debe estar en un directorio determinado, por ejemplo C:\Matlab701\Work. Para apreciar desde el principio la potencia de MATLAB, se puede comenzar por escribir en la *Command Window* la siguiente línea, a continuación del *prompt.* Al final hay que pulsar *intro.*

```
>> A=rand(6), B=inv(A), B*A
A =
0.9501 0.4565 0.9218 0.4103 0.1389 0.0153
0.2311 0.0185 0.7382 0.8936 0.2028 0.7468
0.6068 0.8214 0.1763 0.0579 0.1987 0.4451
0.4860 0.4447 0.4057 0.3529 0.6038 0.9318
0.8913 0.6154 0.9355 0.8132 0.2722 0.4660
0.7621 0.7919 0.9169 0.0099 0.1988 0.4186
B =
 5.7430   2.7510  3.6505  0.1513 -6.2170 -2.4143
-4.4170  -2.5266 -1.4681 -0.5742  5.3399  1.5631
-1.3917  -0.6076 -2.1058 -0.0857  1.5345  1.8561
-1.6896  -0.7576 -0.6076 -0.3681  3.1251 -0.6001
-3.6417  -4.6087 -4.7057  2.5299  6.1284  0.9044
 2.7183   3.3088  2.9929 -0.1943 -5.1286 -0.6537
ans =
 1.0000  0.0000        0   0.0000   0.0000 -0.0000
 0.0000  1.0000   0.0000   0.0000  -0.0000  0.0000
      0       0   1.0000  -0.0000  -0.0000  0.0000
 0.0000       0  -0.0000   1.0000  -0.0000  0.0000
-0.0000  0.0000  -0.0000  -0.0000   1.0000  0.0000
-0.0000 -0.0000  -0.0000  -0.0000  -0.0000  1.0000
```

En realidad, en la línea de comandos anterior se han escrito tres instrucciones diferentes, separadas por comas. Como consecuencia, la respuesta del programa tiene tres partes también, cada una de ellas correspondiente a una de las instrucciones. Con la primera instrucción se define una matriz cuadrada (6×6) llamada A, cuyos elementos son números aleatorios entre cero y uno (aunque aparezcan sólo 4 cifras, han sido calculados con 16 cifras de precisión). En la segunda instrucción se define una matriz B que es igual a la inversa de A. Finalmente se ha multiplicado B por A, y se comprueba que el resultado es la matriz.

Otro de los puntos fuertes de MATLAB son los gráficos, que se verán con más detalle en una sección posterior. A título de ejemplo, se puede teclear la siguiente línea y pulsar *intro:*

```
>> x=-4:.01:4; y=sin(x); plot(x,y), grid, title('Función seno(x)').
```

En la Figura 24 se puede observar que se abre una nueva ventana en la que aparece representada la función sin(x). Esta figura tiene un título "Función seno(x)" y una cuadrícula o "grid". En realidad la línea anterior contiene también varias instrucciones separadas por comas o puntos y comas. En la primera se crea un vector x con 801 valores reales entre -4 y 4, separados por una centésima. A continuación se crea un vector Y, cada uno de cuyos elementos es el seno del correspondiente elemento del vector X. Después se dibujan los valores de Y en ordenadas frente a los de X en abscisas. Las dos últimas instrucciones establecen la cuadrícula y el título.

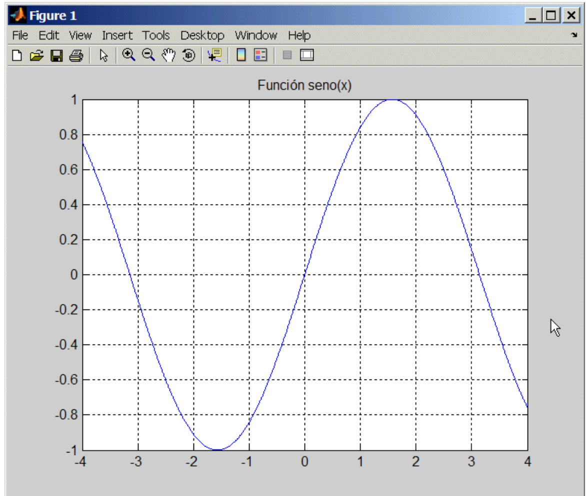


Figura # 2.24: Gráfico de la función *seno(x).*
Fuente: Javier García de Jalón, J. I. (2005). *Aprenda Matlab 7.0 como si estuviera en primero*. Madrid: Escuela Técnica Superior de Ingenieros Industriales Universidad Politécnica de Madrid.

2.15. Uso del *Help*

El autor tomo la explicación textual del tema de**:** Javier García de Jalón, J. I. (2005). MATLAB dispone de un excelente *Help* con el que se puede encontrar la información que se desee. La Figura 2.25 muestra las distintas opciones que aparecen en el menú Help de la ventana principal de la aplicación:

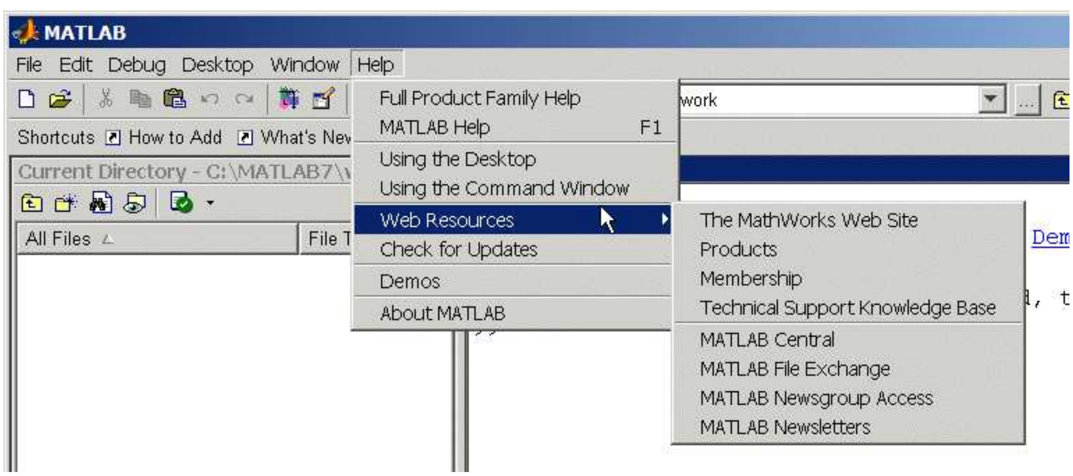


Figura # 2.25: Menú Help de MATLAB.
Fuente: Autores

2.15.1.Full Product Family Help

El autor tomo la explicación textual del tema de**:** Javier García de Jalón, J. I. (2005).

En esta ventana de ayuda, se puede buscar información general sobre MATLAB o sobre otros productos de la familia a los que se tenga acceso. La forma de la ventana de ayuda es típica y común con otros niveles de ayuda. La mayor parte de las páginas de ayuda están en formato HTML.

2.15.2.MATLAB Help

El autor tomo la explicación textual del tema de**:** Javier García de Jalón, J. I. (2005).
En esta ventana, en la que se puede buscar ayuda general sobre MATLAB o sobre la función o el concepto que se desee. La portada de esta ayuda tiene **tres** capítulos principales:

Functions, que contiene información de referencia sobre las funciones por orden alfabético o por categorías; ***Handle Graphics***, que permite acceder a información concreta sobre las distintas propiedades de los objetos gráficos; ***Documentation Set***, que da acceso a versiones completas de los manuales del programa en formato de pantalla fácilmente navegable (con apartados de *Getting Started*, *User Guides*, *Programming Tips* y *Examples in Documentation*), ***Product Demos*** (con una colección de ejemplos programados que se pueden ejecutar y cuyo código se puede examinar para ver cómo están programados), ***What's New*** (con las novedades de esta versión respecto a la anterior), ***Printing the Documentation Set*** (que permite abrir documentos PDF (*Portable Document Format*), que se corresponden con las versiones en papel de los manuales del programa, y que precisan del programa ***Adobe Acrobat Reader 5.0*** o superior.) y un apartado final sobre ***The MathWorks Web Site Resources*** (que permite acceder a una amplísima colección de informaciones adicionales disponibles en la web de la empresa que ha desarrollado MATLAB). En la parte izquierda de la ventana, cuando está seleccionada la pestaña ***Contents***, aparece un índice temático estructurado en forma de árbol que puede ser desplegado y recorrido con gran facilidad. Las restantes pestañas de esta ventana dan acceso a un índice por palabras (***Index***), a un formulario de búsqueda (***Search***) y a la colección de ejemplos ya programados antes citadas (***Demos***).

2.15.3. **Usando el Escritorio***. (Using the Command Window)*

El autor tomo la explicación textual del tema de**:** Javier García de Jalón, J. I. (2005).
Se abre una ventana de ayuda con un formato similar a las de la Figura anterior con información detallada sobre cómo utilizar y configurar el entorno de desarrollo o

Desktop. Las distintas herramientas disponibles se describen sucesivamente. Cada página dispone de flechas y enlaces que permiten ir a la página siguiente o volver a la anterior. Es posible también imprimir aquellas páginas que se desee consultar o archivar sobre papel. Una característica muy importante es la posibilidad de organizar las ventanas con gran flexibilidad, agrupándolas o independizándoles según los propios gustos o deseos.

2.15.4. **Usando el Comando Windows** ***(Using the Command Window).***

El autor tomo la explicación textual del tema de**:** Javier García de Jalón, J. I. (2005).
Esta opción del menú ***Help*** da acceso a la información necesaria para aprovechar las capacidades de la ***Command Window***, que es el corazón de MATLAB.

2.15.5. **Recursos en la WEB** ***(Web Resources)***

El autor autor tomo la explicación textual del tema de**:** Javier García de Jalón, J. I. (2005).

El **¡Error! No se encuentra el origen de la referencia.** Muestra algunas direcciones de Internet con información interesante sobre MATLAB. Todas ellas corresponden a distintas secciones de la web de The Mathworks (la empresa que desarrolla y comercializa MATLAB), cuya página de inicio se muestra en primer lugar.

2.15.6. **Revisando Actualizaciones** ***(Check for Updates)***

El autor tomo la explicación textual del tema de**:** Javier García de Jalón, J. I. (2005).
MATLAB se conecta con The Mathworks y comprueba si has versiones más recientes de los productos instalados. Si se es un usuario registrado, es posible descargar las versiones más actuales.

2.15.7. **Demostraciones** ***(Demos).***

El autor tomo la explicación textual del tema de**:** Javier García de Jalón, J. I. (2005).
Se abre una ventana como la mostrada en la Figura anterior que da acceso a un buen número de ejemplos resueltos con MATLAB, cuyos resultados se presentan gráficamente de diversas formas. Es muy interesante recorrer estos ejemplos para hacerse idea de las posibilidades del programa, tanto en cálculo como en gráficos. Es asimismo muy instructivo analizar los ficheros ***.m** de los ejemplos de características similares a las de la aplicación de se desea desarrollar.

Además, de una forma muy inmediata, es posible también recurrir al ***Help*** desde la línea de comandos de la ***Command Window***. Se aconseja practicar un poco al respecto. Por ejemplo, obsérvese la respuesta a los siguientes usos del comando ***help***:

```
>> help
>> help lang
```

2.16. El entorno de trabajo de *Matlab*

El entorno de trabajo de MATLAB es muy gráfico e intuitivo, similar al de otras aplicaciones profesionales de Windows. (Javier García de Jalón, J. I. 2005).

La interfaz de este programa se compone de diferentes ventanas en las que se puede ejecutar instrucciones o bien obtener información.

- Escritorio o área de trabajo.
- Interfaces gráficas de Usuario. (Manejo y Administración de archivos).
- Barra de Menú
- Barra de herramientas
- Espacio de Trabajo (en esta ventana aparecerán los tamaños de las variables)
- Historia de Comandos.(visualizan todas las instrucciones)
- Ventana de Comandos (Ingresan datos y se recibe resultados)
- Botón de Inicio

2.16.1. Características Generales

- Elementos básicos del escritorio de MatLab (Ejecutar MatLab)
- *Command Windows*: Donde se ejecutan todas las instrucciones y programas. Se escribe la instrucción o el nombre del programa y se da *Enter.*
- *Command History:* Muestra los últimos comandos ejecutados en *Command Windows*. Se puede recuperar el comando haciendo doble click.
- *Current directory*: Situarse en el directorio donde se va a trabajar.
- *Help:* Ayuda de MatLab (también se puede usar desde *comand windows*).
- *Workspace:* Para ver las variables que se están usando y sus dimensiones (si son matrices)

  Editor del MatLab: Todos los ficheros de comandos MatLab deben de llevar la extensión .m ,tal como se muestra en la figura 2.26

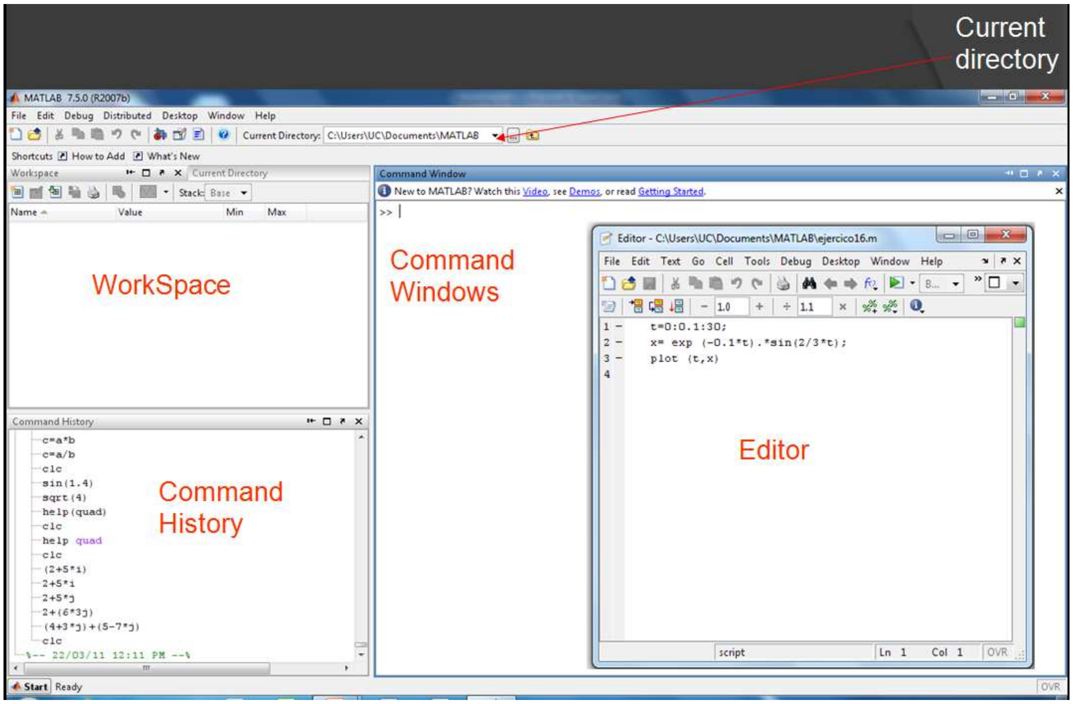


Figura # 2.26: Características generales de Matlab
Fuente: Autores

A continuación se describen brevemente estas componentes. Téngase en cuenta que utilizar MATLAB y desarrollar programas para MATLAB es mucho más fácil si se conoce bien este entorno de trabajo. Para alcanzar la máxima productividad personal en el uso de esta aplicación es por ello muy importante leer con atención las secciones que siguen.

2.16.2. El escritorio de Matlab *(Matlab desktop)*

El autor tomo la explicación textual del tema de**:** Javier García de Jalón, J. I. (2005).
El Matlab Desktop es la ventana más general de la aplicación. El resto de las ventanas o componentes citadas pueden alojarse en la Matlab Desktop o ejecutarse como ventanas independientes. A su vez, los componentes alojados en el Matlab Desktop pueden aparecer como sub-ventanas independientes o como pestañas dentro de una de las sub-ventanas. MATLAB ofrece una gran flexibilidad al respecto y es cada usuario quien decide en qué forma desea utilizar la aplicación.

Cuando se arranca MATLAB por primera vez o cuando se ejecuta el comando View/Desktop Layout/ Default aparece una ventana dividida en tres zonas, en realidad aparecen cuatro componentes, pues la sub-ventana superior izquierda contiene dos componentes superpuestas que se permutan por medio de la pestaña correspondiente.

Se muestra un detalle del menú Desktop, desde el que se controlan las componentes visibles y la forma en que se visualizan. Por ejemplo, la ventana activa es la Command Window, en el menú de la Figura2.27 aparece la opción de dejar de alojar dicha ventana en el Matlab Desktop (Undock Command Window). Dicho menú permite también eliminar del Desktop alguna de las componentes visibles o visualizar el Help (que no está visible). Con los submenús de Desktop Layout se pueden adoptar algunas configuraciones predefinidas, como la configuración por defecto (Default)(figura2.28) o incluir sólo la Command Window. La configuración adoptada por el usuario se mantendrá la siguiente vez que arranque el programa. Es posible también guardar distintas configuraciones con distintos nombres, para su uso posterior.

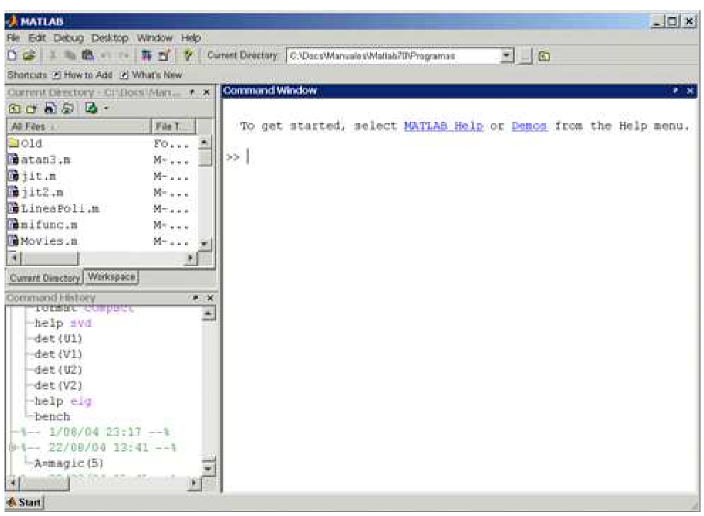


Figura # 2.27: Configuración por defecto del Matlab Desktop.
Fuente: Autores

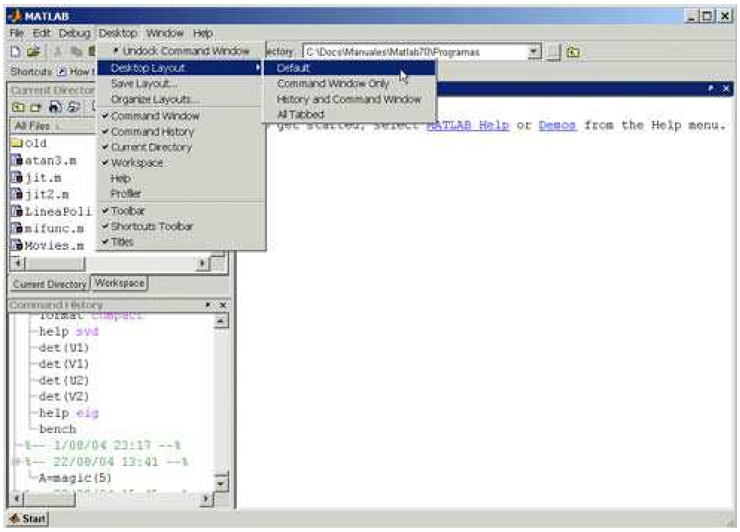


Figura # 2.28: Menú para configurar el Matlab Desktop.
Fuente: Autores

2.16.3. **Ventada de Instrucciones** ***(Command Window)***

El autor tomo la explicación textual del tema de**:** Javier García de Jalón, J. I. (2005).
Ésta es la ventana en la que se ejecutan interactivamente las instrucciones de MATLAB y en donde se muestran los resultados correspondientes, si es el caso. En cierta forma es la ventana más importante y la única que existía en las primeras versiones de la aplicación. En esta nueva versión se han añadido algunas mejoras significativas, como las siguientes:

1. Se permiten líneas de comandos muy largas que automáticamente siguen en la línea siguiente al llegar al margen derecho de la ventana. Para ello hay que activar la opción Wrap Lines, en el menú File/Preferences/Command Window.

2. Clicando con el botón derecho sobre el nombre de una función que aparezca en esta ventana se tiene acceso a la página del Help sobre dicha función. Si el código fuente (fichero *.m) está disponible, también se puede acceder al fichero correspondiente por medio del Editor/ Debugger.

3. Comenzando a teclear el nombre de una función y pulsando la tecla Tab, MATLAB completa automáticamente el nombre de la función, o bien muestra en la línea siguiente todas las funciones disponibles que comienzan con las letras tecleadas por el usuario.

4 Cuando al ejecutar un fichero *.m se produce un error y se obtiene el correspondiente mensaje en la Command Window, MATLAB muestra mediante un subrayado un enlace a la línea del fichero fuente en la que se ha producido el error. Clicando en ese enlace se va a la línea correspondiente del fichero por medio del Editor/Debugger.

2.16.4. Búsqueda de Historial de comando *(Command History Browser)*

El autor tomo la explicación textual del tema de**:** Javier García de Jalón, J. I. (2005).
La ventana ***Command History*** ofrece acceso a las sentencias que se han ejecutado anteriormente en la ***Command Window***. Estas sentencias están también accesibles por medio de las teclas ↑ y ↓ como en las versiones anteriores, pero esta ventana facilita mucho el tener una visión más general de lo hecho anteriormente y seleccionar lo que realmente se desea repetir.

Las sentencias ejecutadas anteriormente se pueden volver a ejecutar mediante un doble clic o por medio del menú contextual que se abre al clicar sobre ellas con el botón derecho. También se pue den copiar y volcar sobre la línea de comandos, pero se ha de copiar toda la línea, sin que se admita la copia de un fragmento de la sentencia. Existen opciones para borrar algunas o todas las líneas de esta ventana. Se puede también hacer un ***profile*** (evaluar la eficiencia relativa) de una sentencia o de un grupo de sentencias.

2.16.5. Directorio activo o Directorio actual *(Current Directory Browser)*

El autor tomo la explicación textual del tema de**:** Javier García de Jalón, J. I. (2005).
El concepto de ***directorio activo*** o ***directorio actual*** es muy importante en MATLAB. Los programas de MATLAB se encuentran en ficheros con la extensión ****.m***. Estos ficheros se ejecutan tecleando su nombre en la línea de comandos (sin la extensión), seguido de los argumentos entre paréntesis, si se trata de funciones. No todos los ficheros ****.m*** que se encuentren en el disco duro o en otras unidades lógicas montadas en una red local son accesibles sin más. Para que un fichero ****.m*** se pueda ejecutar es necesario que se cumpla una de las dos condiciones siguientes:

1. Que esté en el ***directorio actual***. MATLAB mantiene en todo momento un único directorio con esta condición. Este directorio es el primer sitio en el que

MATLAB busca cuando desde la línea de comandos se le pide que ejecute un fichero.

2. Que esté en uno de los directorios indicados en el ***Path*** de MATLAB. El ***Path*** es una lista ordenada de directorios en los que el programa busca los ficheros o las funciones que ha de ejecutar. Muchos de los directorios del ***Path*** son propios de MATLAB, pero los usuarios también pueden añadir sus propios directorios, normalmente al principio o al final de la lista. En un próximo apartado se verá cómo se controla el ***Path***.

El comando ***pwd*** (de *print working directory*) permite saber cuál es el ***directorio actual***. Para cambiar de ***directorio actual*** se puede utilizar el comando ***cd*** (de *change directory*) en la línea de comandos, seguido del nombre del directorio, para el cual se puede utilizar un ***path*** absoluto (por ejemplo ***cd C:\Matlab\Ejemplos***) o relativo (***cd Ejemplos***). Para subir un nivel en la jerarquía de directorios se utiliza el comando ***cd ..***, y ***cd ../..*** para subir dos niveles. Éste es el mismo sistema que se sigue para cambiar de directorio en las ventanas de MS-DOS. MATLAB permite utilizar la barra normal (/) y la barra invertida (\), indistintamente.

La ventana ***Current Directory*** permite explorar los directorios del ordenador en forma análoga a la del ***Explorador*** u otras aplicaciones de ***Windows***. Cuando se llega al directorio deseado se muestran los ficheros y ficheros allí contenidos.
La ventana ***Current Directory*** permite ordenarlos por fecha, tamaño, nombre, etc. El directorio actual cambia automáticamente en función del directorio seleccionado con este explorador, y también se puede cambiar desde la propia barra de herramientas del ***Matlab Desktop***. Los ficheros ****.m*** mostrados en la ventana ***Current Directory*** se pueden abrir con el ***Editor/Debugger*** mediante un doble clic.
A partir del menú contextual que se abre clicando con el botón derecho en cualquier parte de la ventana ***Current Directory*** se tiene la posibilidad de añadir ese directorio al ***Path*** de MATLAB.

2.17. Elementos básicos de Matlab

Los elementos básicos del Matlab, como cualquier otro lenguaje de programación, son: constantes, variables, operaciones, expresiones y funciones.

2.17.1. Constante numérica

a) Números enteros: 2 35 -48

b) Números reales: 2. -35.2 48.45

    1. Máximo de 16 cifras significativas
    2. Utilizando la letra "e" a continuación de un nº con punto decimal [2.2250e-308 1.7e+308].

c) Números complejos: 2+3i 4*j i,j=(-1)½

2.17.2. Operaciones aritméticas elementales

Suma: +	Multiplicación: *	Exponenciación: ^
Resta: -	División: /	

Primero exponenciaciones, luego divisiones y multiplicaciones por último sumas y restas.

2.17.3. Variables

Es la etiqueta que identifica una porción de memoria; Matlab diferencia entre mayúsculas y minúsculas. Para ver las variables definidas en un instante determinado se teclea:

```
>> who
```

o bien

```
>> whos
```

Para eliminar alguna variable se ejecuta

```
>> clear variable1 variable2
```

2.17.4. Expresiones numéricas

Son un conjunto de números, funciones y variables previamente definidas, relacionados todos ellos por operadores aritméticos. Si una expresión es demasiado larga se indica mediante formatos.

2.17.5. Formatos

Por defecto matlab tiene formato corto pero se puede elegir entre los siguientes formatos.

>> format long (14 cifras significativas)

>> format short (5 cifras significativas)

>> format short e (notación exponencial)

>> format long e (notación exponencial)

>> format rat (aproximación racional)

2.17.6. Variables predefinidas en *Matlab*

i = (-1)½ pi = π Inf= ∞ NaN= cálculos indefinidos

eps = < nº que + otro nº=nº coma flotante 2.22e-16

date = valor de la fecha actual

rand = genera números aleatorios [0 1]

realmin = <nº+ realmax= >nº+

2.17.7. Funciones de *matlab*

- Nombre (argumento)

- sqrt (x) raiz cuadrada
- abs(x) módulo de x
- conj(z) conjugado de un complejo
- real(z), imag(z) parte real e imaginaria de z respectivamente
- exp(x) calcula ex, siendo x real o complejo
- sin(x) asin(x) [-π/2 π/2] cos(x) acos(x) [0 π] tan(x)
- atan(x) [-π/2 π/2] angle(z) log(x) (en base e) log10(x)
- rats(x) rem(x,y) resto de x/y round(x) sign(x)

2.17.8. Comandos de ayuda

- help
- lookfor
- what ficheros .m y .mat del directorio actual
- dir ficheros del directorio actual
- type nombre_fichero Muestra el contenido del fichero
- delete nombre_fichero Borra el fichero
- cd cambia de directorio
- pwd indica el directorio actual

- which nombre_fichero india el directorio donde esta
- ! Abre una ventana de MSDOS que se cierra cuando volvemos a Matlab
- startup.m fichero de arranque al ejecutar matlab. Para guardar en un fichero los comandos que se ejecutan en una sesión se pone:
- >> diary nombre_fichero
- ...
- >> diary off
- >> diary tema1.dia
- >> clear

2.18. Gráficos: 2D Y 3D

Funciones gráficas elementales:

MATLAB dispone de 4 funciones básicas para crear gráficos. Estas se diferencian principalmente por el *tipo de escala* que utilizan en los ejes Estas cuatro funciones son las siguientes:

- plot() crea un gráfico a partir de vectores y/o columnas de matrices, con escalas lineales sobre ambos ejes.
- loglog() ídem con escala logarítmica en ambos ejes.
- semilogx() ídem con escala lineal en el eje de ordenadas y logarítmica en el eje de abscisas.
- semilogy() ídem con escala lineal en el eje de abscisas y logarítmica en el eje de ordenadas.

Existen funciones orientadas a añadir títulos al gráfico, a los ejes, a dibujar una cuadrícula auxiliar, a introducir texto, etc.

- title('título') añade un título al dibujo
- xlabel('tal') añade una etiqueta al eje de abscisas. Con ***xlabel off*** desaparece
- ylabel('cual') idem al eje de ordenadas. Con ***ylabel off*** desaparece
- text(x,y,'texto') introduce 'texto' en el lugar especificado por las coordenadas **x** e y. Si **x** e **y** son vectores, el texto se repite por cada par de elementos.
- gtext('texto') introduce **texto** con ayuda del ratón: legend() define rótulos para las distintas líneas o ejes utilizados en la figura.

- grid activa una cuadrícula en el dibujo.
- Con ***grid off*** desaparece la cuadrícula

*Plo*t es la función clave de todos los gráficos en MATLAB. Ya se ha dicho que el elemento básico de los gráficos bidimensionales es el **vecto**r. Se utilizan también cadenas de 1, 2 ó 3 caracteres para indicar *colores* y *tipos de líne*a. La función ***plot()***, no hace otra cosa que dibujar vectores (figura2.29) . Ejemplo1:

- **» x=[1 3 2 4 5 3]**
- x =
- 1 3 2 4 5 3
- **» plot(x)**

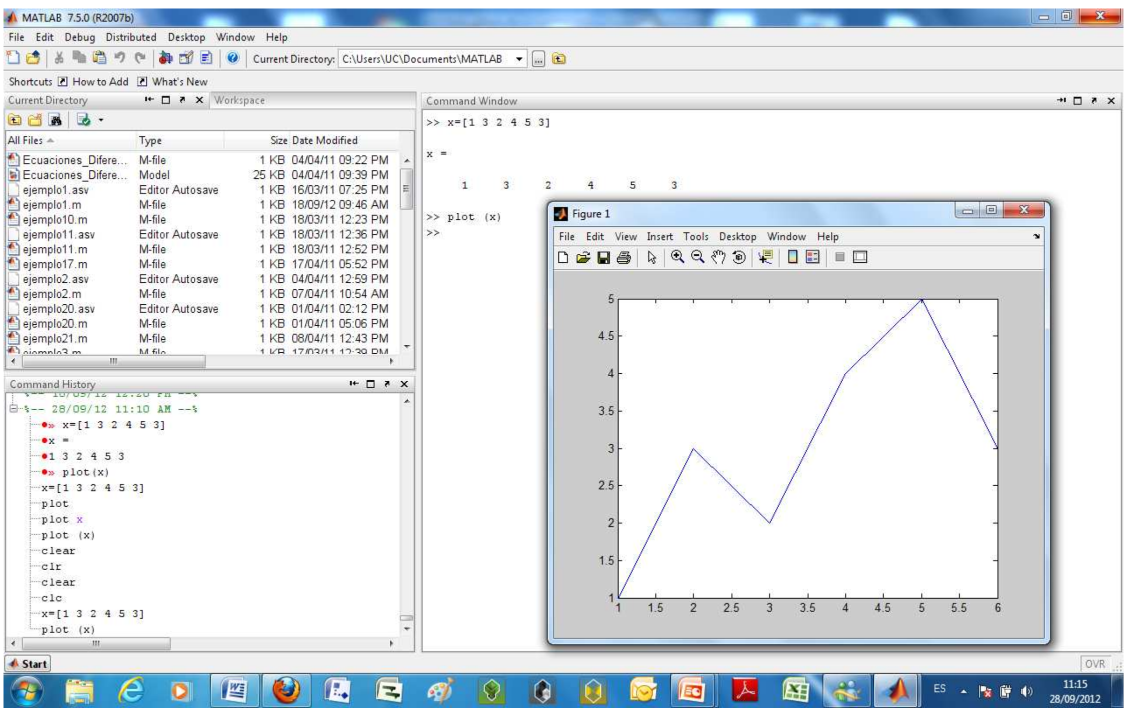


Figura # 2.29: Gráfico de un vector en Matlab
Fuente: Autores

2.19. Transformadas de Laplace en Matlab

La transformada de Laplace es un método que transforma una ecuación diferencial en una ecuación algebraica más fácil de resolver. El matemático francés P.S. de Laplace (1749-1827) descubrió una forma de resolver ecuaciones diferenciales: *Multiplicar cada término de la ecuación por* e^{-st} *y, así, integrar cada uno de esos términos* *respecto del tiempo desde cero hasta infinito; s* es una constante con unidades de $\frac{1}{tiempo}$. El resultado es lo que hoy día se conoce como la *transformada de Laplace* y definida de la forma:

$$F(s) = \int_0^{\infty} f(t) e^{-st} dt$$

MATLAB, resuelve la transformada de Laplace, mediante el comando: **laplace**

Sintaxis: laplace(F)
laplace(F,t)

donde t es el símbolo de la variable en *f* que viene determinada por el comando **syms**

Ejemplos.-

1.- Calcular la transformada de la función $f(t) = t^4$, mediante MATLAB.

Procedimiento: Introduzca los siguientes comandos:

$$>> syms \quad t$$
$$>> f = t.\wedge 4;$$
$$>> laplace(f) \hookleftarrow$$

Nos devuelve:

$$24/s \wedge 5$$

Escribiendo la el comando:

$$>> pretty(ans) \hookleftarrow$$

Nos devuelve el resultado en formato matemático normal: $\frac{24}{s^5}$ tal como se muestra en la figura 2.30

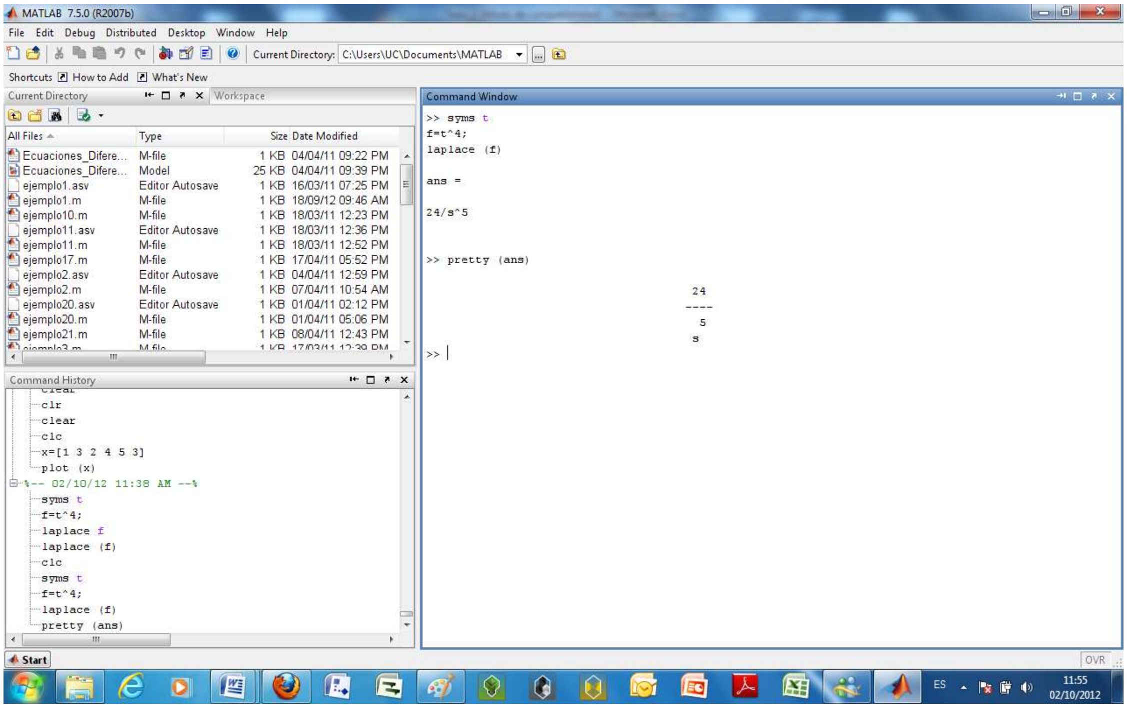


Figura # 2.30 : Ejemplo realizado en el programa matlab
Fuente: Autores

2.20. Transformada inversa de Laplace en Matlab

El comando **ilaplace**, calcula la transformada inversa de Laplace.

Ejemplo.- Calcular la transformada inversa de $f = 1/s \wedge 2$

Procedimiento.- Introduzca los siguientes comandos:

$$>> syms \quad s$$
$$>> f = 1/s.\wedge 2;$$
$$>> ilaplace(f) \hookleftarrow$$

Nos devuelve:

t , tal como se muestra en la figura 2.31

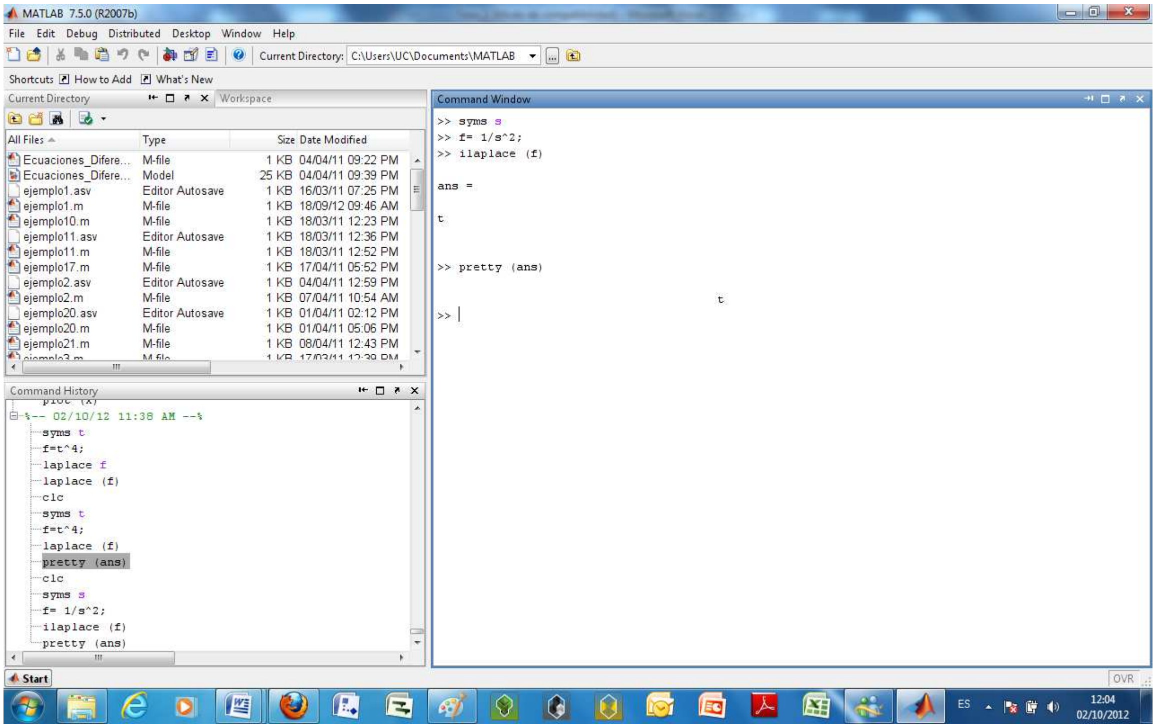


Figura # 2.31: Ejemplo realizado en el programa matlab
Fuente: Autores

2.21. El programa Simulink

El software Simulink es una herramienta que se utiliza por un gran número de colectivos para realizar tareas de simulación de modelos y controladores avanzados.

El programa Simulink presenta ventajas frente a otros programas matemáticos que podrían ser también utilizados para resolver las ecuaciones de los sistemas, tales como un entorno interactivo y un conjunto de librerías con bloques personalizables que permiten simular, implementar y probar una serie de sistemas variables con el tiempo.

Además Simulink está integrado en Matlab y por ello es posible tener acceso a una amplia gama de herramientas que permiten desarrollar algoritmos, analizar y visualizar simulaciones.

2.22. Entorno gráfico

El autor tomo la explicación textual del tema de**:** Alicia Arce Rubio, G. V. (2009)

Para poder utilizar el programa Simulink correctamente es importante estar familiarizado con las ventanas y las herramientas. Al programa Simulink se accede a través del programa Matlab.

Una vez que se inicia una sesión en MATLAB, SIMULINK se selecciona con el comando:

>> simulink↵

Esto abre la *ventana de control del* SIMULINK con sus iconos y menús de persiana (pull-down) en la barra de encabezado., observe la figura #

Para iniciar hacer clic en **file** luego en **new** del menú de persiana. Con ello abre una ventana en la que el sistema se puede formar llamado *untitled*., Observe la figura2.32

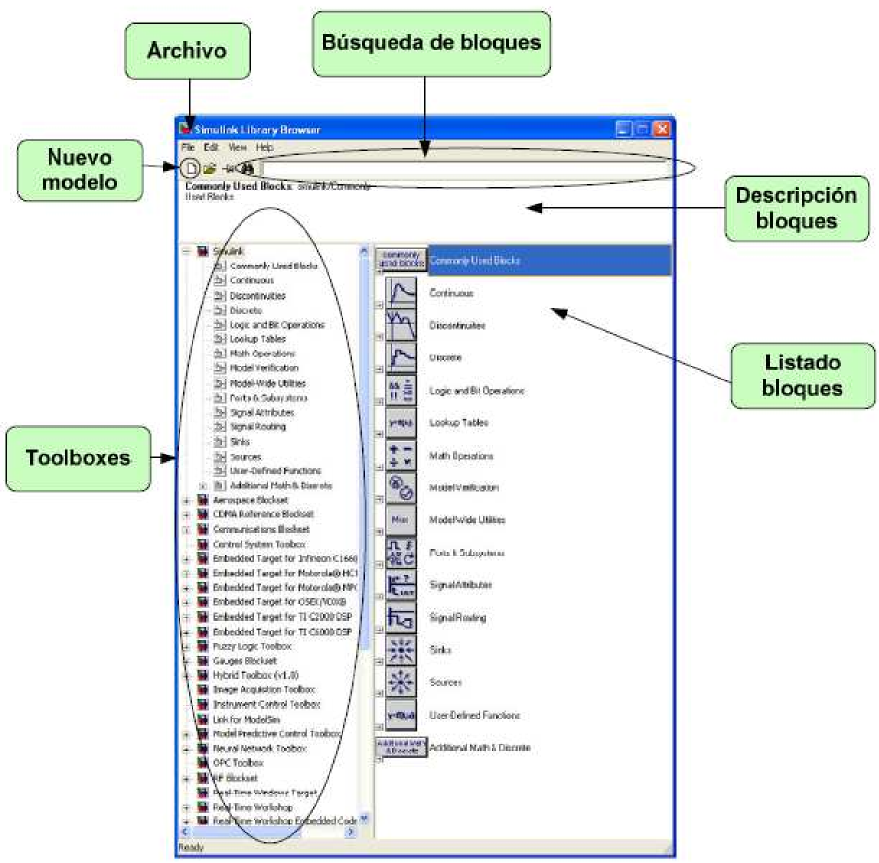


Figura # 2.32: Ventana de la librería de Simulink
Fuente: Alicia Arce Rubio, G. V. (2009).

La figura 2.33 muestra la ventana principal de Simulink. En esta ventana se encuentra el icono marcado como '*Librería de Simulink*'. Haciendo clic en esta opción se abre la librería (figura 2.32) donde se encuentran los bloques que permiten crear cualquier tipo de modelo o controlador que se desee simular. También son importantes las opciones marcadas como '*Activar simulación*', '*Parar simulación*' y '*Tiempo de simulación*' que permiten comenzar a simular cualquier archivo Simulink creado en esa misma ventana, parar esa simulación en cualquier momento o bien cambiar el tiempo que se desea simular correspondientemente.

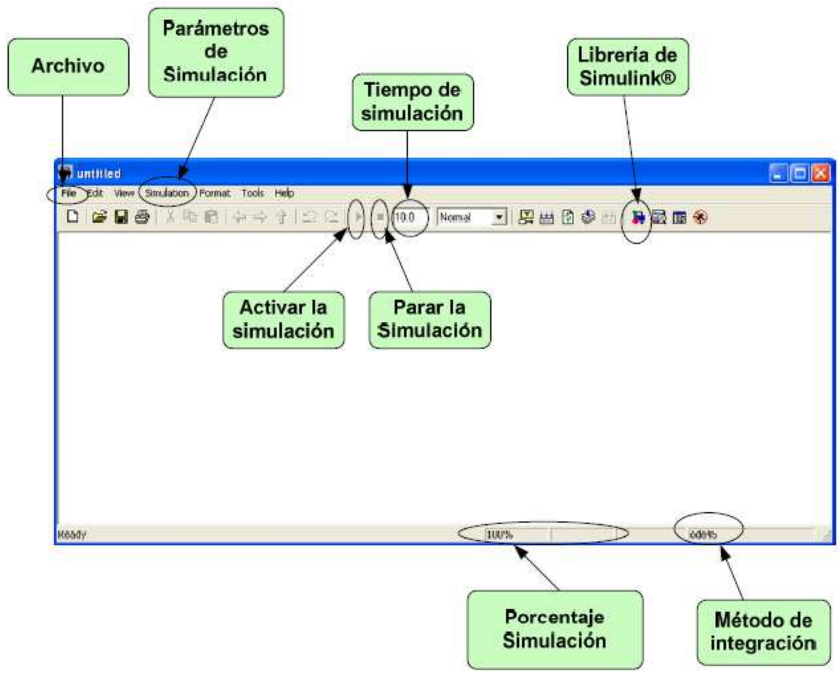


Figura # 2.33: Ventana Principal de Simulink
Fuente: Alicia Arce Rubio, G. V. (2009).

Debe ser aclarado que simular un sistema significa resolver unas ecuaciones que describen un sistema real o un controlador durante un periodo de tiempo, por lo que el tiempo de simulación es el periodo total de tiempo para el cual el programa resuelve las ecuaciones, siendo por lo tanto un parámetro que debe ser elegido adecuadamente para poder analizar y representar correctamente los resultados simulados. Por otro lado la opción marcada como '*Parámetros de simulación*' abre una ventana más completa donde es posible configurar parámetros más avanzados y mejorar los resultados de las simulaciones.

2.23. Bloques Principales

El autor tomo la explicación textual del tema de**:** Alicia Arce Rubio, G. V. (2009)
A continuación se presentarán los bloques más utilizados en Simulink y necesarios para la asignatura de control. En la figura 2.34 se ilustran cada uno de los bloques. Cada uno de estos bloques podrán ser configurados por el alumno dependiendo de las exigencias del problema que se pretenda resolver mediante la ventana de dialogo de configuración de cada bloque. A esta ventana se accede haciendo doble clic sobre el bloque seleccionado.

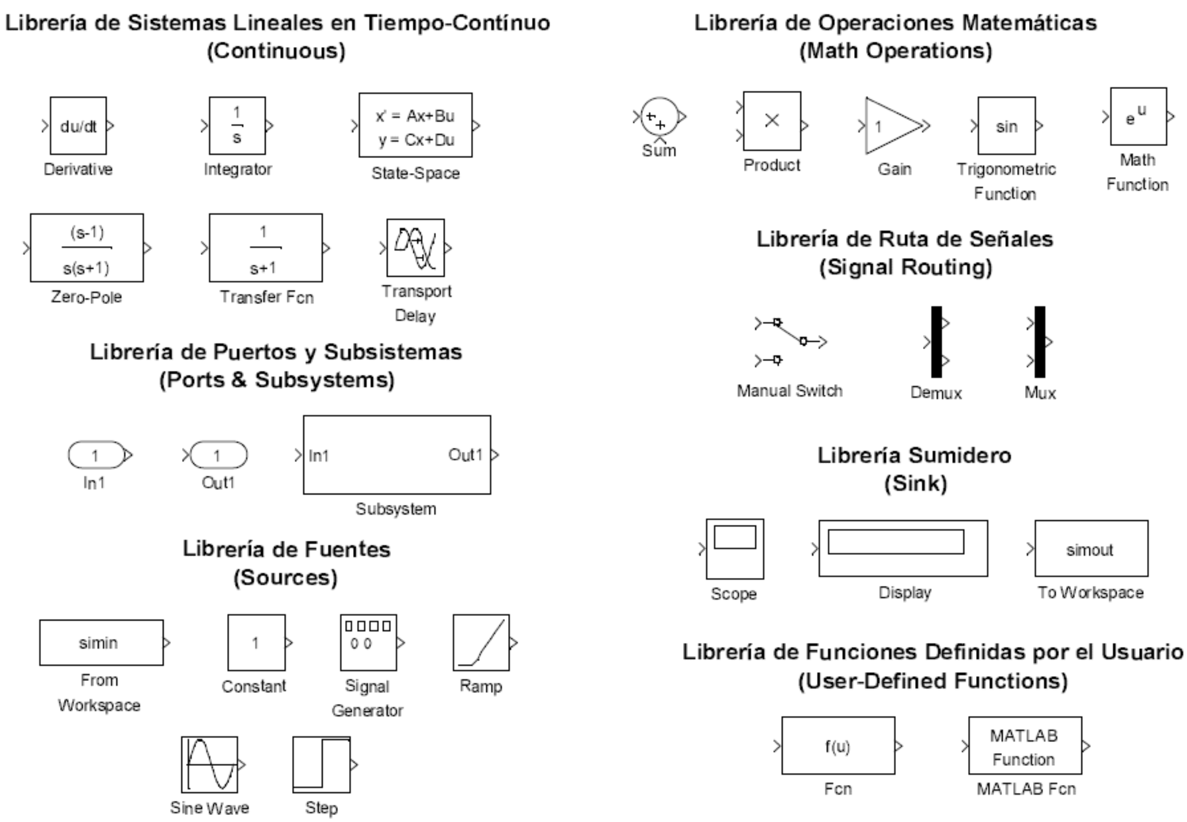


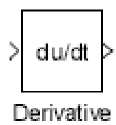
Figura # 2.34: Bloques principales
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.1. Librería de Sistemas Lineales en Tiempo-Continuo

El autor tomo la explicación textual del tema de**:** Alicia Arce Rubio, G. V. (2009)

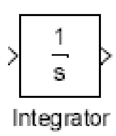
En este apartado se describirán los bloques para la implementación de sistemas lineales en tiempo-continuo comúnmente utilizados en la asignatura.

2.23.1.1. Bloque Derivative



El bloque *'Derivative'* aproxima la derivada de su entrada considerando los valores iniciales de la salida igual a 0. La exactitud de los resultados depende del tamaño del periodo de muestreo utilizado en la simulación. Pequeños pasos de muestreo permiten obtener una curva de la salida más suave y exacta.

2.23.1.2. Bloque Integrador



El Bloque '*Integrator* ' integra su entrada. Los resultados de la integración van a depender del método de integración que se seleccione en el menú 'Configuration Parameters' al que se accede mediante la opción marcada como '*Parámetros de simulación*'. El programa Simulink trata el bloque integrador como un sistema dinámico con un estado, su salida. La entrada de este bloque es la derivada en el tiempo del estado. El algoritmo de integración numérica seleccionado calcula la salida del bloque integrador en el periodo de muestreo actual usando el valor de entrada actual y del paso anterior. El bloque también provee una opción de condición inicial que permite configurar el estado inicial.

La ventana de dialogo de parámetros para el bloque 'Integrator ' es presentada en la Figura 2.35

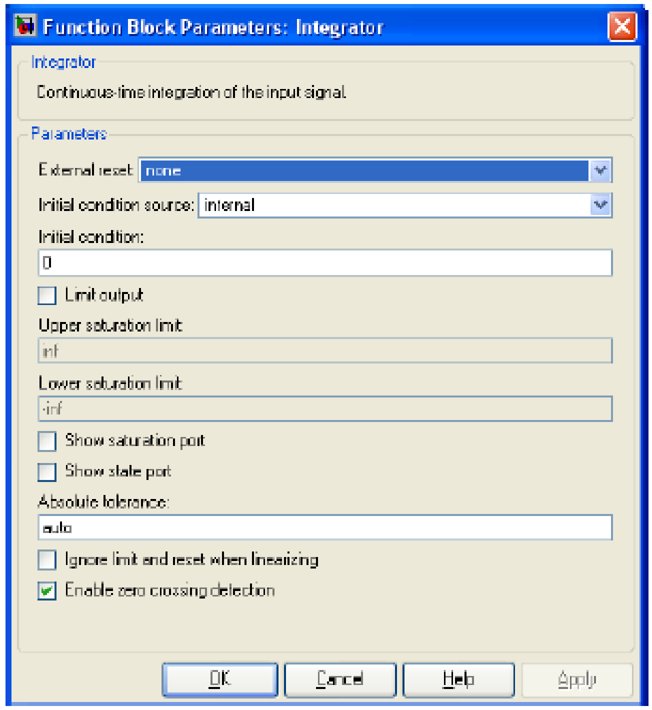
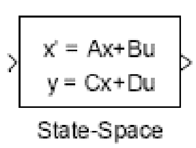


Figura # 2.35: Parámetros del bloque integrador
Fuente: Alicia Arce Rubio, G. V. (2009).

La ventana de dialogo *'Function Block Parameter '* permite especificar valores para la condición inicial. Esta ventana también permite especificar límites inferiores y superiores para la integración. El resto de parámetros y configuraciones de este bloque no serán necesarios para esta asignatura.

Para determinar los limites inferiores y superiores se debe seleccionar la opción *'Limit output '* y asignar valores en los campos: *'Upper saturation limit ' y 'Lower saturation limit '.*

2.23.1.3. Bloque State-Space



El bloque '*State-Space*' implementa un sistema definido a través de ecuaciones en el espacio de estados. (Alicia Arce Rubio, G. V. 2009).

$$x = Ax+Bu$$

$$y = Cx + Du$$

Donde x y u son vectores columna, la matriz A debe ser definida cuadrada $n \times n$, con n siendo el número de estados del sistema, la matriz B debe de ser definida con dimensión $n \times m$, con m siendo el número de entradas, la matriz C es definida con dimensión $r \times n$, donde r representa el número de salidas, y la matriz D posee dimensión $r \times m$.

La ventana de dialogo de configuración de parámetros para el bloque '*State-Space*' se presenta en la Figura 2.36.

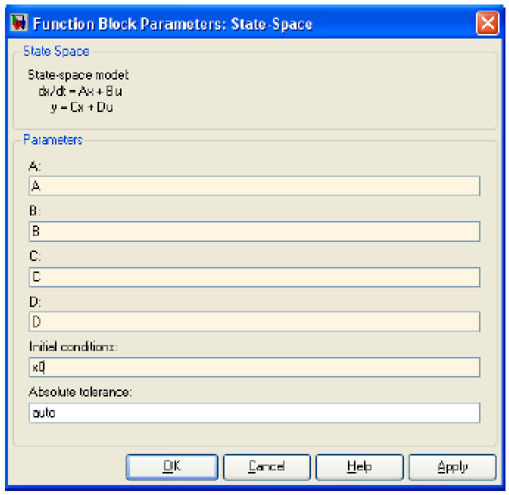
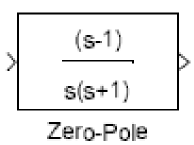


Figura # 2.36: Parámetros del bloque State Space
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.1.4. Bloque Zero Pole



El bloque '*Zero-Pole*' implementa un sistema con ceros, polos y ganancia especificada en el dominio-*s*. Este bloque representa la función de transferencia particularizada con los parámetros especificados para un sistema dado tal como se muestra en la figura 2.37

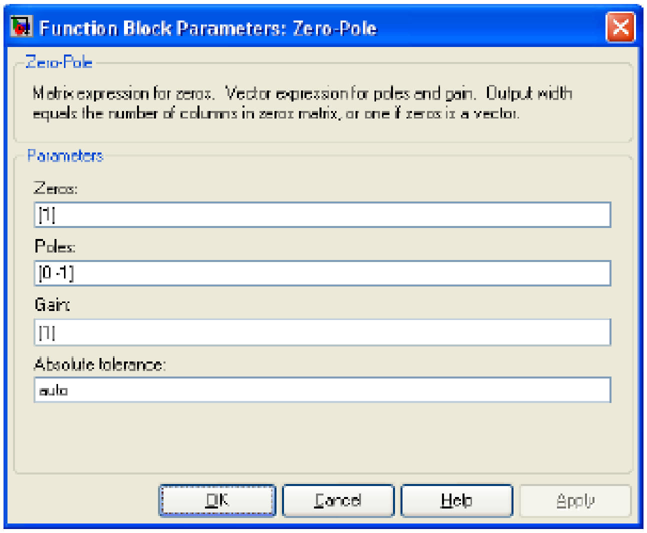


Figura # 2.37: Parámetros del bloque Zero Pole
Fuente: Alicia Arce Rubio, G. V. (2009).

Los parámetros del sistema pueden ser especificados en la ventana de configuración como una expresión o como un vector. El bloque modificará su apariencia dependiendo de la especificación de los parámetros. Por ejemplo, si en la ventana de dialogo ‘Function Block Parameters’ se especifica ‘Zeros’ como [−2 −4 −6 −8], ‘Poles’ como [−1 − 3 −5 −7 −9], y ‘Gain’ como 25, el bloque se presentará como se muestra en la figura 3.6. Si cada parámetro es especificado como una variable simbólica, por ejemplo, ceros, polos, Ganancia, a la que se le asigna unos valores en la ventana de comando del, tal como se muestra en la figura 2.38

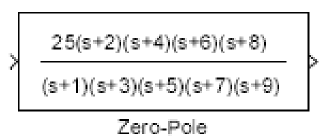


Figura # 2.38: Bloque Zero Pole especificado a través de vectores
Fuente: Alicia Arce Rubio, G. V. (2009).

En la ventana de comando del programa MatLab: **ceros=[-2 -4 -6 -8]; polos=[- 1 -3 -5 -7 -9]; Ganancia=25**, el bloque representará la función de transferencia a través de las variables simbólicas seguidas por (s), como se muestra en la siguiente figura2.39.

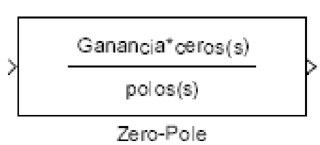
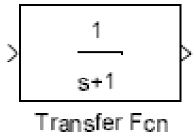


Figura # 2.39: Bloque Zero Pole especificado a través de variables
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.1.5. Bloque Transfer Fcn



El bloque '*Transfer Fcn*' implementa una función de transferencia con la entrada *U*(*s*) y la salida *Y*(*s*), como se muestra a continuación:

$$G(s) = \frac{U(s)}{Y(s)} = \frac{num(s)}{dens}$$

Asumiendo un sistema de primera orden con un polo en $s = -10$ y un zero en $s = -2$, modelado por la función de transferencia:

$$G(s) = \frac{U(s)}{Y(s)} = \frac{s+2}{s+10}$$

Este modelo se programa utilizando el bloque '*Transfer Fcn*', a través de la ventana de dialogo presentada en la figura 2.40, donde el numerador es [1 2] y el denominador es [1 10].

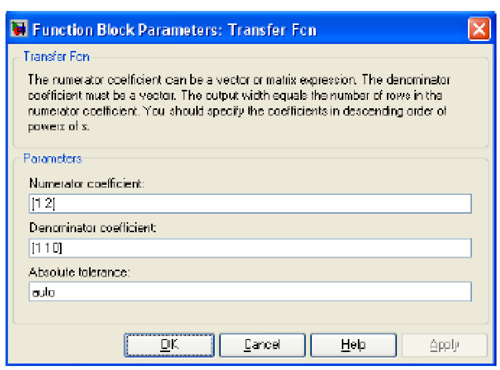


Figura # 2.40: Parámetros del Bloque Transfer FCN
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.1.6. Bloque Transport Delay

El bloque '*Transport Delay*' retrasa la entrada del bloque por un determinado periodo de tiempo. Este bloque puede ser usado para simular retrasos de tiempo. En la ventana de configuración, el bloque permite seleccionar un valor inicial '*Initial output* ' que se mantendrá constante hasta que el tiempo de la simulación exceda el valor del retraso '*Time delay*'. El parámetro '*Time delay*' no debe de ser negativo. El tiempo de muestreo debe ser seleccionado adecuadamente de manera que el tiempo de retraso sea mayor que ese valor. Tal como se muetra en la figura 2.41

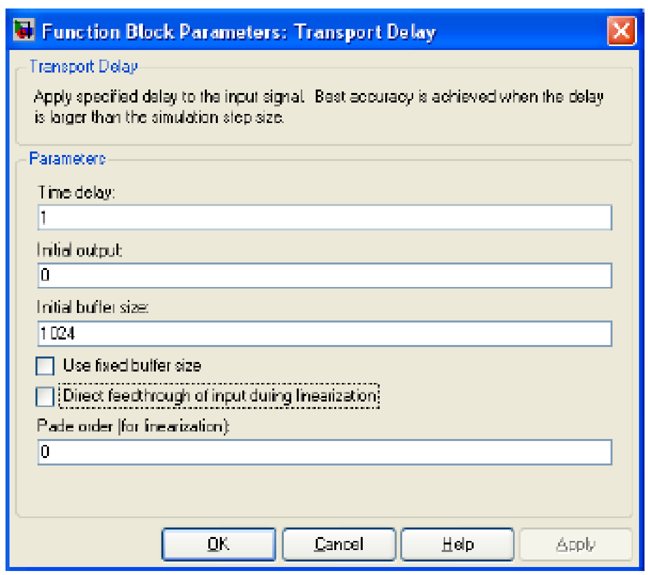


Figura # 2.41: Parámetros del bloque *Transport Delay*.
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.2. Librería de Puertos y Subsistemas (Ports & Subsys-tems)

El autor tomo la explicación textual del tema de**:** Alicia Arce Rubio, G. V. (2009)
En este apartado se detallarán únicamente los bloques de esta librería presentados en la figura 2.42 Los bloques de puerto de entrada y puerto de salida (del inglés *Inport Block* y *Outport Block*) son puertos que sirven para conectar un sistema externo con un subsistema (interno). El bloque Subsistema (del inglés *Subsystem*) representa un

subsistema de un sistema. Cuando el modelo o el sistema de control aumenta en tamaño y complejidad se puede simplificar agrupando bloques en subsistemas.

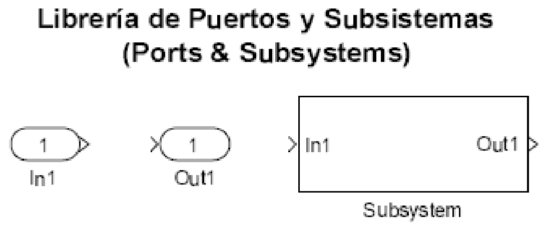


Figura # 2.42: Bloque de la librería Ports & Subsystems
Fuente: Alicia Arce Rubio, G. V. (2009).

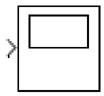
Para crear un subsistema, se inserta el bloque '*Subsystem*' en la ventana Simulink .Este bloque abre una ventana (doble clic sobre el bloque) en la que se debe programar los bloques pertenecientes al subsistema. Si se desea crear un subsistema directamente de un grupo de bloques que se hayan programado en la ventana principal de Simulinkr, se debe seleccionar los bloques deseados y pulsar el botón derecho del ratón sobre estos eligiendo la opción '*Create Subsystem*' del menú.

2.23.3. Librería de Sumidero (Sink)

El autor tomo la explicación textual del tema de**:** Alicia Arce Rubio, G. V. (2009)
Este apartado describirá los bloques más utilizados de la librería '*Sink*'. Los bloques detallados son presentados en la figura 2.43.

Figura # 2.43: Librería Sink
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.3.1. Bloque Scope



El bloque '*Scope*' representa gráficamente la entrada conectada a este bloque con respecto al tiempo de simulación. Este bloque permite representar varias variables a la vez para el mismo periodo de tiempo. El '*Scope*' permite ajustar el tiempo y el rango de los valores de entrada presentados. Se puede mover y redefinir el tamaño de la ventana '*Scope*' y se puede modificar los valores de sus parámetros durante la simulación. Si la señal de entrada al bloque '*Scope*' está formada por varias variables (en lugar de un vector es una matriz), éste asigna colores a cada elemento de la señal en el siguiente orden: amarillo, magenta, cían, rojo, verde y azul oscuro. Cuando la señal posee más de seis elementos, se repite el orden de los colores. Se pueden ajustar los limites del eje-*y* pulsando el botón derecho sobre la gráfica y seleccionando la opción '*Axis Properties*'.

La ventana del bloque 'Scope' posee varios iconos en la barra de herramienta que permiten realizar 'zoom' en la gráfica, preservar las configuraciones de los ejes para la simulación siguiente, limitar los datos presentados y guardar los datos en el espacio de trabajo.

Entretanto, el icono con mayor utilidad es el denominado 'Parameters'. Si se pulsa este botón, la primera pestaña que aparece es la de los parámetros generales, mostrado en la figura 2. 44. En esta pestaña un parámetro importante es el 'Number of axes' que

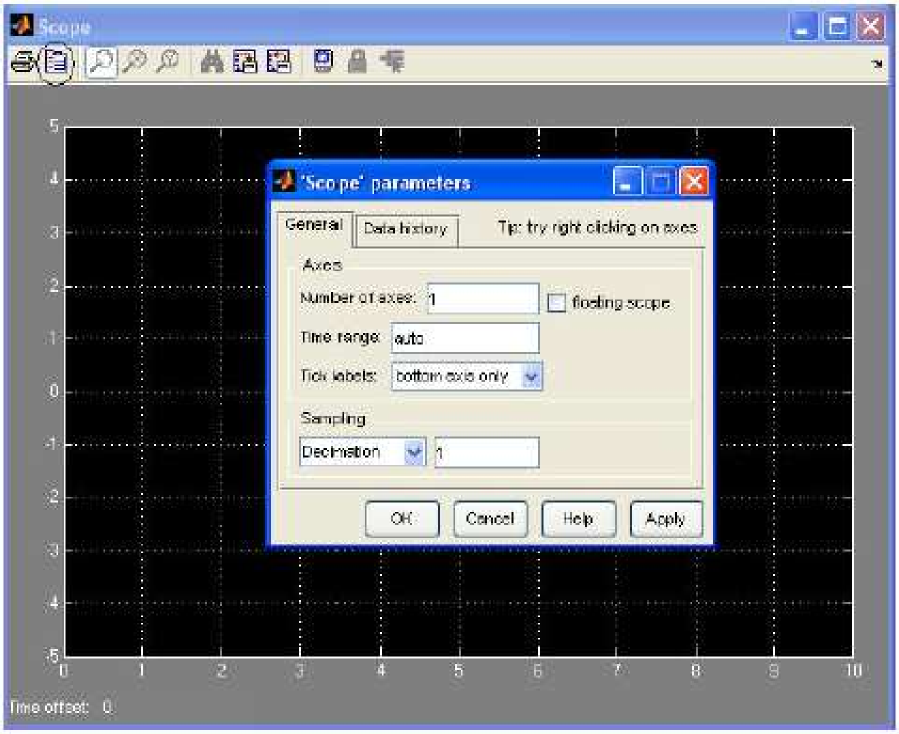


Figura # 2.44: Icono *Parameters*.
Fuente: Alicia Arce Rubio, G. V. (2009).

permite seleccionar el número de ejes que se desean representar en la gráfica del bloque '*Scope*'. En la figura 2.45 se presentan los parámetros de la pestaña '*Data history*'. El parámetro '*Limit data points to last* ' permite especificar cuantos puntos serán representados durante la simulación. Por ejemplo, si la simulación posee un periodo de muestreo muy pequeño, durante la simulación se generará un número muy grande de puntos, lo que implica que está opción debe de tener un número muy elevado para poder visualizar toda la simulación correctamente. Si esta opción aparece como no seleccionada, el bloque '*Scope*' por defecto representará todos los puntos generados.

A través de este cuadro de diálogo es posible también guardar los datos de las variables representadas en el espacio de trabajo del programa Matlab. En la opción '*Variable name*' se define el nombre de la variable y en '*Format* ' se configura el formato con él se guardarán los datos: '*Structure with time*', '*Structure*' y '*Array*'. Por ejemplo, si se selecciona el formato '*Array*', los datos serán guardados en una matriz, donde en la primera columna se almacenan el vector del tiempo de simulación, y de la segunda columna el vector de la señal de entrada del bloque.

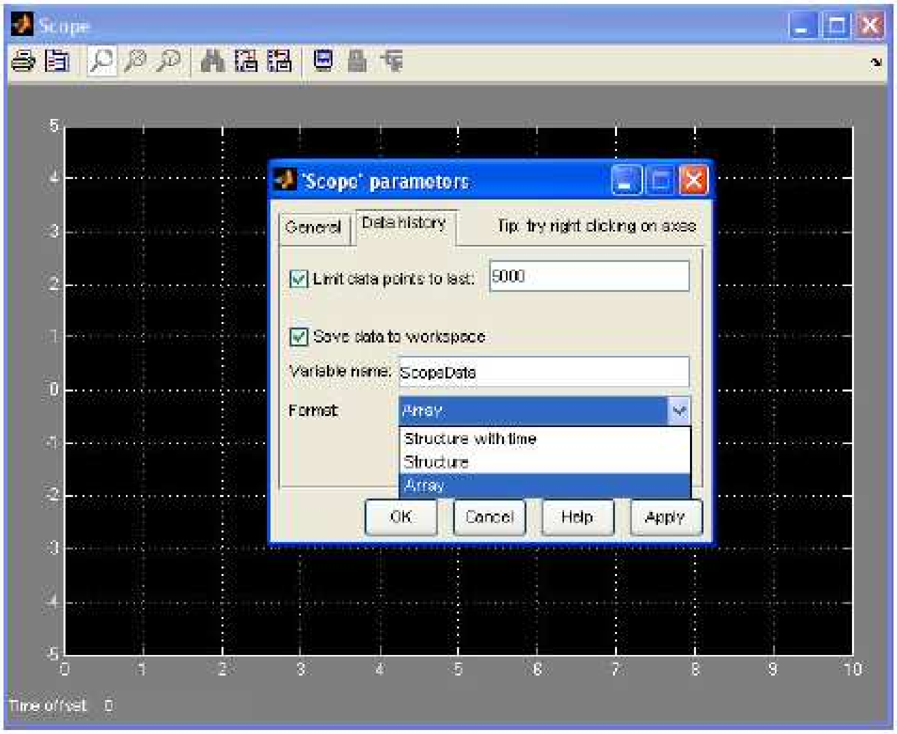
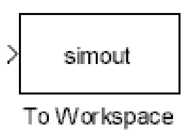


Figura # 2.45: Icono *Parameters*.
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.3.2. Bloque To Workspace



El bloque '*To Workspace*' envía su entrada al espacio de trabajo '*workspace*' de MatLab. Este bloque envía los valores de la entrada a una variable con el nombre especificado en la opción '*Variable name*'. La opción '*Save format* ' determina el formato de la variable de salida. Tal como se muestra en la figura 2.46.

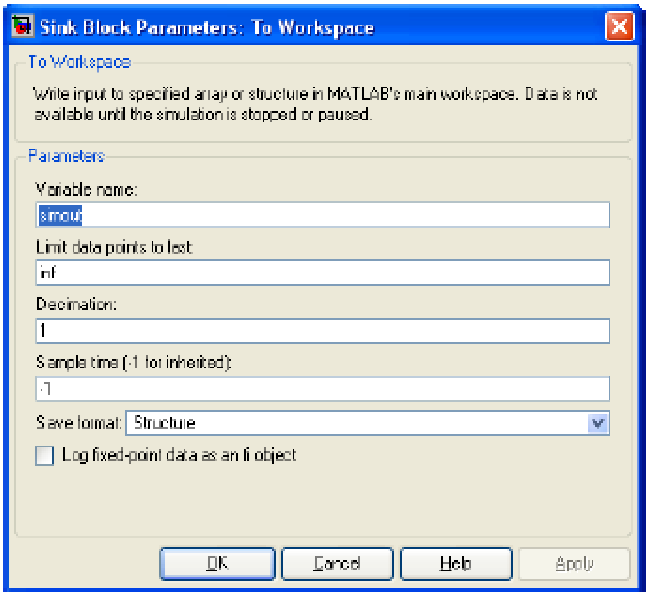


Figura # 2.46: Parámetros del bloque *To Workspace*
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.4. Librería de Fuentes (Sources)

El autor tomo la explicación textual del tema de**:** Alicia Arce Rubio, G. V. (2009)
En este apartado se presentarán los principales bloques que sirven como fuentes de señales en la simulación. Estos bloques están definidos en la librería '*Source*'. Serán comentados los siguientes bloques: '*From Workspace*', '*Constant* ', '*Signal Generator* ', '*Ramp*', '*Sine Wave*' y '*Step*'.

2.23.4.1. Bloque From Workspace

El bloque '*FromWorkspace*' lee datos del espacio de trabajo 'workspace' de MatLab. Los datos del espacio de trabajo son especificados con el parámetro '*Data*' a través de una matriz de dos dimensiones (ej. [t,u] son dos variables definidas en el espacio de trabajo). Tal como se muestra en la figura 2.47.

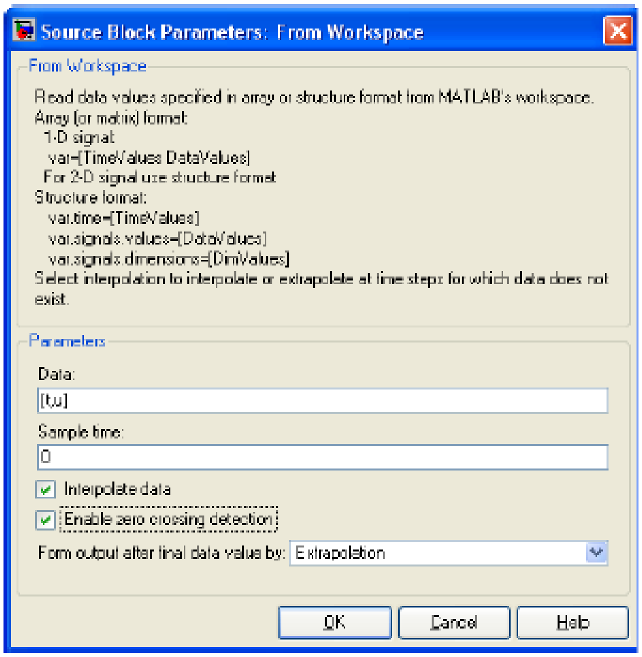
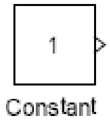


Figura # 2.47: Parámetros del bloque *From Workspace*
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.4.2. Bloque Constant



El bloque ‘*Constant* ’ es usado para definir un valor constante real o complejo. Este bloque acepta salidas escalares, vectores (1-D) o matrices (2-D), dependiendo de la dimensión del parámetro ‘*Constant value*’ que se especifica y si la opción ‘*Interpreter vector parameters as 1-D*’ está seleccionada o no. La salida del bloque posee la misma dimensión y los mismos elementos que la opción ‘*Constant value*’. Si se configura esta opción como un vector (matriz de 1-D), se debe marcar ‘*Interpreter vector parameters*’ como 1-D. Si esta opción no es debidamente configurada el bloque considera el parámetro ‘*Constant value*’ como una matriz 2-D. Tal como se muestra en la figura 2.48.

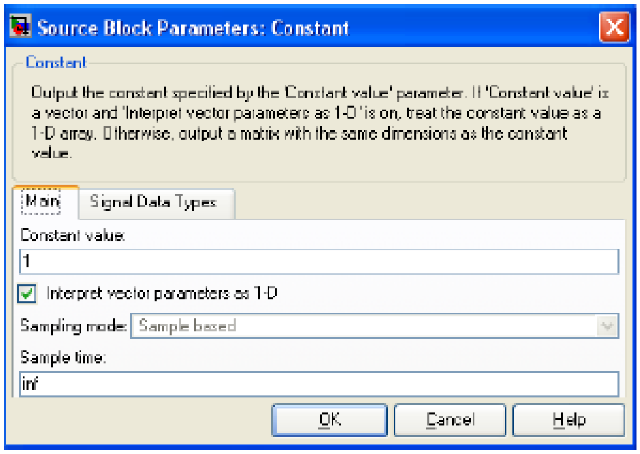
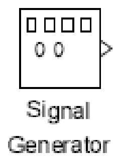


Figura # 2.48: Parámetros del bloque *Constant*.
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.4.3. Bloque Signal Generator



Bloque '*Signal Generator* ' puede generar cuatro diferentes tipos de formas de onda: onda seno ('*sine*'), onda cuadrada ('*square*'), onda diente de sierra ('*sawtooth*') y onda aleatoria ('*ramdom*'). Los parámetros de las señales son expresados en hercios o radianes por segundo. Se puede invertir la onda configurando el valor de la amplitud en negativo en la ventada de dialogo de parámetros. Tal como se muestra en la figura 2.49.

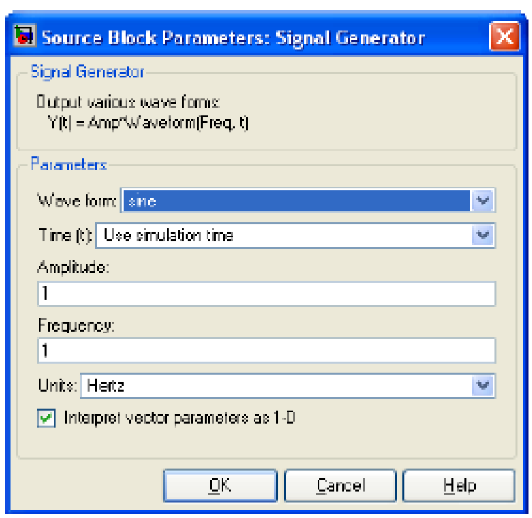
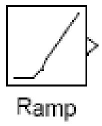


Figura # 2.49: Parámetros del bloque *Signal Generator*.

Fuente: Alicia Arce Rubio, G. V. (2009)

2.23.4.4. Bloque Ramp



El bloque '*Ramp*' genera una señal que empieza en un instante de tiempo especificado con un valor también previamente configurado y que evoluciona con una pendiente determinada en el bloque. Las características de la señal generada son configuradas mediante las siguientes opciones: pendiente ('*Slope*'), tiempo de inicio ('*Start Time*') y la condición inicial de la salida ('*Initial Output* '). Tal como se muestra en la figura 2.50.

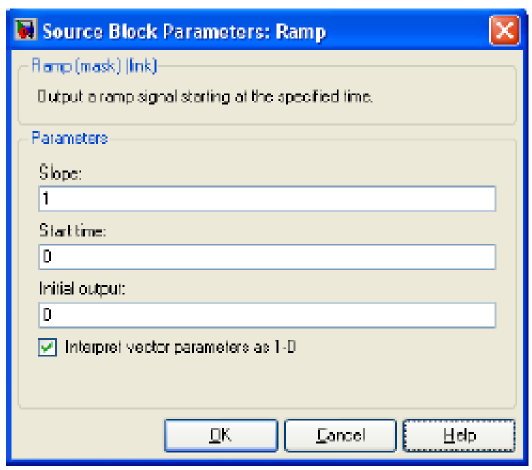
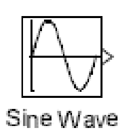


Figura # 2.50: Parámetros del bloque *Ramp*.
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.4.5. Bloque Sine Wave



El bloque '*Sine Wave*' genera una onda seno. Se puede generar una onda coseno configurando el parámetro de fase ('*Phase*') con el valor p/2. El bloque '*SineWave*' puede ser definido de dos modos diferentes a través del parámetro '*Sine type*' como modo basado en tiempo o como modo basado en muestras. El modo basado en tiempo posee dos sub-modos: sub-modo continuo o sub modo discreto. Se utiliza el parámetro '*Sample time*' para especificar que el bloque trabaje en sub-modo continuo o discreto. Para el sub-modo continuo se especifica el valor 0, y para el sub-modo discreto se especifica un valor mayor que cero. El modo basado en muestras requiere un tiempo discreto finito. Un valor del parámetro '*Sample time*' mayor que cero provoca que el bloque se comporte como si estuviera siendo modificado por un mantenedor de orden cero '*Zero Order Holder* '. La ventana de dialogo de configuración de parámetros de este bloque es presentada en la figura 2.51.

Los parámetros de configuración son descritos a continuación,

- *Amplitud*: la amplitud de la señal;
- *Bias*: valor (DC) constante agregado al seno para producir una salida con '*offset* ' en el eje-*y*;.
- *Frequency*: la frecuencia en radianes por segundo. Este parámetro aparece solo para el modo basado en tiempo.
- *Samples per period*: número de muestras por periodo. Este parámetro aparece solo para el modo basado en muestras.
- *Phase*: el desfase en radianes. Este parámetro aparece para el modo basado en tiempo.
- *Number of offset samples*: El desplazamiento en número de muestras de tiempo. Este parámetro aparece solo para el modo basado en muestras.
- *Sample time:* Periodo de muestreo. El valor patrón es cero, pero si el tipo de seno es basado en muestras se debe definir el periodo de muestreo mayor que cero.

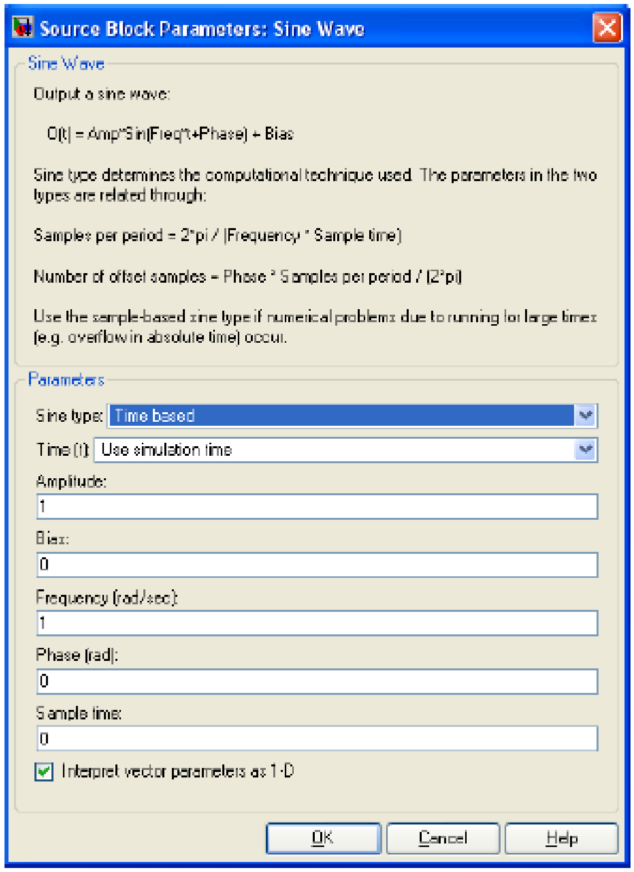
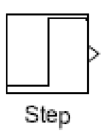


Figura # 2.51: Parámetros del bloque *Sine Wave*.
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.4.6. Bloque Step



El bloque '*Step*' genera un escalón entre dos niveles definidos en un espacio de tiempo especificado. Si el tiempo de simulación es menor que el valor del parámetro '*Step time*', la salida del bloque será el valor del parámetro '*Initial value*'. Para tiempos de simulación mayores o iguales que el valor de '*Step time*', la salida es el valor del parámetro '*Final value*'. Tal como se muestra en la figura 2.52.

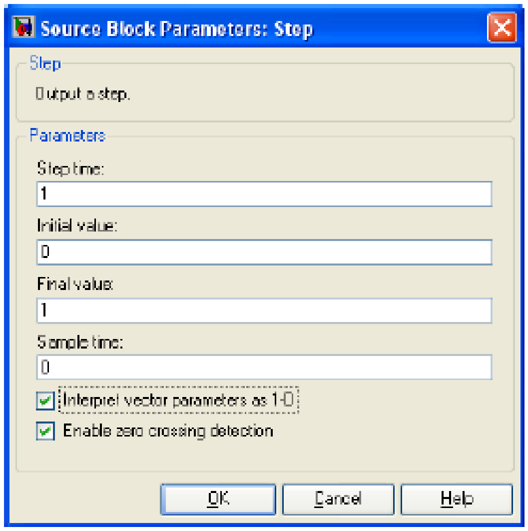


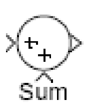
Figura # 2.52: Parámetros del bloque *Step*.
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.5. Librería de Operaciones Matemáticas (Math Opera-tions)

El autor tomo la explicación textual del tema de**:** Alicia Arce Rubio, G. V. (2009)

En este apartado se presentarán algunos bloques de operaciones matemáticas comúnmente utilizadas en el entorno Simulink.

2.23.5.1. Bloque Sum



El bloque '*Sum*' es la implementación del bloque suma. Este bloque realiza las operaciones de adición o sustracción de sus entradas, pudiendo sumar o sustraer entradas escalares, vectoriales o matriciales. Se puede también sumar los elementos de un único vector entrada. Las operaciones del bloque son definidas en el parámetro '*List of Signs*': más (+), menos (-) y separador (|). El separador crea un espacio extra entre puertos en el icono del bloque. La forma del icono se puede definir como redonda o rectangular (del inglés *round* o *rectangular*) a través del parámetro '*Icon shape*' en la ventana de dialogo de parámetros. Tal como se muestra en la figura 2.53.

Si hay dos o más entradas, el número de operaciones de suma o resta debe ser igual al número de entradas. Por ejemplo, "+−+" requiere tres entradas y se configura el bloque para sustraer la segunda entrada a la primera entrada, y luego sumar la tercera.

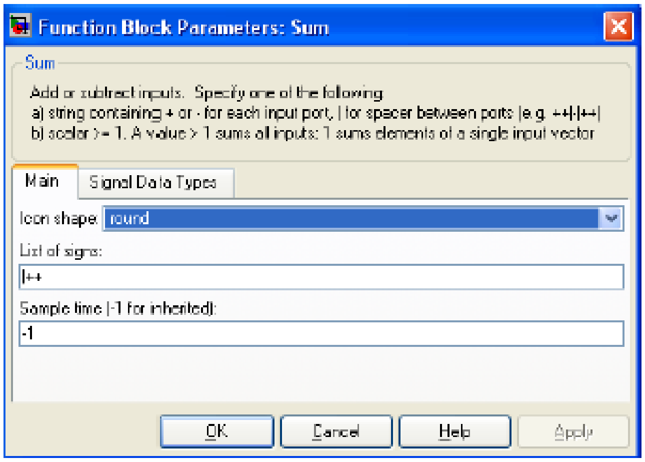
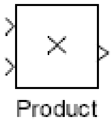


Figura # 2.53: Parámetros del bloque *Sum*
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.5.2. Bloque Product



El bloque '*Product* ' realiza la multiplicación o división de sus entradas. Este bloque calcula la salida multiplicando elemento a elemento o matricialmente, dependiendo del valor del parámetro '*Multiplication*'. El número de operaciones se configura con el parámetro '*Number of inputs*'. Tal como se muestra en la figura 2.54.

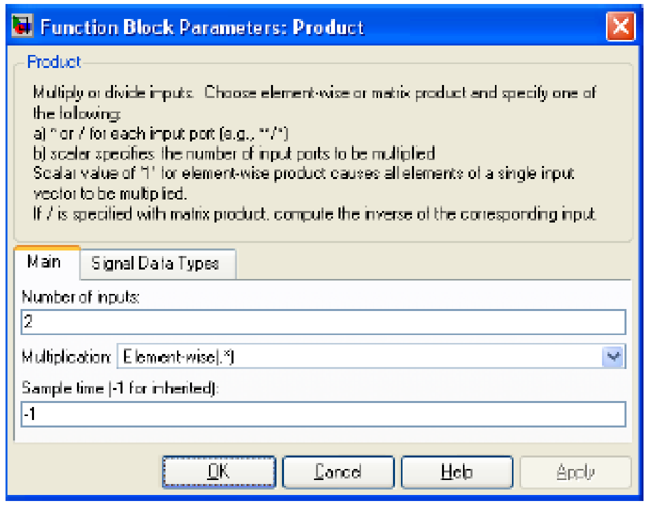
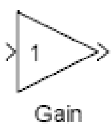


Figura # 2.54:Parámetros del bloque *Product*.
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.5.3. Bloque Gain



El bloque '*Gain*' multiplica la entrada por un valor constante (ganancia). La entrada y la ganancia pueden ser un escalar, un vector o una matriz. El valor de la ganancia se especifica a través del parámetro '*Gain*'. El parámetro '*Multiplication*' determina se la multiplicación es matricial o elemento a elemento. El orden de las multiplicaciones en las operaciones matriciales es configurado a través de este parámetro. Tal como se muestra en la figura 2.55.

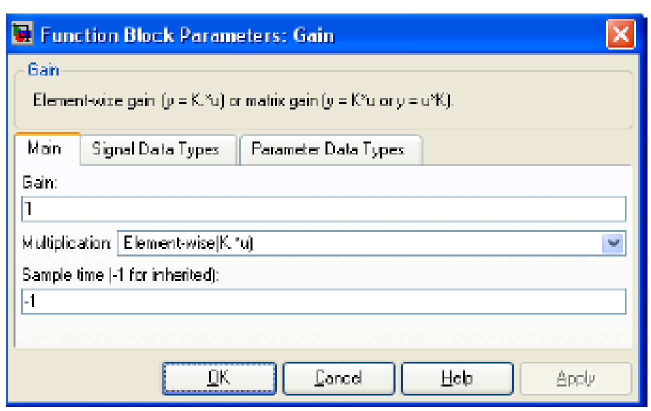
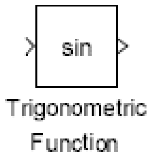


Figura # 2.55: Parámetros del bloque *Gain*.
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.5.4. Bloque Trigonometric Function



El bloque '*Trigonometric Function*' realiza las principales funciones trigonométricas: seno *sine*, coseno *cosine* y tangente *tangent* ; y las funciones trigonométricas inversas: arco seno *asin*, arco coseno *acos*, arco tangente *atan* y *atan2*; funciones hiperbólicas: *sinh*, *cosh* y *tanh*, y las funciones hiperbólicas inversas: *asinh*, *acosh*, *atanh*. Si se elige la función *atan2*, el bloque presenta dos entradas, la primera entrada es el eje-*y* o la parte compleja del argumento de la función y la segunda entrada es el eje-*x* o la parte real del argumento de la función. Tal como se muestra en la figura 2.56.

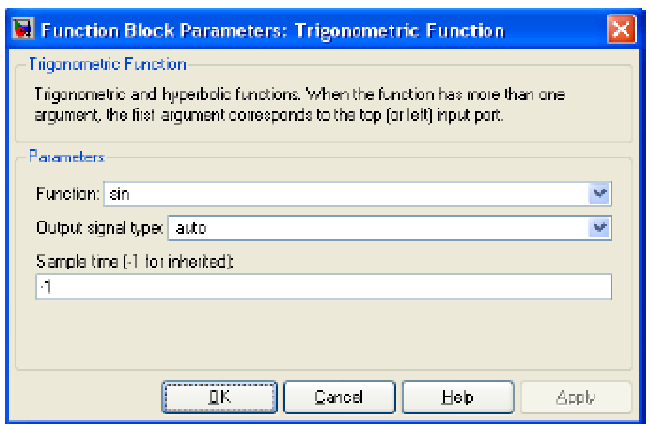
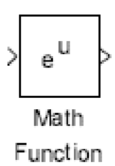


Figura # 2.56: Parámetros del bloque *Trigonometric Function*.
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.5.5. Bloque Math Function



El bloque '*Math Function*' implementa las siguientes funciones matemáticas: exp,log, 10*u*, log10, *magnitude*2, square, sqrt, pow, conj (conjugado complejo), reciprocal, hypot (cálculo de la raíz cuadrada de la suma de cuadrados), rem (resto de la división), mod (entero de la división), transpose (traspuesta de un vector o matriz) y hemiltian (una matriz cuadrada, tal que *AT* = *A*).

La salida del bloque es el resultado de la operación de la función sobre la entrada. El nombre de la función aparece sobre el bloque. Se utiliza el bloque '*Math Function*' en el caso que se desee una salida vectorial o matricial ya que el bloque '*Fcn*' tiene características similares pero las salidas son sólo escalares. Tal como se muestra en la figura 2.57.

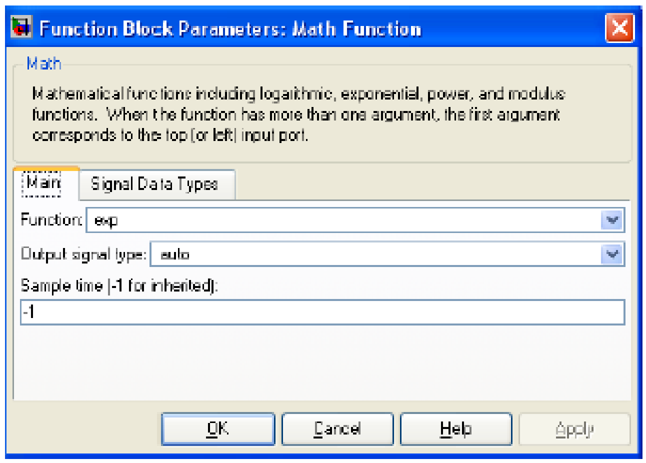
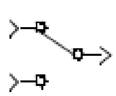


Figura # 2.57: Parámetros del bloque *Math Function*.
Fuente: Alicia Arce Rubio, G. V. (2009).

2.23.6. Librería de Ruta de Señales (Routing Signals)

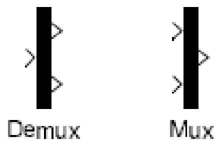
El autor tomo la explicación textual del tema de**:** Alicia Arce Rubio, G. V. (2009)
En este apartado se presentarán tres bloques de la librería '*Signal Routing*': '*Manual Switch*', '*Mux*' y '*Demux*'. Bloque Manual Switch



Manual Switch

El bloque '*Manual Switch*' conmuta sus entradas pasando sólo una de ellas a través de su salida. No existe ventana de dialogo para este bloque, por lo que para conmutar entre las entradas se debe pulsar dos veces sobre el bloque. Este bloque mantiene el estado determinado cuando el archivo Simulink es guardado.

2.23.6.1. Bloques Mux y Demux



El bloque '*Mux*' combina sus entradas en una única salida. Las señales de entrada pueden ser escalares, vectores o matrices. El parámetro '*Number of Inputs*' permite especificar el número de señales de entrada y su dimensión. Un valor de −1 significa que el puerto correspondiente puede aceptar señales de cualquiera dimensión. Tal como se muestra en la figura 2.58.

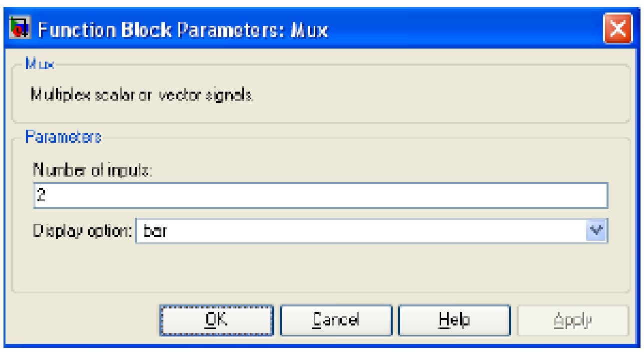


Figura # 2.58: Parámetros del bloque *Mux*
Fuente: Alicia Arce Rubio, G. V. (2009).

El bloque '*Demux*' extrae las componentes de una señal de entrada y provee las componentes en separadas señales. El bloque acepta tanto señales vectoriales como buses de señales. El parámetro '*Number of outputs*' permite especificar el número y dimensión de cada puerto de salida. Si no se configura la dimensión de las salidas, el bloque lo determina automáticamente.

CAPÍTULO 3 - METODOLOGIA

DISEÑO Y MODALIDAD DE LA INVESTIGACIÓN

Para alcanzar los objetivos propuestos se seleccionaron los siguientes métodos de investigación:

- Método de análisis
- Método de Organización
- Método de investigación acción y Exploratoria
- Métodos de comprobación y de observación (pre-Experimental)

3.1. Justificación de la Elección del Método

El entorno de simulación Matlab/Simulink representa una herramienta muy adecuada para el estudio, análisis, comprobación y observación de los circuitos transitorios o sistemas de control de primero y segundo orden. Las ventajas más importantes de Matlab-Simulink son las siguientes:

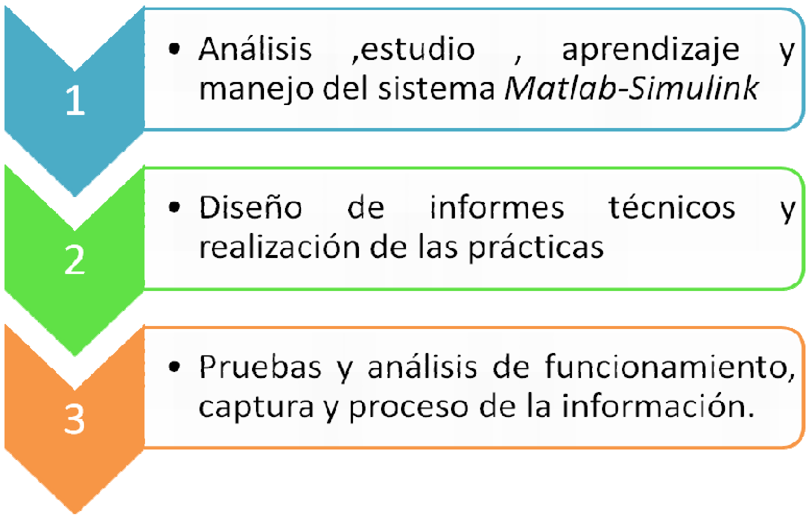
El lenguaje de programación de Matlab-Simulink contiene herramientas necesarias para la simulación del circuito transitorio mediante tres técnicas disponibles entre los bloques de Simulink. Estos bloques son los integradores y multiplicadores (para plantear la ecuación diferencial), el bloque "State Space" (para resolver un sistema mediante variables de estado) y el bloque "Transfer Fcn" (para resolver un circuito conocida su función de transferencia).

Matlab posee los comandos adecuados para el estudio y análisis de los circuitos transitorios mediante las tres técnicas citadas incluyendo comandos que de una forma directa analizan la función de transferencia de un circuito.

El entorno Matlab-Simulink permite la visualización de cualquier forma de onda del circuito mediante los bloques "scope" y "plot xy" de forma interactiva.

3.2. Procedimientos a seguir para la realización de las prácticas.

Continuando con la organización que demanda la realización de esta tesis se prestó un computador con el programa matlab-simulink del laboratorio de Electrónica, para inmediatamente iniciar con la planificación y procedimientos a seguir para las prácticas:



Adicionalmente para el estudio de los diferentes temas de la tesis se investigó en distintos textos de bibliotecas virtuales como E-libro y ProQuest también en páginas web como libros Google y documentos de Universidades sobre información acerca de los sistemas de control de primero y segundo orden usando el lenguaje de programación Matlab-Simulink.

3.3. Prácticas básicas Realizadas

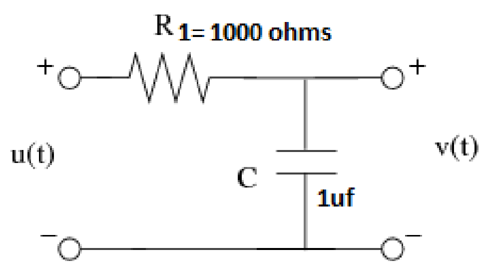
En esta tesis utilizamos el método de comprobación y de observación (pre-Experimental), del paradigma empírico analítico con un enfoque cuantitativo, para la realización de las prácticas, con procedimientos propios de ciencias naturales con conocimientos científicos de la realidad, observando y comprobando los diferentes sucesos de cada uno de los circuitos y sistemas simulados.

Practica # 1: Sistemas de Primer Orden: Circuito Fuente RC en Serie, Respuesta a una entrada escalón.

1. Objetivo

Verificar el comportamiento del sistema a una entrada del tipo escalón o paso, comprobar si el tiempo de establecimiento de la señal del circuito, es decir, cuanto se tarda el sistema en alcanzar su estado estacionario.

Obtener la función de transferencia en matlab y el diagrama en bloques en Simulink del filtro RC del siguiente circuito.



2. Equipos y software Utilizados

Computadora

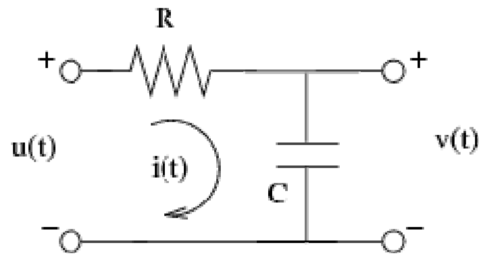
Software Matlab

Software Simulink

3. Desarrollo de la practica

Para obtener una función de transferencia de este circuito se puede realizar el siguiente cálculo matemático manual con las siguientes expresiones:

Considerando que i(t) es la intensidad que recorre la única malla del circuito, tal y como se observa en la siguiente figura:



La caída de tensión en la misma u(t) y la que se produce en los bornes del condensador se pueden expresar de la forma:

$$\begin{aligned} u(t) &= Ri(t) + v(t) \\ v(t) &= \frac{1}{C}\int i(\tau)d\tau \end{aligned}$$

Despejando la intensidad de la segunda ecuación se tiene que la misma es i(t) = C$\frac{dv(t)}{dt}$, que sustituyendo en la primera permite obtener la ecuación diferencial del sistema en función únicamente de las variables de entrada u(t) y de salida v(t):

$$u(t) = RC\frac{dv(t)}{dt} + v(t)$$

Finalmente, aplicando la transformada de Laplace a la anterior ecuación (considerando las condiciones iniciales nulas) y despejando se tiene la función de transferencia del circuito:

$$U(s) - RCsV(s) + V(s) \quad \longrightarrow \quad \frac{V(s)}{U(s)} - \frac{1}{1+RCs}$$

En matlab se obvia todo este proceso y se introducen los datos de la siguiente manera:

```
%ANALISIS DE SISTEMA DE PRIMER ORDEN
%====================================
%======== SISTEMA ELECTRICO ========
%========== Fuente R  C ============
%====================================
%UTILIZANDO FUNCION DE TRANSFERENCIA
%Ingresando las constantes.
%Valor de la Resistencia R=1[KOhmio]
R=1000;
%Valor del Capacitor C=1[uF]
C=1*10^(-6);
%Ingresando Numerador
num1=1;
%Ingresando Denominador
den1=[R*C 1];
%Definiendo la Funcion de Transferencia
G1=tf(num1,den1);
%Mostrando la Funcion de Transferencia
display(G1)
-------------------------------------------
```

```
Transfer function:
     1
-----------
0.001s + 1

%==================================
%Respuesta a una entrada Paso
%Asignacion a una figura
figure(10)
%Aplicando una entrada paso
step(G1)
%Definiendo la cuadricula
grid
%Titulo
title('Respuesta a una Entrada Paso')
%Etiqueta de la Axisa
xlabel('Tiempo ')
%Etiqueta de la Ordenada
ylabel('Voltaje [V]')
```

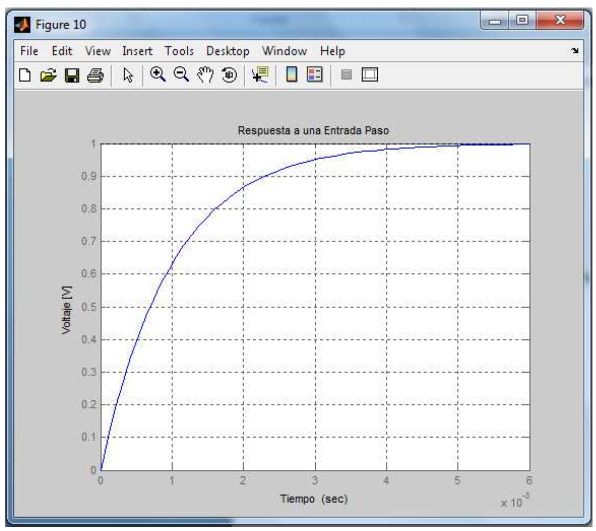


Figura # 3.1: Respuesta a una entrada paso en Matlab
Fuente: Autores

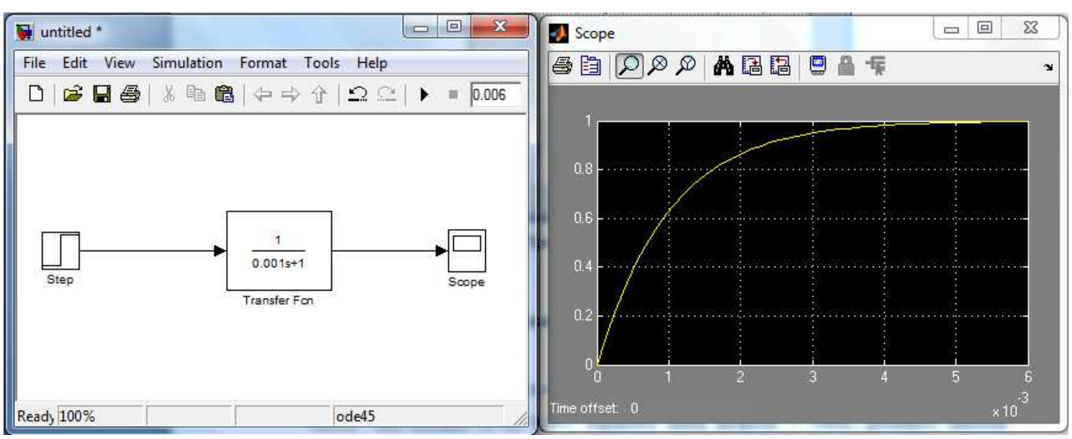


Figura # 3.2: Diagrama de bloques de la función de transferencia en Simulink del filtro RC
Fuente: Autores

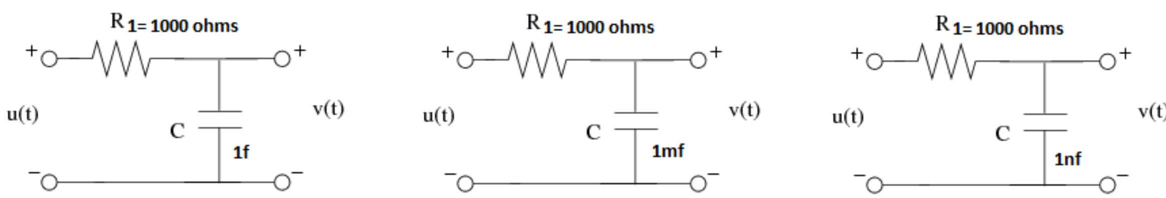
4. Resultados y conclusiones

En la gráfica el resultado nos indica que el tiempo de establecimiento de la señal de salida del circuito es de 0,001 segundos hasta alcanzar 1 voltio, podemos también concluir que lo estudiado en lo teórico con respecto al análisis en el dominio del tiempo en los sistemas de primer orden si se cumple.

Practica # 2: Sistemas de Primer Orden: Circuito Fuente RC en Serie: Respuesta a una entrada Escalón o Paso ante Cambios de C:[1f; 1mf ;1nf]

1. Objetivo

Lo que se quiere lograr en esta práctica es observar que comportamiento tiene el tiempo y el voltaje al cambiar el valor de la capacitancia en el circuito.



2. Equipos y software Utilizados

Computadora

Software Matlab

Software Simulink

3. Desarrollo de la practica

Lo primero que haremos es Obtener la función de transferencia en matlab y el diagrama en bloques en Simulink del filtro RC de los tres circuitos de la siguiente manera:

```
%Respuesta a una entrada Paso ante
%Cambios de C:[1F 1mF 1nF]
Cv=[1 1*10^(-3) 1*10^(-9)];
%Lazo para generar cambios de Cv
for i=1:3
    %Ingresando Denominador
    den=[R*Cv(i) 1];
    %Definiendo la Funcion de Transferencia
```

```
    G=tf(num1,den);
    %Mostrando la Funcion de Transferencia
    display(G)
    %Respuesta a una entrada Paso
    %Asignacion a una figura
    figure(20+i)
    %Aplicando una entrada paso
    step(G)
    %Definiendo la cuadricula
    grid
    %Titulo
    title(['Respuesta a una Entrada Paso con',' C=
',num2str(Cv(i)),'F'])
    %Etiqueta de la Accisa
    xlabel('Tiempo ')
    %Etiqueta de la Ordenada
    ylabel('Voltaje [V]')
end


Transfer function:
    1
----------
1000 s + 1


Transfer function:
  1
-----
s + 1


Transfer function:
     1
------------
1e-006 s + 1
```

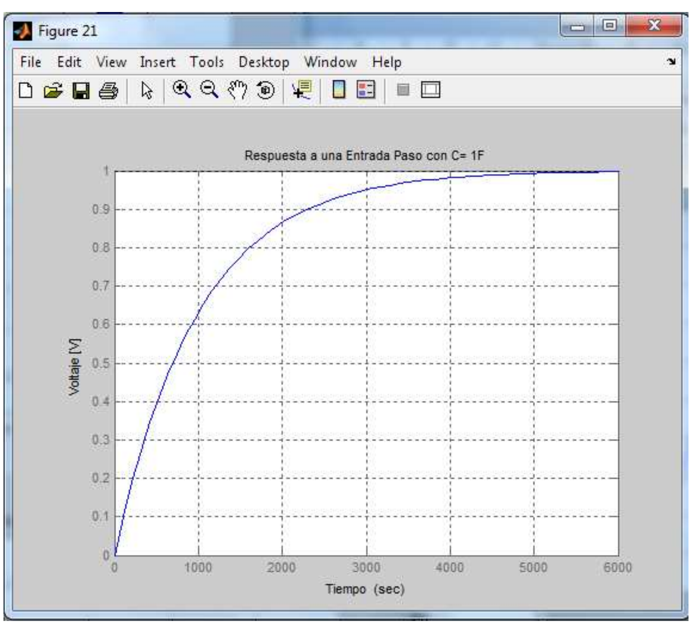


Figura # 3.3: Respuesta a una entrada paso con un capacitor de 1f
Fuente: Autores

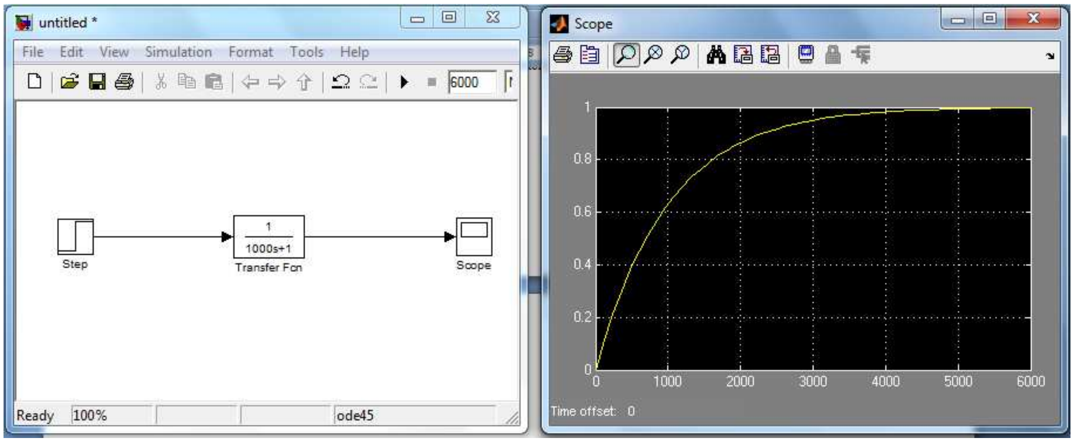


Figura # 3.4: Diagrama de bloques en Simulink con su respectiva señal a una entrada paso con un capacitor de 1f
Fuente: Autores

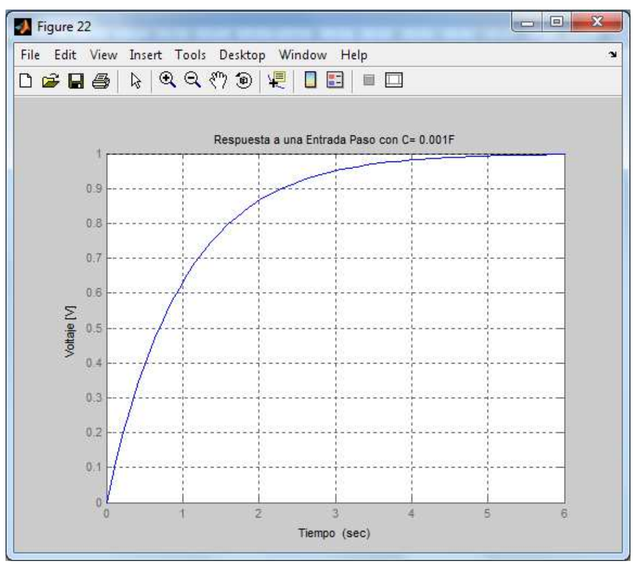


Figura # 3.5: Respuesta a una entrada paso con un capacitor de 1mf
Fuente: Autores

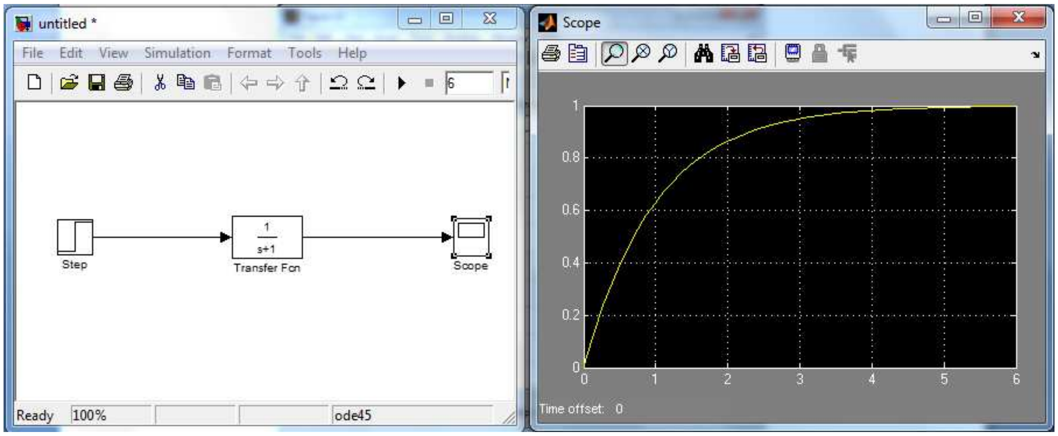


Figura # 3.6: Diagrama en bloques en Simulink con su respectiva señal a una entrada paso con un capacitor de 1mf
Fuente: Autores

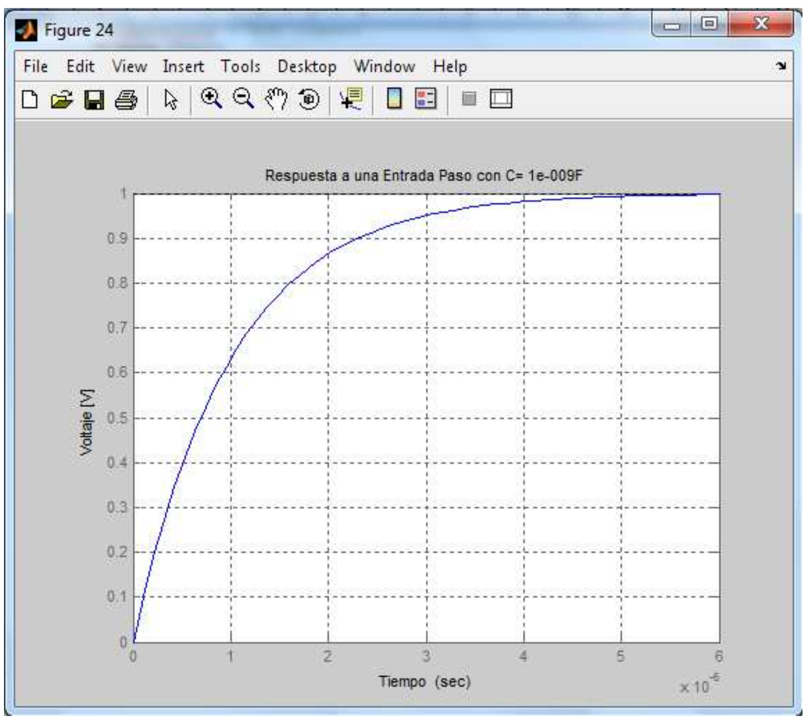


Figura # 3.7: Respuesta a una entrada paso con un capacitor de 1nf
Fuente: Autores

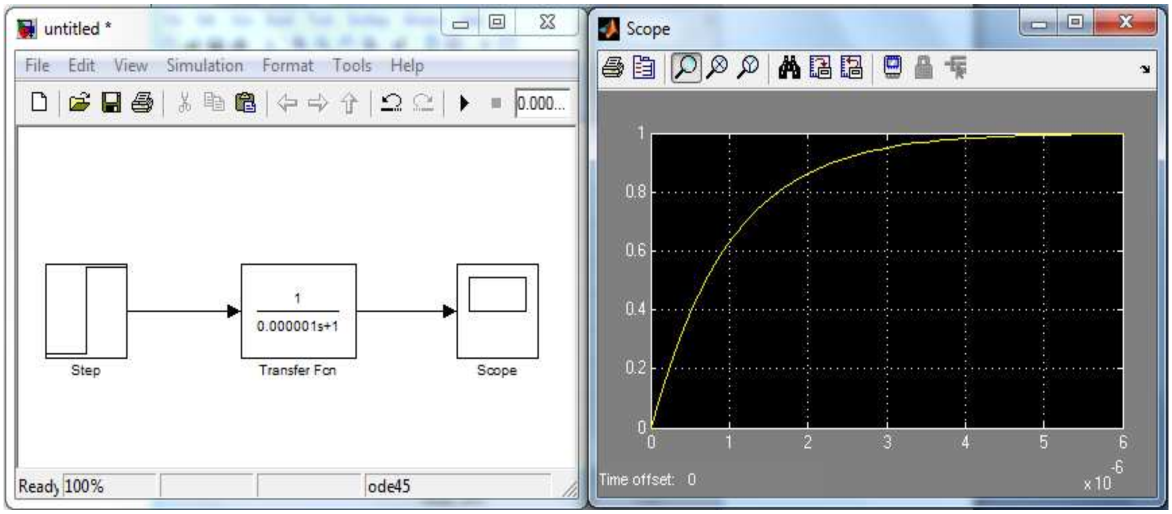


Figura # 3.8: Diagrama en bloques en Simulink con su respectiva señal a una entrada paso con un capacitor de 1nf
Fuente: Autores

4. Resultados y conclusiones

La grafica de los circuitos nos muestra que la fuente de voltaje de un voltio y la resistencia de 1000omhs son iguales para los tres, mientras que el capacitor se lo cambia con uno de 1f, 1mf y 1uf, esto nos arroja diferentes funciones de transferencia y deferentes comportamientos del circuito en el tiempo, podemos observar que en la figura3.4 el tiempo que pasa para estabilizarse el circuito en su salida es de 6000 segundos hasta llegar a un voltio, con un capacitor de 1f, la figura 3.6 nos indica que con un capacitor de 1mf el tiempo que pasa para estabilizarse el circuito en su salida es de 6 segundos , hasta llegar a un voltio , mientras que en la figura 3.7 con un capacitor de 1uf el tiempo que pasa para estabilizarse el circuito en su salida de 0,000006 segundos, esto quiere decir que manteniendo el voltaje y la resistencia constantes y el capacitor variable, el tiempo de disparo o carga en el circuito para estabilizarse varía, con lo que podemos concluir que a mayor capacitancia mayor el tiempo de disparo o carga del capacitor en el circuito para estabilizarse, mientras que a menor capacitancia , menor el tiempo de disparo o carga del capacitor en el circuito para estabilizarse; esto quiere decir que en un circuito RC en serie con un voltaje y una resistencia constantes su tiempo de carga de capacitancia o de estabilización en el tiempo es directamente proporcional al valor del capacitor.

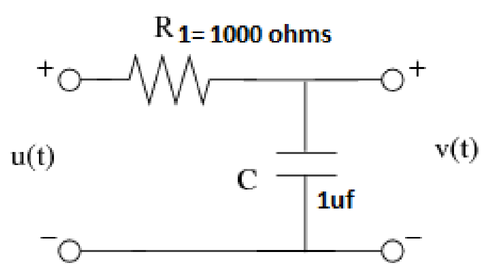
Practica # 3: Sistemas de Primer Orden: Circuito Fuente RC en Serie: Respuesta a una entrada Impulso.

La transformada de Laplace de un impulso unitario es igual a la unidad por lo que la transformada de la respuesta impulsional es igual a la expresión de la función de transferencia.

La respuesta impulsional de un sistema es pues la transformada inversa de la función de transferencia, en el caso de los sistemas de primer orden.

1. Objetivo

Verificar el comportamiento del sistema a una entrada del tipo impulso.
Obtener la función de transferencia en matlab y el diagrama en bloques en Simulink del circuito.



2. Equipos y software Utilizados

Computadora
Software Matlab
Software Simulink

3. Desarrollo de la practica

En Matlab la respuesta impulsional se puede obtener del mismo modo que la respuesta escalón paso, pero usando en este caso la función **impulse**, que tiene una sintaxis similar al comando **step**. , utilizado en la respuesta escalón-paso

```
%Respuesta a una entrada Impulso
%Asignacion a una figura
figure(30)
%Aplicando una entrada Impulso
impulse(G1)
%Definiendo la cuadricula
grid
```

```
%Titulo
title('Respuesta a una Entrada Impulso')
%Etiqueta de la Accisa
xlabel('Tiempo ')
%Etiqueta de la Ordenada
ylabel('Voltaje [V]')
Transfer function:

     1
-----------
0.01  s + 1
```

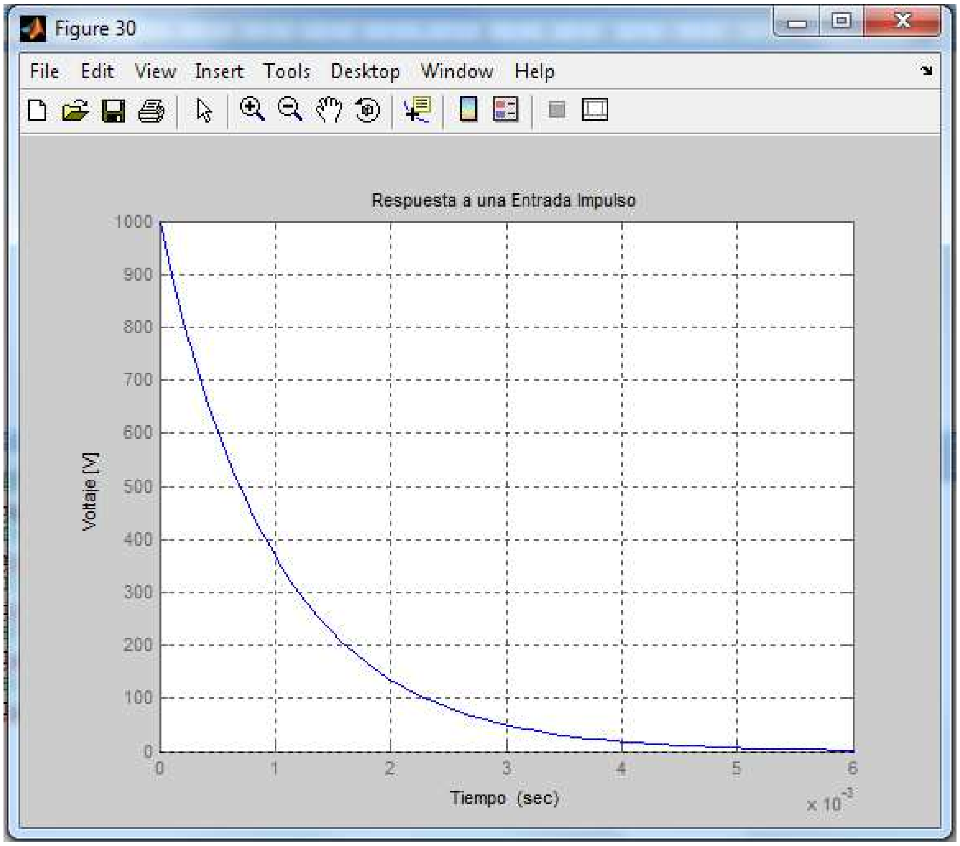


Figura # 3.9: Respuesta a una entrada impulso
Fuente: autores

4. Resultados y conclusiones

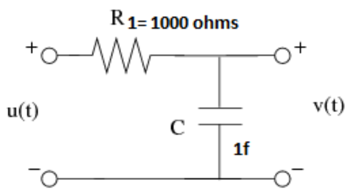
Un impulso es una señal que tiene una magnitud infinita y un ancho infinitesimalmente estrecho con un área uno, centrado en cero y puede ser representado como una suma infinita de sinusoides que incluye todas las frecuencias posibles.

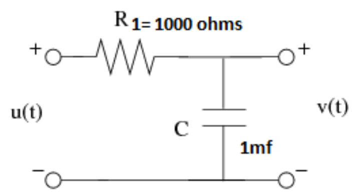
En este caso el impulso se utiliza como entrada a un sistema, la salida se conoce como la respuesta impulsional. La respuesta impulsional define el sistema, ya que todas las frecuencias posibles se representan en la entrada, con lo que podemos concluir que el sistema es estable por que vuelve a cero en un tiempo estimado.

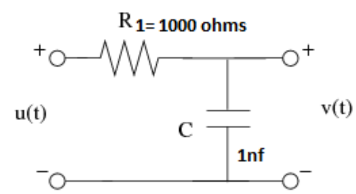
Practica # 4: Sistemas de Primer Orden: Circuito Fuente RC en Serie: Respuesta a una entrada Impulso ante cambios de C: [1f; 1mf; 1nf].

1. Objetivo

Lo que se quiere lograr en esta práctica es observar que comportamiento tiene el tiempo y el voltaje al cambiar el valor de la capacitancia en el circuito.







2. Equipos y software Utilizados

Computadora

Software Matlab

Software Simulink

3. Desarrollo de la Práctica

Lo primero que haremos es Obtener la función de transferencia en matlab y el diagrama en bloques en Simulink del filtro RC de los tres circuitos de la siguiente manera:

```
%Respuesta a una entrada Impulso ante
%Cambios de C:[1F 1mF 1nF]
Cv=[1 1*10^(-3) 1*10^(-9)];
%Lazo para generar cambios de Cv
for i=1:3
    %Ingresando Denominador
    den=[R*Cv(i) 1];
    %Definiendo la Funcion de Transferencia
    G=tf(num1,den);
    %Mostrando la Funcion de Transferencia
    display(G)
    %Respuesta a una entrada Impulso
    %Asignacion a una figura
    figure(40+i)
    %Aplicando una entrada Impulso
    Impulse(G)
    %Definiendo la cuadricula
    grid
    %Titulo
    title(['Respuesta a una Entrada Impulso con',' C=
',num2str(Cv(i)),'F'])
    %Etiqueta de la Accisa
    xlabel('Tiempo ')
```

```
    %Etiqueta de la Ordenada
    ylabel('Voltaje [V]')
end
Transfer function:
    1
----------
1000 s + 1


Transfer function:


  1
-----
s + 1


Transfer function:
     1
------------
1e-006 s + 1
```

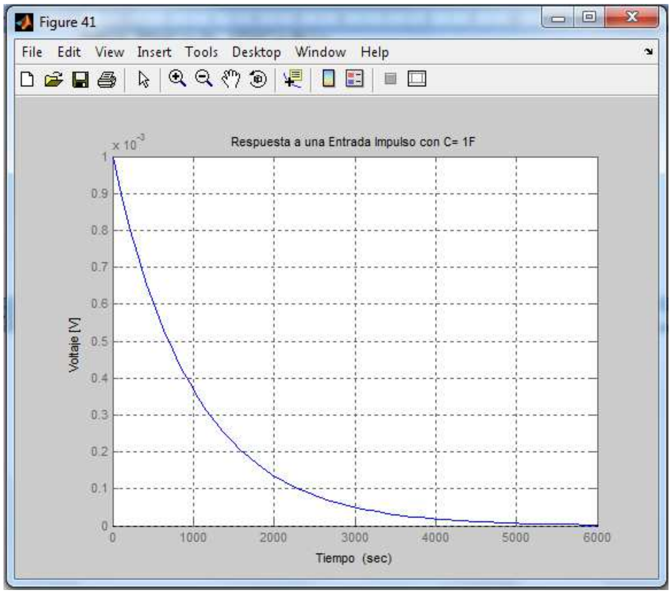


Figura # 3.10: Respuesta a una entrada impulso con un capacitor de 1f
Fuente: Autores

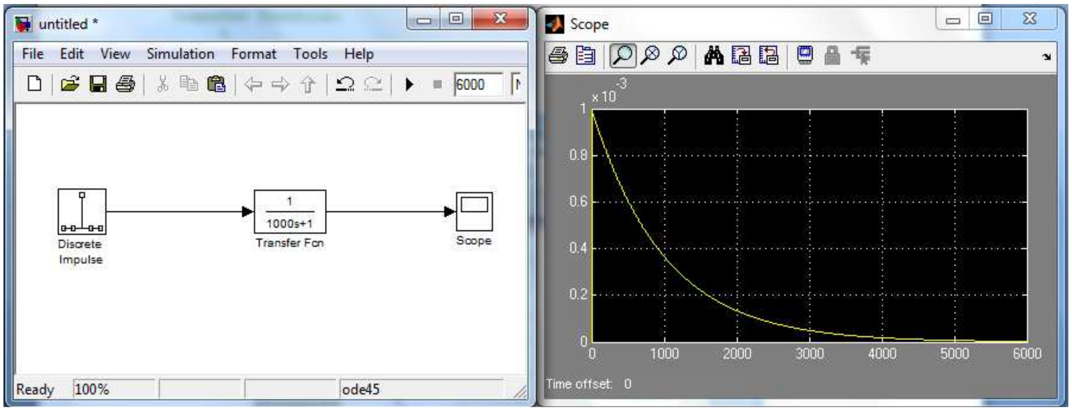


Figura # 3.11: Diagrama de bloques en Simulink con su respectiva señal a una entrada impulso con un capacitor de 1f
Fuente: Autores

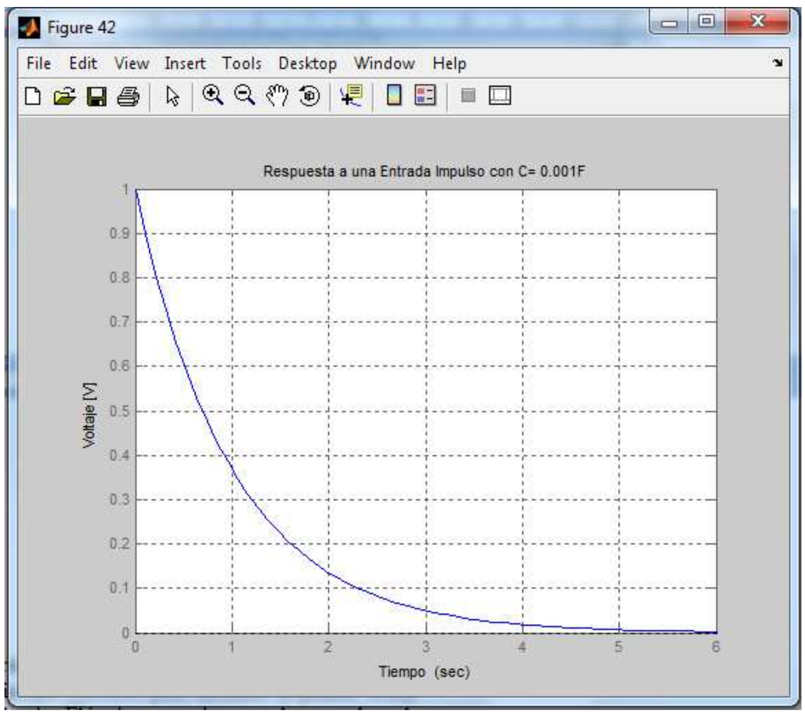


Figura # 3.12: Respuesta a una entrada impulso con un capacitor de 1mf
Fuente: Autores

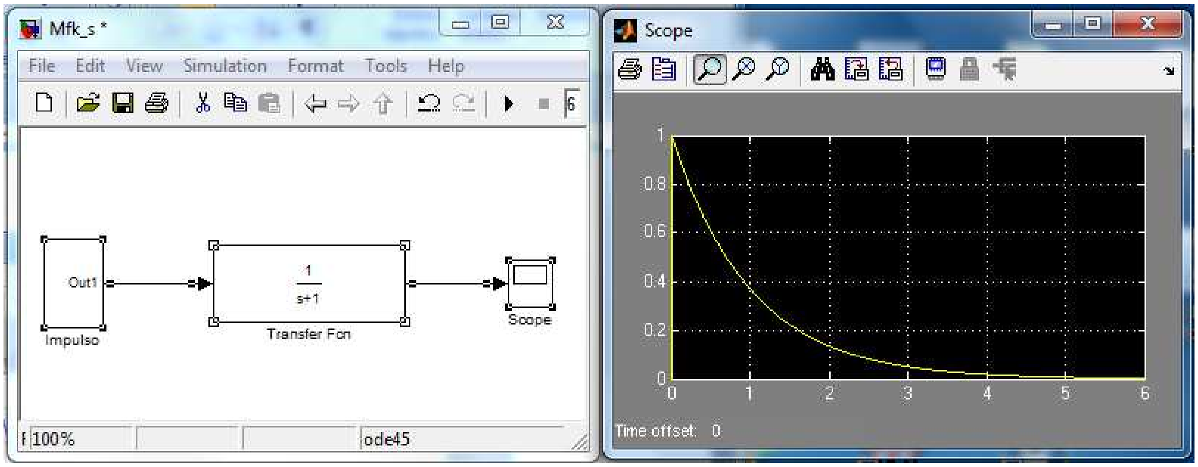


Figura # 3.13: Diagrama de bloques en Simulink con su respectiva señal a una entrada impulso con un capacitor de 1f
Fuente**: Autores**

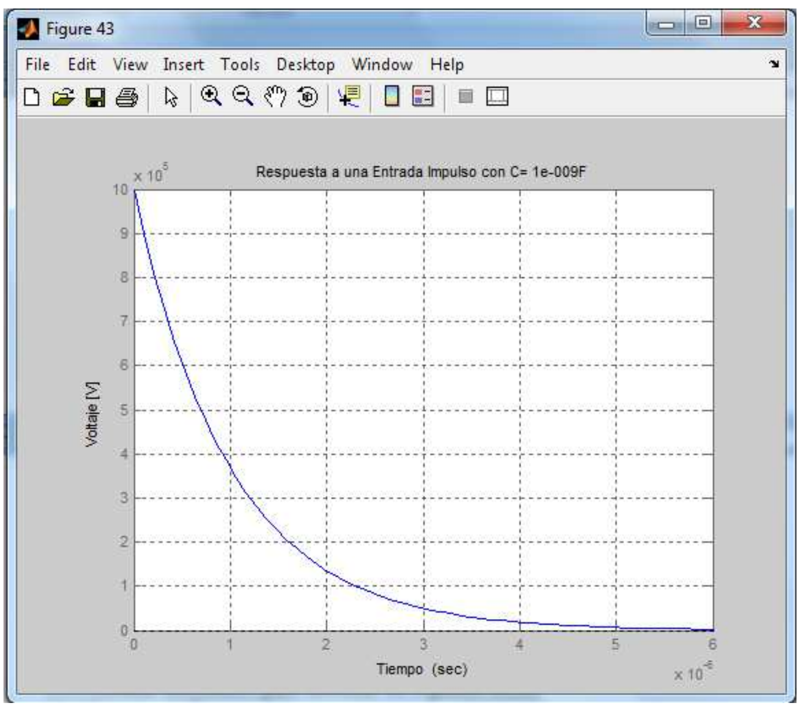


Figura # 3.14: Respuesta a una entrada impulso con un capacitor de 1nf
Fuente: Autores

4. Resultados y conclusiones

La respuesta a una entrada Impulso ante cambios de C:[1F 1mF 1nF] se la conoce como respuesta impulsional, en las gráficas nos podemos dar cuenta que aunque cambien los valores de los capacitores, el tiempo de estabilización del sistema también cambia, no importa la cantidad de tiempo que demore lo importante es que el sistema responde establemente y que tiende su respuesta a cero.

Practica # 5: Sistemas de Segundo Orden: Sistema Mecánico Fuerza M, F, K, Respuesta a una entrada paso.

1. Introducción

Los sistemas mecánicos son una parte fundamental de la vida común, ya que cualquier cuerpo físico se comporta como tal. En general los sistemas mecánicos son gobernados por la segunda ley de Newton, la cual establece para sistemas mecánicos de traslación que "la suma de fuerzas en un sistema, sean estas aplicadas o reactivas, igualan a la masa por la aceleración a que está sometida dicha masa".

$$\sum f = ma$$

Cuando se trata de sistemas mecánicos de rotación la segunda ley de Newton declara que "la suma de torques es igual al momento de inercia multiplicado por la aceleración angular".

En cualquiera de los casos anteriores se tiene diferentes elementos cuyo acoplamiento conforma al sistema mecánico completo, pudiendo además interactuar entre cada caso. A continuación se describen las generalidades de ambos tipos de sistemas mecánicos.

$$\sum T = J\alpha$$

Sistemas mecánicos de traslación.

Los sistemas mecánicos de traslación están integrados por el conjunto de elementos básicos resumidos en la siguiente tabla5.

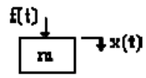
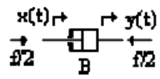
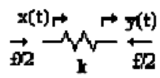
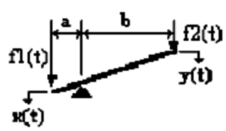
ELEMENTO	SIMBOLO	ECUACIÓN DE EQUILIBRIO	UNIDADES
MASA		$f_m = m \cdot \frac{d^2}{dt^2} x(t)$	[Kg]. ó [N.s/m]
AMORTIGUADOR		$f_B = B \cdot \frac{d}{dt}\left[x(t) - y(t)\right]$	coeficiente de fricción viscosa B = [N.seg/m]
RESORTE		$f_k = k \cdot \left[x(t) - y(t)\right]$	módulo de elasticidad k= [N/m]
PALANCA		$y = \frac{b}{a+b} \cdot x$ $f_2 = \frac{a}{a+b} \cdot f_1$	adimensional

Tabla 5: Elementos mecánicos de Traslación

En este caso las variables involucradas son desplazamientos, velocidades, aceleraciones y fuerzas. La disposición que guardan estos elementos entre sí da lugar a dos configuraciones denominadas arreglos mecánicos en serie y arreglos mecánicos en paralelo.

Elementos mecánicos en serie.

En un elemento mecánico en serie, la fuerza aplicada f(t) es igual a la suma de las fuerzas actuantes en cada elemento y todos los elementos tienen el mismo desplazamiento. Tal como se muestra en la figura 3.15.

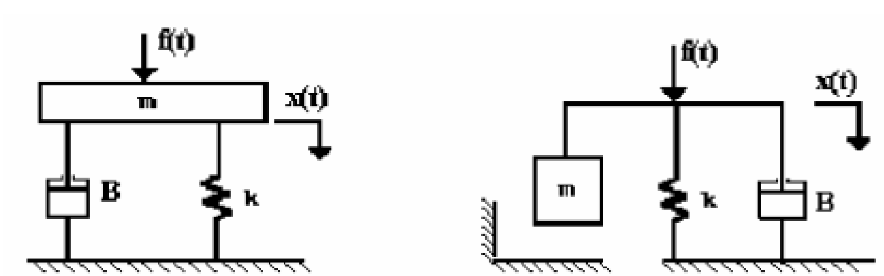


Figura # 3.15: Elementos mecánicos en Serie

La ecuación de equilibrio para el arreglo de la figura 3.15 es:

$$f(t) = m \cdot \frac{d^2}{dt^2}x(t) + B \cdot \frac{d}{dt}x(t) + k \cdot x(t)$$

Y su transformada de Laplace considerando condiciones iniciales iguales a cero es:

$$F(s) = (ms^2 + Bs + k)X(s)$$

Donde la impedancia mecánica es:

$$Z(s) = ms^2 + Bs + k$$

Elementos Mecánicos en Paralelo.

En este tipo de arreglo la fuerza aplicada f(t) se transmite a través de todos los elementos. Además, la deformación o corrimiento total es la suma de los desplazamientos de cada elemento. Tal como se muestra en la figura 3.16.

Un ejemplo de este tipo de

$$X(s) = \frac{F(s)}{k} + \frac{F(s)}{B_1 s} + \frac{F(s)}{B_2 s}$$

Arreglo en el que considerando las ecuaciones ya transformadas el desplazamiento total está dado por:

La relación fuerza a desplazamiento queda como:

$$F(s) = \frac{1}{\frac{1}{k} + \frac{1}{B_1 s} + \frac{1}{B_2 s}} X(s)$$

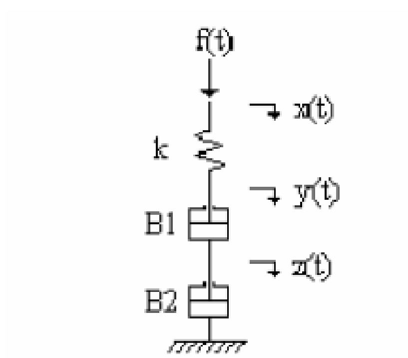


Figura # 3.16: Arreglo mecánico en paralelo

Donde la impedancia mecánica es:

$$Z(s) = \frac{1}{1/k + 1/B_1 s + 1/B_2 s}$$

Un comentario importante respecto al comportamiento de una masa es que esta no puede estar en paralelo con otros elementos a menos que sea el último de los elementos. Para ilustrar lo anterior veamos que en la figura 3.16, la masa, al ser el último elemento, participa como si estuviera en paralelo dando la ecuación que relaciona la fuerza con el desplazamiento de la forma:

$$F(s) = \frac{1}{1/k + 1/Bs + 1/ms} X(s)$$

Mientras que en la figura 3.17 al estar la masa colocada como un elemento intermedio, y tener el mismo desplazamiento y(t) en la parte superior e inferior, la sitúa en serie tanto con k1 como con k2 y B respecto al desplazamiento y(t) mientras que no tiene nada que ver con los desplazamientos x(t) y z(t) que afectan al comportamiento de los elementos k1 y k2-B respectivamente.

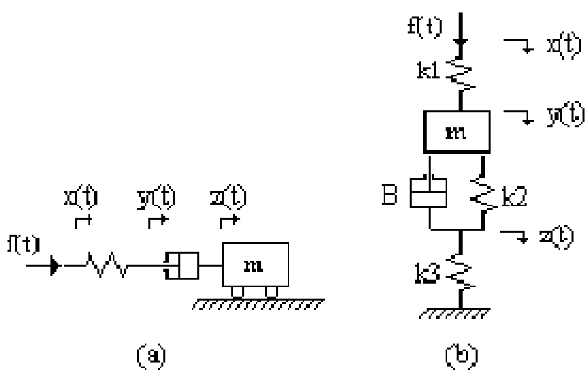


Figura # 3.17 a : Masa como elemento en paralelo b: Masa como elemento en serie

Para el caso de la figura 3.17 las ecuaciones de equilibrio en cada desplazamiento son:

En x(t):

$$f(t) = k_1 \cdot [x(t) - y(t)]$$

en y(t):

$$m \cdot \frac{d^2}{dt^2} y(t) + k_1 \cdot [y(t) - x(t)] + B \cdot \frac{d}{dt}[y(t) - z(t)] + k_2 \cdot [y(t) - z(t)] = 0$$

en z(t):

$$k_2 \cdot [z(t) - y(t)] + B \cdot \frac{d}{dt}[z(t) - y(t)] + k_3 \cdot z(t) = 0$$

La determinación de la función de transferencia sigue los pasos expuestos con anterioridad.

2. Objetivo

Analizar el comportamiento de un sistema de segundo orden mecánico MFK, utilizando Función de Transferencia con respuesta a una entrada paso.

Obtener la función de transferencia en matlab y el diagrama en bloques en Simulink

3. Equipos y software Utilizados

Computadora

Software Matlab

Software Simulink

4. Desarrollo de la Práctica

```
%ANALISIS DE SISTEMA DE SEGUNDO ORDEN
%====================================
%======== SISTEMA MECANICO =========
%========== Fuerza M f k ===========
%====================================
%UTILIZANDO FUNCION DE TRANSFERENCIA
%Ingresando las constantes.
%Valor de k=1
k=1;
%Valor de f=1[m]
f=10;
%Valor de la Masa M=10[]
M=10;
%Ingresando Numerador
num1=1;
%Ingresando Denominador
den1=[M f k];
%Definiendo la Función de Transferencia
G1=tf(num1,den1);
%Mostrando la Función de Transferencia
display(G1)


Transfer function:
        1
-----------------
10 s^2 + 10 s + 1
```

```
%Respuesta a una entrada Paso
%Asignacion a una figura
figure(10)
%Aplicando una entrada paso
step(G1)
%Definiendo la cuadricula
grid
%Titulo
title('Respuesta a una Entrada Paso')
%Etiqueta de la Accisa
xlabel('Tiempo ')
%Etiqueta de la Ordenada
ylabel('Desplzamiento [m]')
```

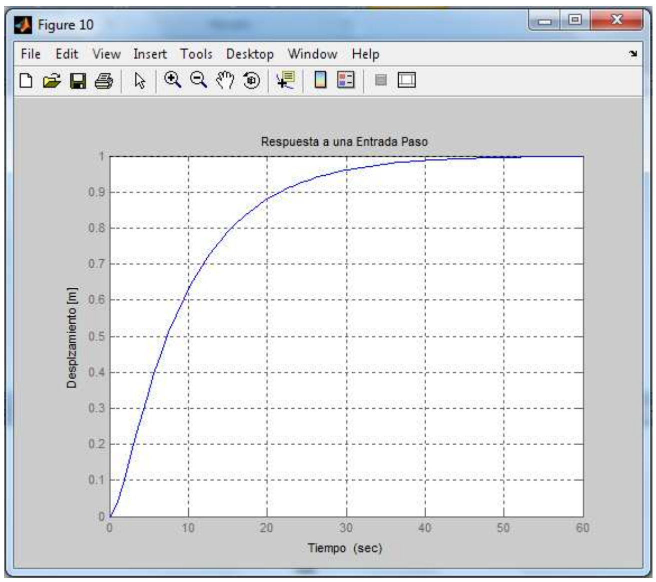


Figura # 3.18: Respuesta a una entrada paso
Fuente: **Autores**

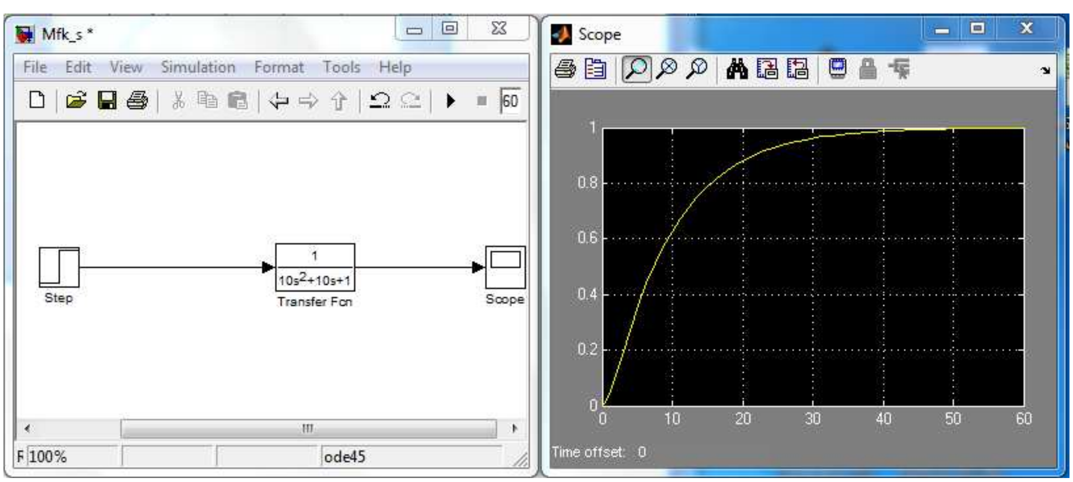


Figura # 3.19 : Diagrama de bloques en Simulink con su respectiva señal a una entrada paso
Fuente: Autores

5. Resultados y conclusiones

De la respuesta en el grafico se observa que el sistema es estable y el valor que alcanza en estado estacionario es de 1, consiguiendo estabilizarse en un tiempo de aproximadamente 50segundos.

Practica # 6: Sistemas de Segundo Orden: Sistema Mecánico Fuerza M, F, K, Respuesta a una entrada paso ante cambios de M:[1 100 1000]

1. Objetivo

Obtener el modelo y la función de transferencia del sistema Mecánico

Hallar por medio de la simulación la respuesta en el tiempo de este modelo de sistema ante cambios en la masa y así poder analizar su estabilidad.

2. Equipos y software utilizados

Computadora

Software Matlab

Software Simulink

3. Desarrollo de la practica

Lo primero que haremos es Obtener la función de transferencia en matlab y el diagrama en bloques en Simulink del sistema mecánico de los tres cambios de masa realizado.

```
%Respuesta a una entrada Paso ante
%Cambios de M:[1 100 1000]
Mv=[1 100 1000];
%Lazo para generar cambios de Mv
for i=1:3
    %Ingresando Denominador
    den=[Mv(i)  f k];
    %Definiendo la Funcion de Transferencia
    G=tf(num1,den);
    %Mostrando la Funcion de Transferencia
    display(G)
    %Respuesta a una entrada Paso
    %Asignacion a una figura
    figure(20+i)
    %Aplicando una entrada paso
    step(G)
    %Definiendo la cuadricula
    grid
    %Titulo
    title(['Respuesta a una Entrada Paso con',' M= ',num2str(Mv(i)),'
'])
    %Etiqueta de la Accisa
    xlabel('Tiempo ')
    %Etiqueta de la Ordenada
    ylabel('Desplzamiento [m]')
end
```

```
Transfer function:
      1
--------------
s^2 + 10 s + 1


Transfer function:
        1
------------------
100 s^2 + 10 s + 1


Transfer function:
         1
-------------------
1000 s^2 + 10 s + 1
```

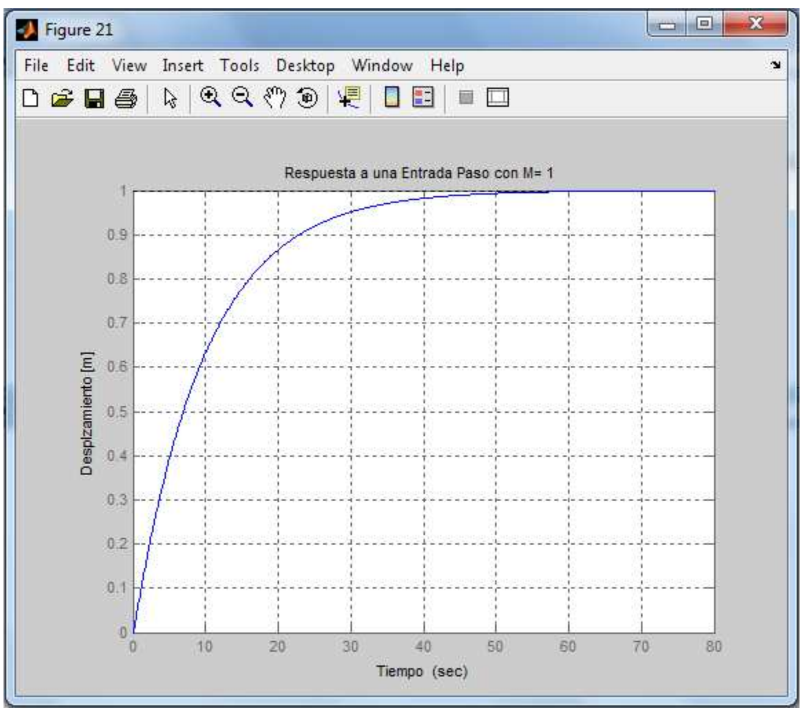


Figura # 3.20: Respuesta a una entrada paso con una masa de 1
Fuente: Autores

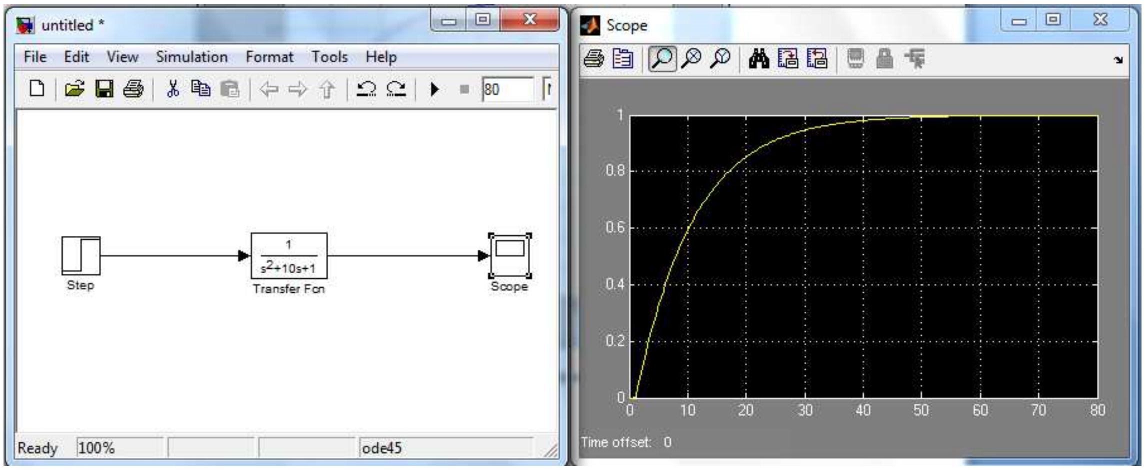


Figura # 3.21: Diagrama de bloques en Simulink con su respectiva señal a una entrada paso con una M= 1
Fuente: Autores

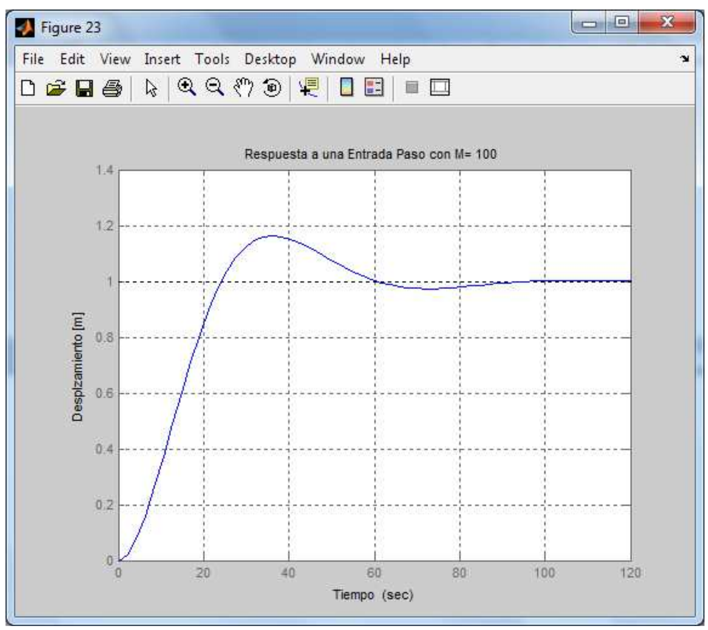


Figura # 3.22: Respuesta a una entrada paso con una masa de 100
Fuente: Autores

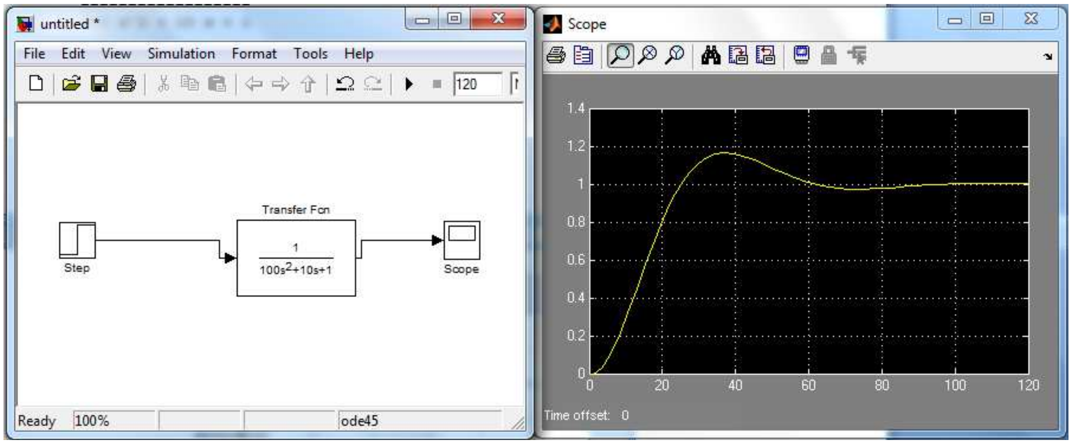


Figura # 3.23: Diagrama de bloques en Simulink con su respectiva señal a una entrada paso con una M= 1
Fuente: Autores

}

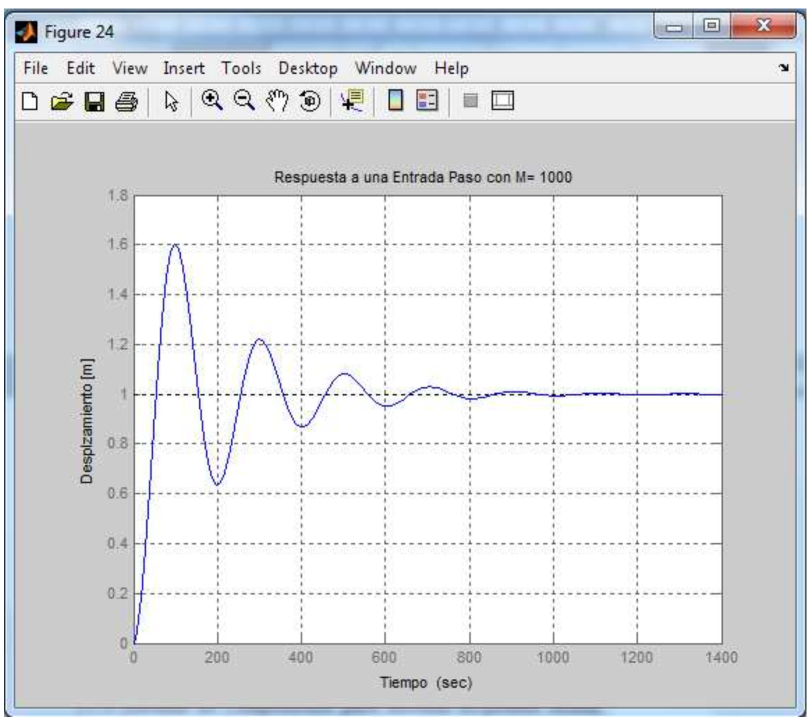


Figura # 3.24: Respuesta a una entrada paso con una masa de 1000
Fuente: Autores

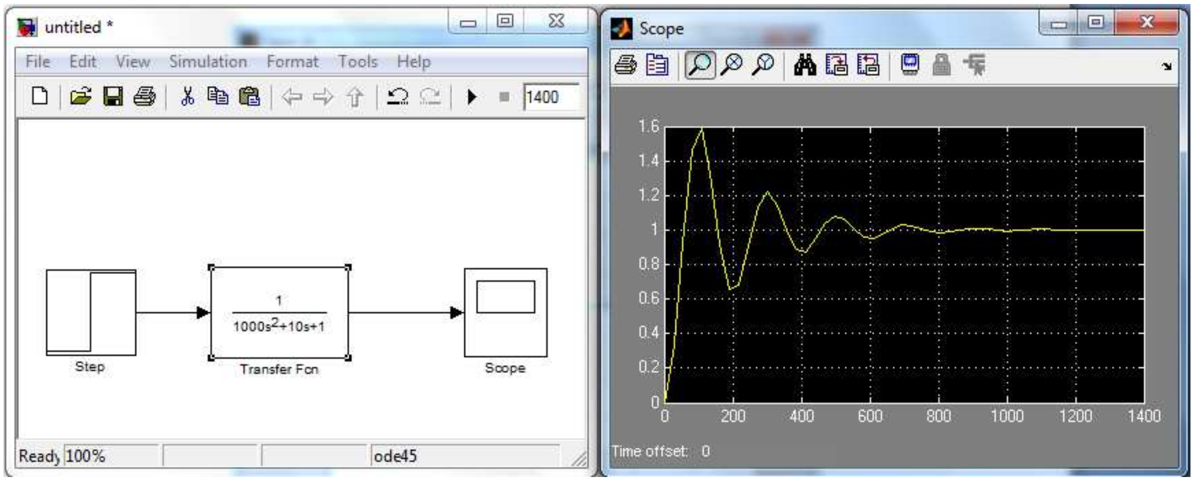


Figura # 3.25: Diagrama de bloques en Simulink con su respectiva señal a una entrada paso con una M= 100
Fuente: Autores

4. Resultados y conclusiones

El sistema responde con amortiguación a las señales de una entrada paso cambiando su masa en los diferentes valores anteriormente indicados.

Practica # 7: Sistema de Segundo Orden: Sistema Mecánico Fuerza M,F,K, respuesta a una entrada impulso.

1. Objetivo

Analizar el comportamiento de un sistema de segundo orden mecánico MFK, utilizando Función de Transferencia con respuesta a una entrada impulso.

Obtener la función de transferencia en matlab y el diagrama en bloques en Simulink

2. Equipos y software Utilizados

Computadora

Software Matlab

Software Simulink

3. Desarrollo de la Práctica

En Matlab la respuesta impulsional se puede obtener del mismo modo que la respuesta escalón paso, pero usando en este caso la función **impulse**, que tiene una sintaxis similar al comando **step**. , utilizado en la respuesta escalón-paso.

```
%Respuesta a una entrada Impulso
%Asignacion a una figura
```

```
figure(30)
%Aplicando una entrada Impulso
impulse(G1)
%Definiendo la cuadricula
grid
%Titulo
title('Respuesta a una Entrada Impulso')
%Etiqueta de la Accisa
xlabel('Tiempo ')
%Etiqueta de la Ordenada
ylabel('Desplzamiento [m]')
```

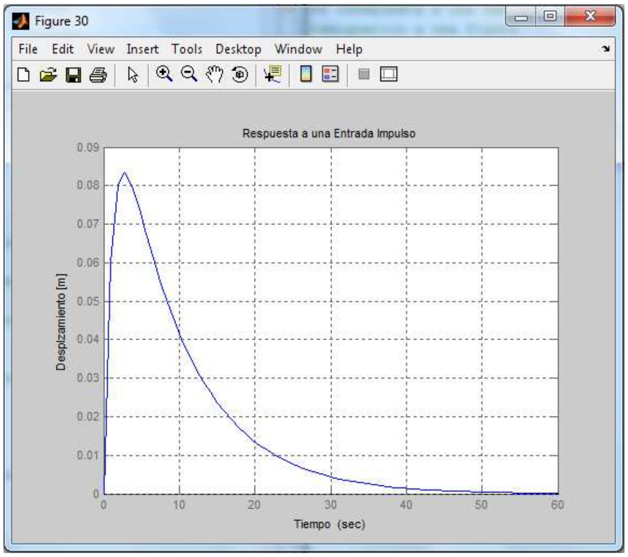


Figura # 3.26: Respuesta a una entrada impulso
Fuente: Autores

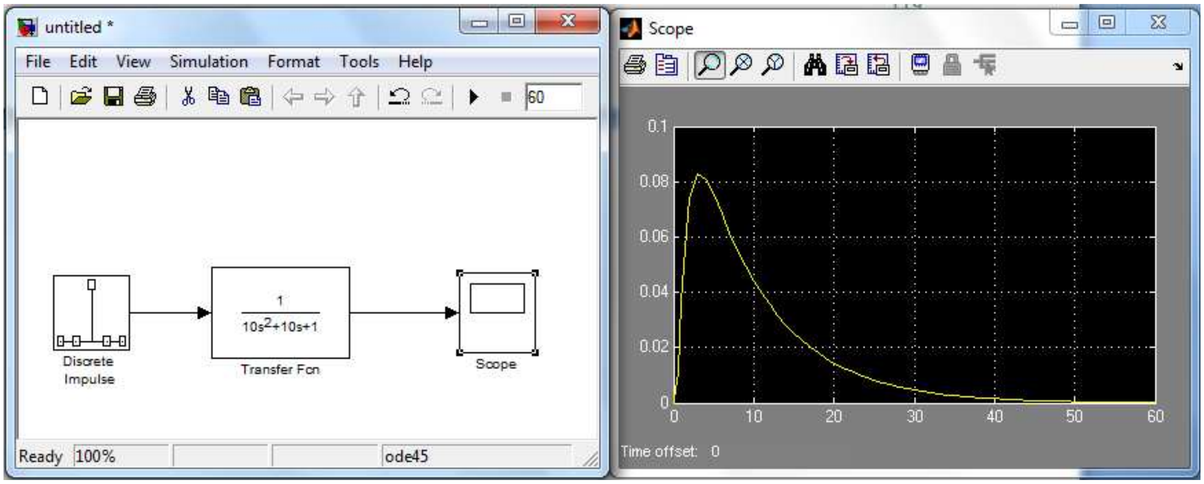


Figura # 3.27: Diagrama de bloques en Simulink con su respectiva señal a una entrada impulso
Fuente: Autores

4. Resultados y Conclusiones

En la gráfica se aprecia la respuesta del sistema de segundo orden ante una entrada impulso y se observa que efectivamente este responde ante un cambio en la entrada pero luego retorna al valor de salida cero. Esta respuesta nos indica que el sistema es estable.

Practica # 8: Sistema de Segundo Orden: Sistema Mecánico Fuerza M,F,K, respuesta a una entrada impulso ante Cambios de M=[1 100 1000]

1. Objetivo

Obtener el modelo y la función de transferencia del sistema Mecánico.
Hallar por medio de la simulación la respuesta en el tiempo de este modelo de sistema ante cambios en la masa y así poder analizar su estabilidad.

2. Equipos y software utilizados

Computadora

Software Matlab

Software Simulink

3. Desarrollo de la Práctica

Lo primero que haremos es Obtener la función de transferencia en matlab y el diagrama en bloques en Simulink del sistema mecánico de los tres cambios de masa realizado.

```
%Respuesta a una entrada Impulso ante
%Cambios de M=[1 100 1000]
Mv=[1 100 1000];
%Lazo para generar cambios de Cv
for i=1:3
    %Ingresando Denominador
    den=[Mv(i)  f k];
    %Definiendo la Funcion de Transferencia
    G=tf(num1,den);
    %Mostrando la Funcion de Transferencia
    display(G)
    %Respuesta a una entrada Impulso
    %Asignacion a una figura
    figure(40+i)
    %Aplicando una entrada Impulso
    Impulse(G)
```

```
    %Definiendo la cuadricula
    grid
    %Titulo
    title(['Respuesta a una Entrada Impulso con',' M=
',num2str(Mv(i)),' '])
    %Etiqueta de la Accisa
    xlabel('Tiempo ')
    %Etiqueta de la Ordenada
    ylabel('Desplzamiento [m]')
end
```

```
Transfer function:
      1
--------------
s^2 + 10 s + 1


Transfer function:
        1
------------------
100 s^2 + 10 s + 1


Transfer function:
         1
-------------------
1000 s^2 + 10 s + 1
```

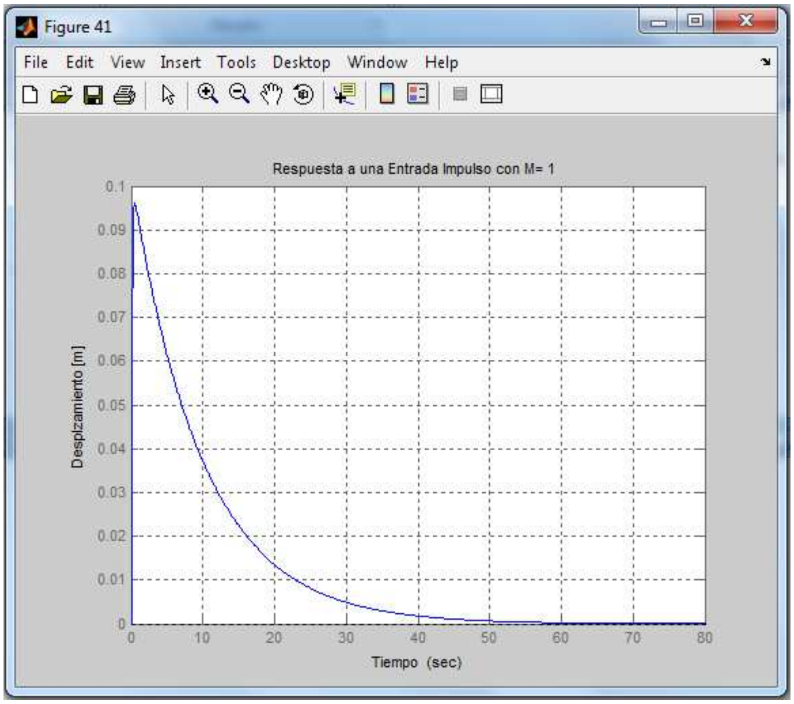


Figura # 3.28: Respuesta a una entrada impulso con masa 1
Fuente: Autores

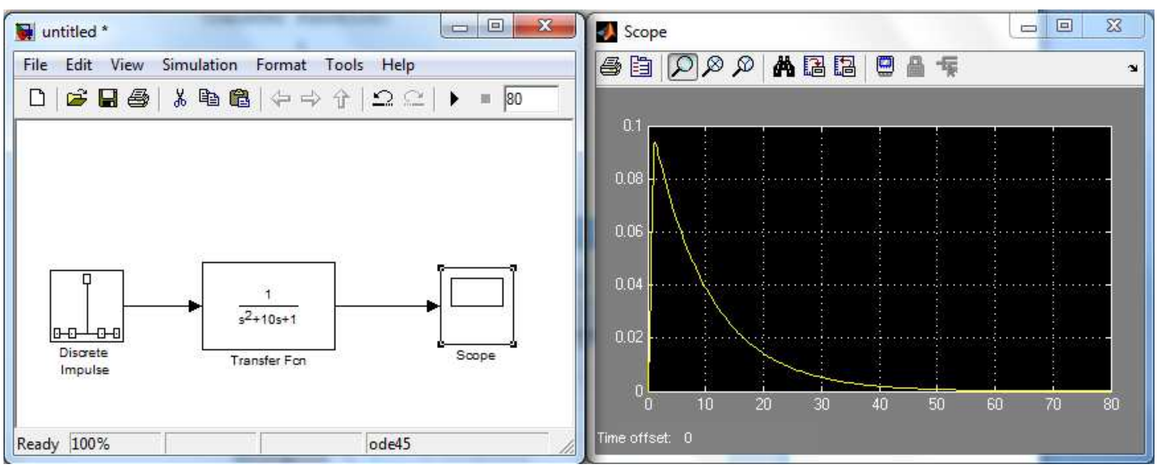


Figura # 3.29: Diagrama de bloques en Simulink con su respectiva señal a una entrada impulso con una M=1
Fuente: Autores

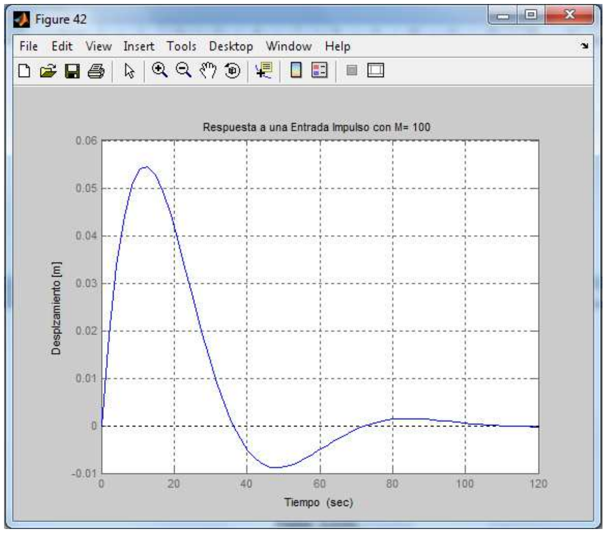


Figura # 3.30: Respuesta a una entrada impulso con masa 100
Fuente: Autores

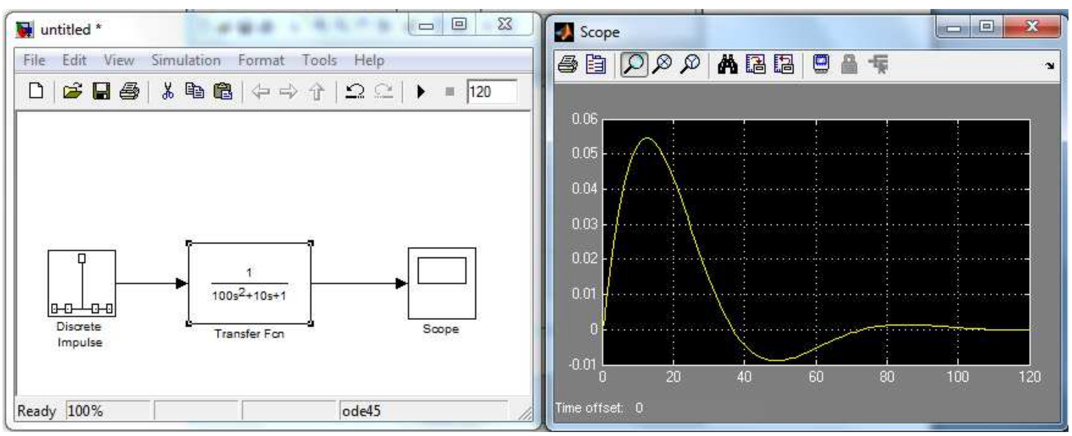


Figura # 3.31: Diagrama de bloques en Simulink con su respectiva señal a una entrada impulso con una M=100
Fuente: Autores

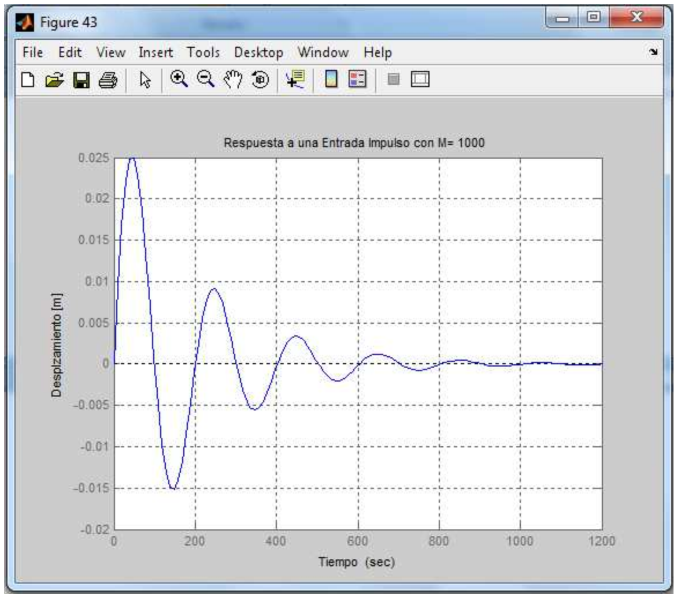


Figura # 3.32: Respuesta a una entrada impulso con masa 1000
Fuente: Autores

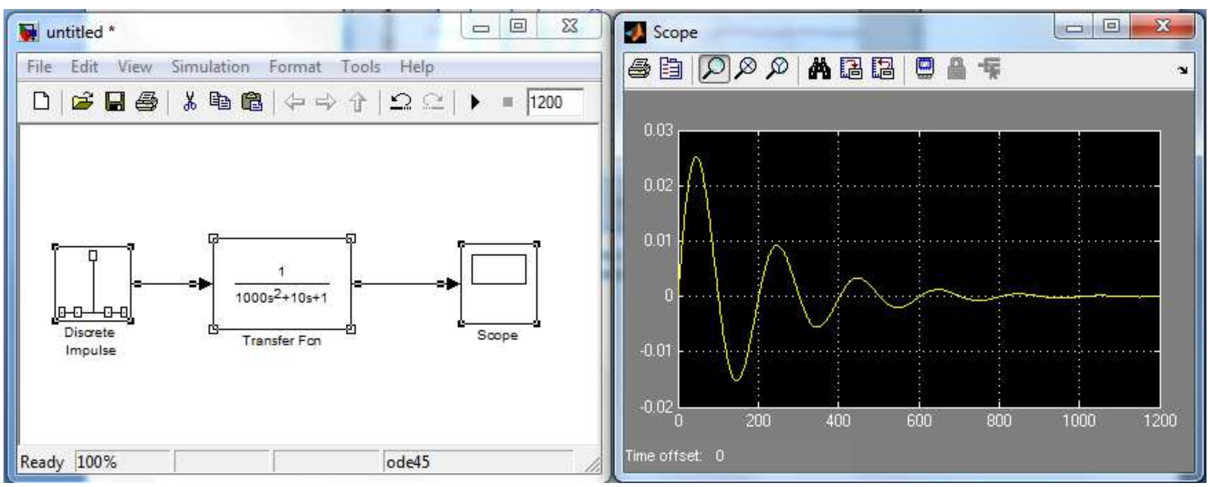


Figura # 3.33: Diagrama de bloques en Simulink con su respectiva señal a una entrada impulso con una M=1000
Fuente: Autores

4. Resultados y conclusiones

En las gráficas se aprecia las respuestas del sistema de segundo orden ante una entrada impulso y se observa que efectivamente este responde ante un cambio en la entrada, con la única variación que al cambiar las masas aparecen algunos amortiguamientos, pero luego retorna al valor de salida cero. Esta respuesta nos indica que el sistema es estable

3.4. Análisis y Resultados

Por medio de estas prácticas de laboratorio podremos lograr un enorme avance a la forma de cómo los estudiantes pueden aprender a analizar sistemas de control de primer y segundo orden usando herramientas interactivas en tiempo real y contrastar de manera acertada lo teórico explicado con la practica en el laboratorio.

Utilizando los comandos adecuados, pudimos aprender a modelar de una manera práctica y clara la función de transferencia de cualquier sistema.

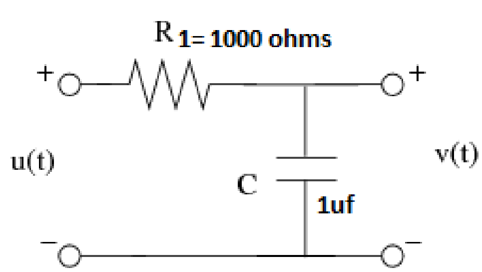
3.5. Propuesta (Guía de prácticas)

Practica # 1: Sistemas de Primer Orden: Circuito Fuente RC en Serie, Respuesta a una entrada escalón.

1. **Objetivo**

Verificar el comportamiento del sistema a una entrada del tipo escalón o paso, comprobar si el tiempo de establecimiento de la señal del circuito, es decir, cuanto se tarda el sistema en alcanzar su estado estacionario.

Obtener la función de transferencia en matlab y el diagrama en bloques en Simulink del filtro RC del siguiente circuito.



2. **Equipos y software Utilizados**

   Computadora

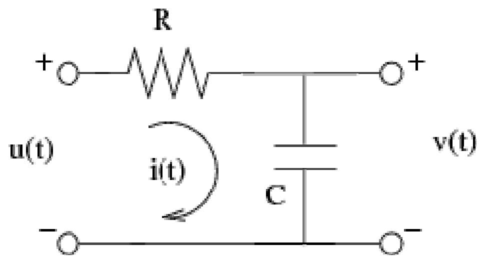
   Software Matlab

   Software Simulink

3. Desarrollo de la practica

Para obtener una función de transferencia de este circuito se puede realizar el siguiente cálculo matemático manual con las siguientes expresiones:

Considerando que i(t) es la intensidad que recorre la única malla del circuito, tal y como se observa en la siguiente figura:



La caída de tensión en la misma u(t) y la que se produce en los bornes del condensador se pueden expresar de la forma:

$$\begin{aligned} u(t) &= Ri(t) + v(t) \\ v(t) &= \frac{1}{C}\int i(\tau)d\tau \end{aligned}$$

Despejando la intensidad de la segunda ecuación se tiene que la misma es i(t) = C$\frac{dv(t)}{dt}$, que sustituyendo en la primera permite obtener la ecuación diferencial del sistema en función únicamente de las variables de entrada u(t) y de salida v(t):

$$u(t) = RC\frac{dv(t)}{dt} + v(t)$$

Finalmente, aplicando la transformada de Laplace a la anterior ecuación (considerando las condiciones iniciales nulas) y despejando se tiene la función de transferencia del circuito:

$$U(s) = RCsV(s) + V(s) \qquad \Longrightarrow \qquad \frac{V(s)}{U(s)} = \frac{1}{1+RCs}$$

En matlab se obvia todo este proceso y se introducen los datos de la siguiente manera:

```
%ANALISIS DE SISTEMA DE PRIMER ORDEN
%====================================
%======== SISTEMA ELECTRICO ========
%========== Fuente R  C ============
%====================================
%UTILIZANDO FUNCION DE TRANSFERENCIA
%Ingresando las constantes.
%Valor de la Resistencia R=1[KOhmio]
```

```
R=1000;
%Valor del Capacitor C=1[uF]
C=1*10^(-6);
%Ingresando Numerador
num1=1;
%Ingresando Denominador
den1=[R*C 1];
%Definiendo la Funcion de Transferencia
G1=tf(num1,den1);
%Mostrando la Funcion de Transferencia
display(G1)
-------------------------------------------

Transfer function:
     1
-----------
0.001s + 1

%====================================
%Respuesta a una entrada Paso
%Asignacion a una figura
figure(10)
%Aplicando una entrada paso
step(G1)
%Definiendo la cuadricula
grid
%Titulo
title('Respuesta a una Entrada Paso')
%Etiqueta de la Accisa
xlabel('Tiempo ')
%Etiqueta de la Ordenada
ylabel('Voltaje [V]')
```

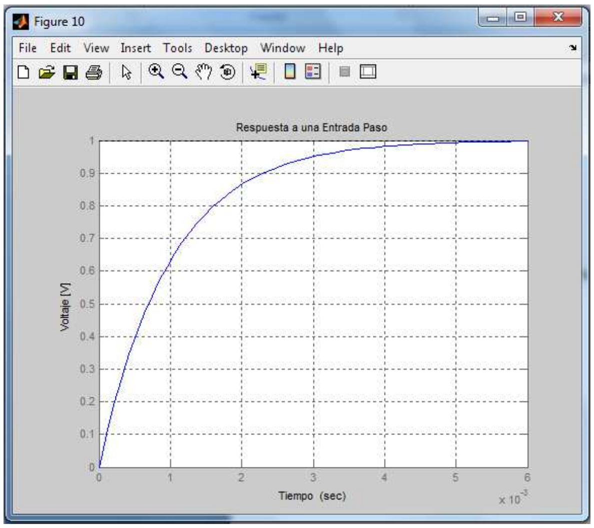


Figura # 3.34: Respuesta a una entrada paso en Matlab
Fuente: Autores

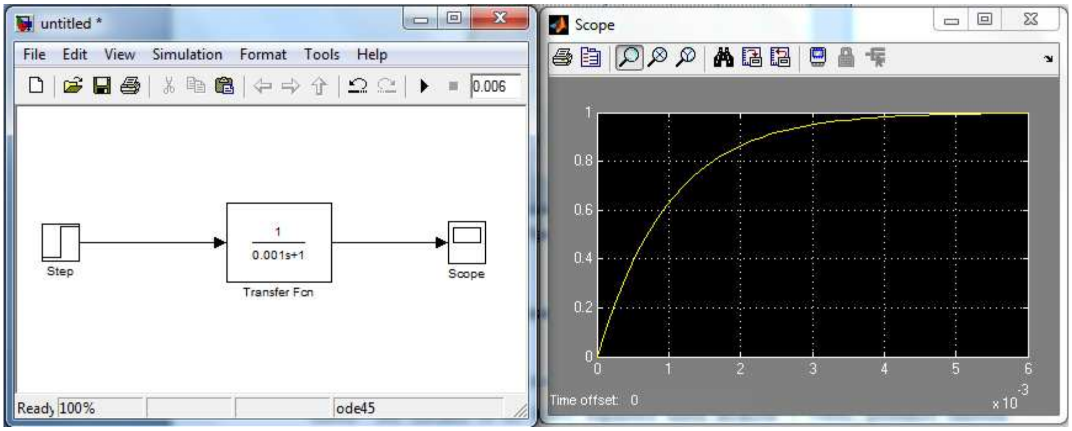


Figura # 3.35: Diagrama de bloques de la función de transferencia en Simulink del filtro RC
Fuente: Autores

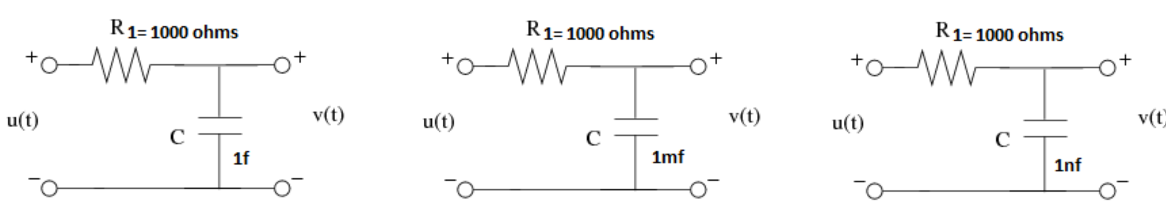
4. Resultados y conclusiones

En la gráfica el resultado nos indica que el tiempo de establecimiento de la señal de salida del circuito es de 0,001 segundos hasta alcanzar 1 voltio, podemos también concluir que lo estudiado en lo teórico con respecto al análisis en el dominio del tiempo en los sistemas de primer orden si se cumple.

Practica # 2: Sistemas de Primer Orden: Circuito Fuente RC en Serie: Respuesta a una entrada Escalón o Paso ante Cambios de C:[1f; 1mf ;1nf]

1. Objetivo

Lo que se quiere lograr en esta práctica es observar que comportamiento tiene el tiempo y el voltaje al cambiar el valor de la capacitancia en el circuito.



2. Equipos y software Utilizados

Computadora

Software Matlab

Software Simulink

3. Desarrollo de la practica

Lo primero que haremos es Obtener la función de transferencia en matlab y el diagrama en bloques en Simulink del filtro RC de los tres circuitos de la siguiente manera:

```
%Respuesta a una entrada Paso ante
%Cambios de C:[1F 1mF 1nF]
Cv=[1 1*10^(-3) 1*10^(-9)];
%Lazo para generar cambios de Cv
for i=1:3
    %Ingresando Denominador
    den=[R*Cv(i) 1];
    %Definiendo la Funcion de Transferencia
    G=tf(num1,den);
    %Mostrando la Funcion de Transferencia
    display(G)
    %Respuesta a una entrada Paso
    %Asignacion a una figura
    figure(20+i)
    %Aplicando una entrada paso
    step(G)
    %Definiendo la cuadricula
    grid
    %Titulo
    title(['Respuesta a una Entrada Paso con',' C=
',num2str(Cv(i)),'F'])
    %Etiqueta de la Accisa
    xlabel('Tiempo ')
    %Etiqueta de la Ordenada
    ylabel('Voltaje [V]')
end

Transfer function:
    1
----------
1000 s + 1


Transfer function:
  1
-----
s + 1


Transfer function:
     1
------------
1e-006 s + 1
```

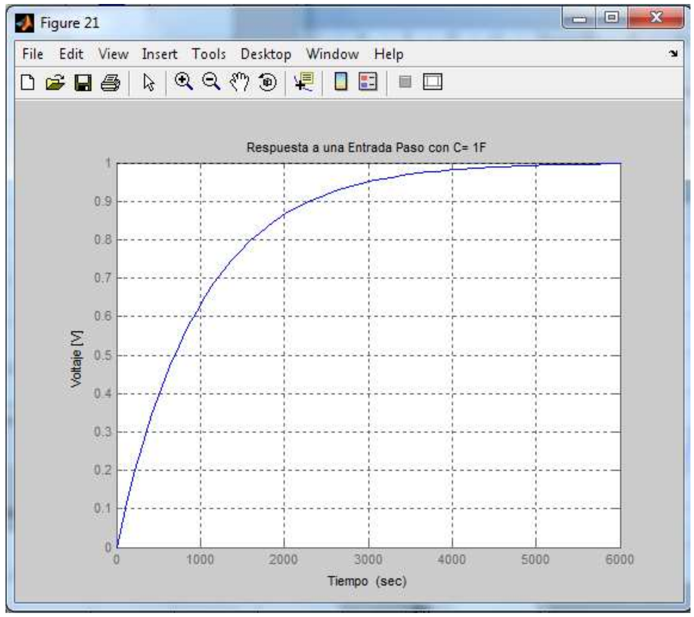


Figura # 3.36: Respuesta a una entrada paso con un capacitor de 1f
Fuente: Autores

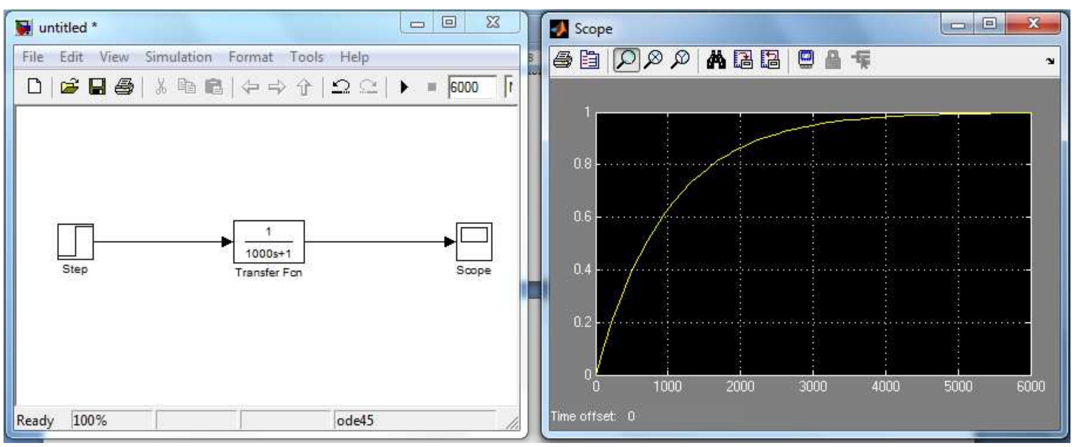


Figura # 3.37: Diagrama de bloques en Simulink con su respectiva señal a una entrada paso con un capacitor de 1f
Fuente: Autores

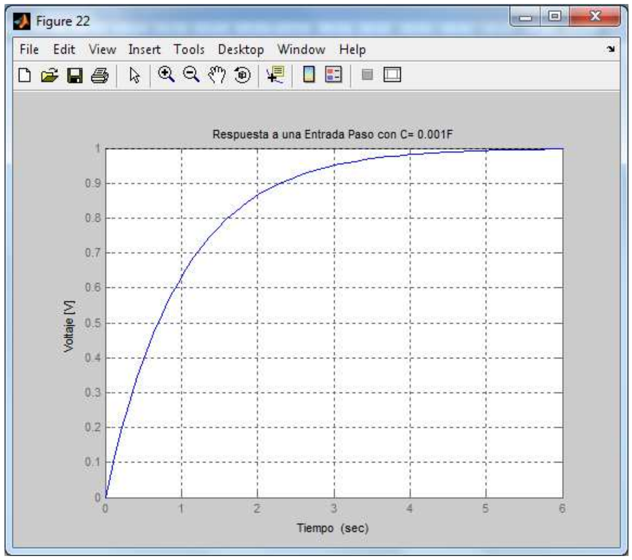


Figura # 3.38: Respuesta a una entrada paso con un capacitor de 1mf
Fuente: Autores

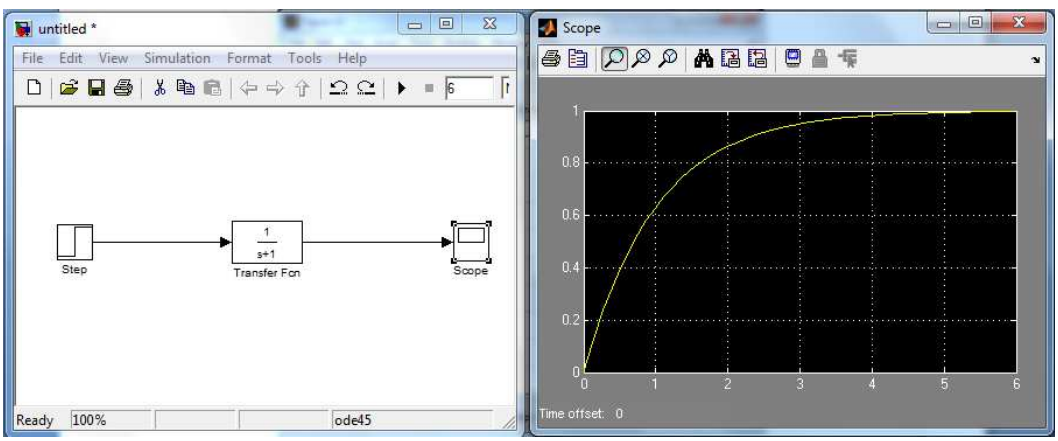


Figura # 3.39: Diagrama en bloques en Simulink con su respectiva señal a una entrada paso con un capacitor de 1mf
Fuente: Autores

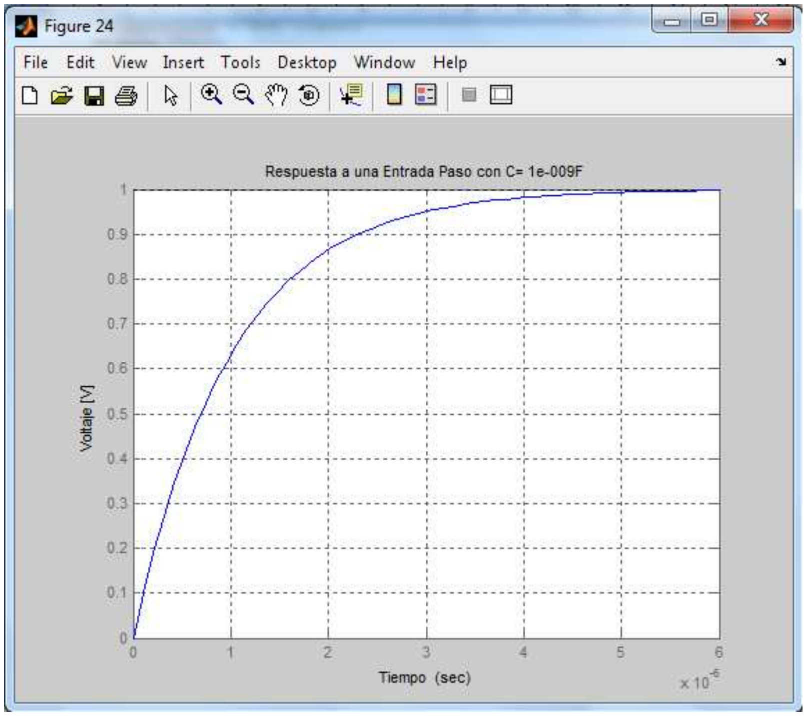


Figura # 3.40: Respuesta a una entrada paso con un capacitor de 1nf
Fuente: Autores

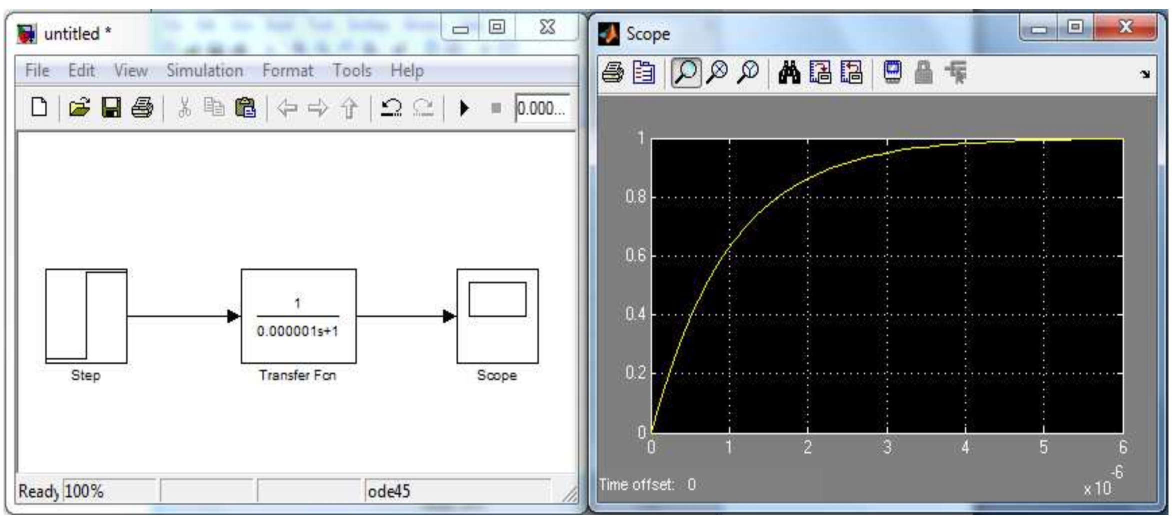


Figura # 3.41: Diagrama en bloques en Simulink con su respectiva señal a una entrada paso con un capacitor de 1nf
Fuente: Autores

4. **Resultados y conclusiones**

La grafica de los circuitos nos muestra que la fuente de voltaje de un voltio y la resistencia de 1000omhs son iguales para los tres, mientras que el capacitor se lo cambia con uno de 1f, 1mf y 1uf, esto nos arroja diferentes funciones de transferencia y deferentes comportamientos del circuito en el tiempo, podemos observar que en la figura3.37 el tiempo que pasa para estabilizarse el circuito en su salida es de 6000 segundos hasta llegar a un voltio, con un capacitor de 1f, la figura 3.39 nos indica que con un capacitor de 1mf el tiempo que pasa para estabilizarse el circuito en su salida es de 6 segundos , hasta llegar a un voltio , mientras que en la figura 3.40 con un capacitor de 1uf el tiempo que pasa para estabilizarse el circuito en su salida de 0,000006 segundos, esto quiere decir que manteniendo el voltaje y la resistencia constantes y el capacitor variable, el tiempo de disparo o carga en el circuito para estabilizarse varía, con lo que podemos concluir que a mayor capacitancia mayor el tiempo de disparo o carga del capacitor en el circuito para estabilizarse, mientras que a menor capacitancia , menor el tiempo de disparo o carga del capacitor en el circuito para estabilizarse; esto quiere decir que en un circuito RC en serie con un voltaje y una resistencia constantes su tiempo de carga de capacitancia o de estabilización en el tiempo es directamente proporcional al valor del capacitor.

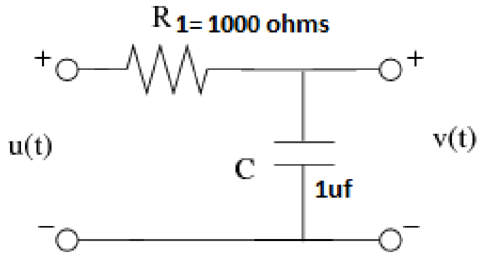
Practica # 3: Sistemas de Primer Orden: Circuito Fuente RC en Serie: Respuesta a una entrada Impulso.

La transformada de Laplace de un impulso unitario es igual a la unidad por lo que la transformada de la respuesta impulsional es igual a la expresión de la función de transferencia.

La respuesta impulsional de un sistema es pues la transformada inversa de la función de transferencia, en el caso de los sistemas de primer orden.

1. **Objetivo**

Verificar el comportamiento del sistema a una entrada del tipo impulso.
Obtener la función de transferencia en matlab y el diagrama en bloques en Simulink del circuito.



2. Equipos y software Utilizados

Computadora

Software Matlab

Software Simulink

3. Desarrollo de la practica

En Matlab la respuesta impulsional se puede obtener del mismo modo que la respuesta escalón paso, pero usando en este caso la función **impulse**, que tiene una sintaxis similar al comando **step. ,** utilizado en la respuesta escalón-paso

```
%Respuesta a una entrada Impulso
%Asignacion a una figura
figure(30)
%Aplicando una entrada Impulso
impulse(G1)
%Definiendo la cuadricula
grid
%Titulo
title('Respuesta a una Entrada Impulso')
%Etiqueta de la Accisa
xlabel('Tiempo ')
%Etiqueta de la Ordenada
ylabel('Voltaje [V]')
Transfer function:
     1
-----------
0.02  s + 1
```

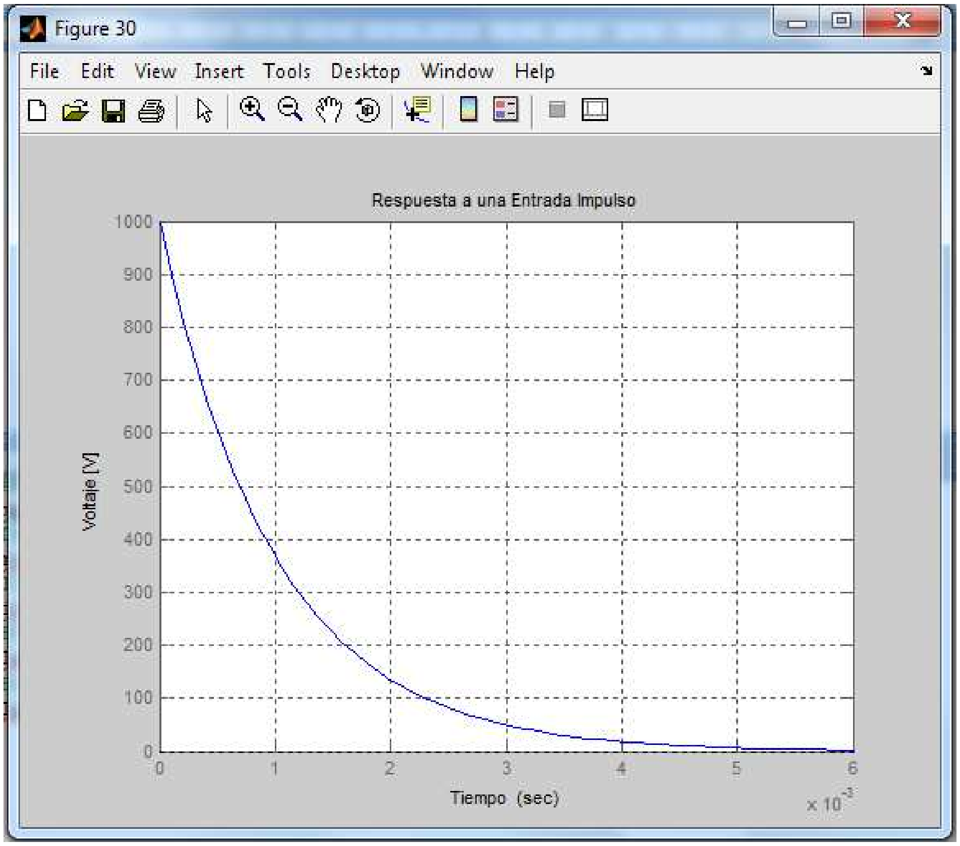


Figura # 3.42: Respuesta a una entrada impulso
Fuente: autores

4. Resultados y conclusiones

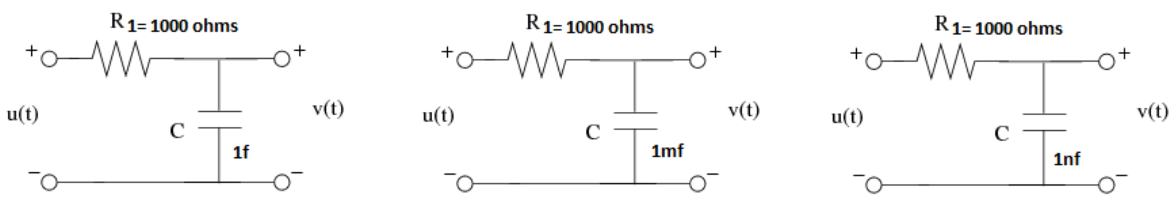
Un impulso es una señal que tiene una magnitud infinita y un ancho infinitesimalmente estrecho con un área uno, centrado en cero y puede ser representado como una suma infinita de sinusoides que incluye todas las frecuencias posibles.

En este caso el impulso se utiliza como entrada a un sistema, la salida se conoce como la respuesta impulsional. La respuesta impulsional define el sistema, ya que todas las frecuencias posibles se representan en la entrada, con lo que podemos concluir que el sistema es estable por que vuelve a cero en un tiempo estimado.

Practica # 4: Sistemas de Primer Orden: Circuito Fuente RC en Serie: Respuesta a una entrada Impulso ante cambios de C: [1f; 1mf; 1nf].

1. **Objetivo**

Lo que se quiere lograr en esta práctica es observar que comportamiento tiene el tiempo y el voltaje al cambiar el valor de la capacitancia en el circuito.



2. **Equipos y software Utilizados**

   Computadora

   Software Matlab

   Software Simulink

3. **Desarrollo de la Práctica**

Lo primero que haremos es Obtener la función de transferencia en matlab y el diagrama en bloques en Simulink del filtro RC de los tres circuitos de la siguiente manera:

```
%Respuesta a una entrada Impulso ante
%Cambios de C:[1F 1mF 1nF]
Cv=[1 1*10^(-3) 1*10^(-9)];
%Lazo para generar cambios de Cv
for i=1:3
    %Ingresando Denominador
    den=[R*Cv(i) 1];
    %Definiendo la Funcion de Transferencia
    G=tf(num1,den);
    %Mostrando la Funcion de Transferencia
    display(G)
    %Respuesta a una entrada Impulso
    %Asignacion a una figura
    figure(40+i)
    %Aplicando una entrada Impulso
    Impulse(G)
    %Definiendo la cuadricula
    grid
    %Titulo
    title(['Respuesta a una Entrada Impulso con',' C=
',num2str(Cv(i)),'F'])
    %Etiqueta de la Accisa
    xlabel('Tiempo ')
    %Etiqueta de la Ordenada
    ylabel('Voltaje [V]')
```

```
end
Transfer function:
    1
----------
1000 s + 1


Transfer function:
  1
-----
s + 1


Transfer function:
     1
------------
1e-006 s + 1
```

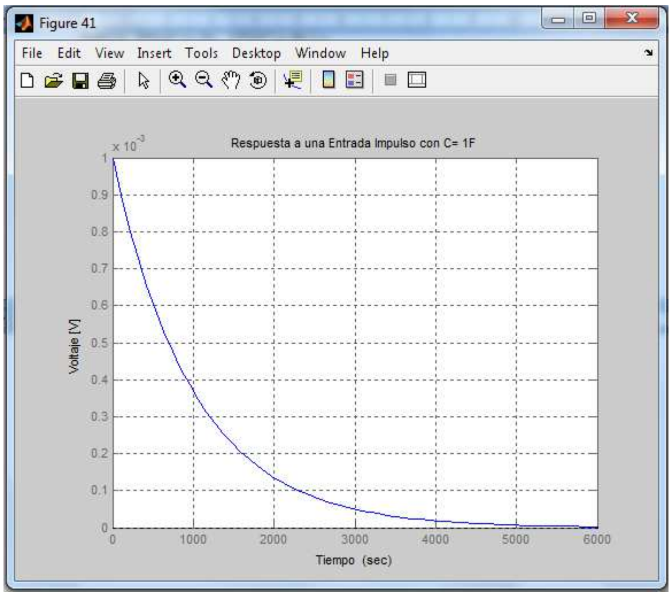


Figura # 3.43: Respuesta a una entrada impulso con un capacitor de 1f
Fuente: Autores

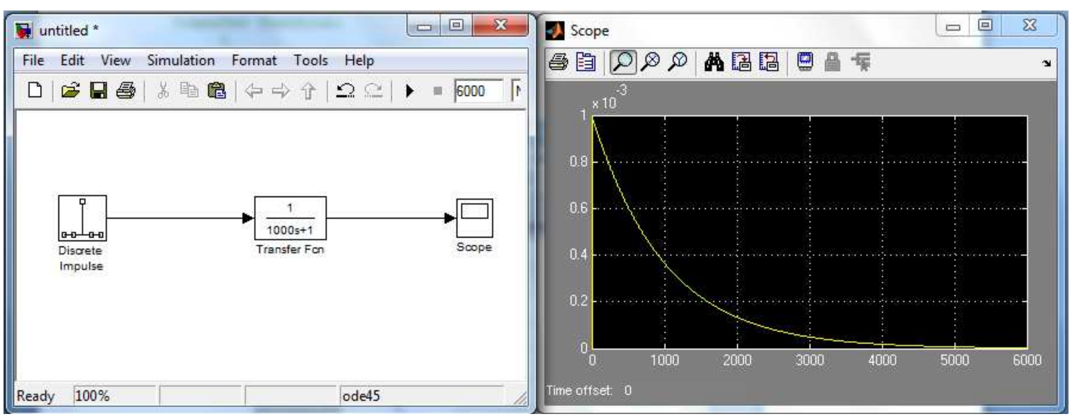


Figura # 3.44: Diagrama de bloques en Simulink con su respectiva señal a una entrada impulso con un capacitor de 1f
Fuente: Autores

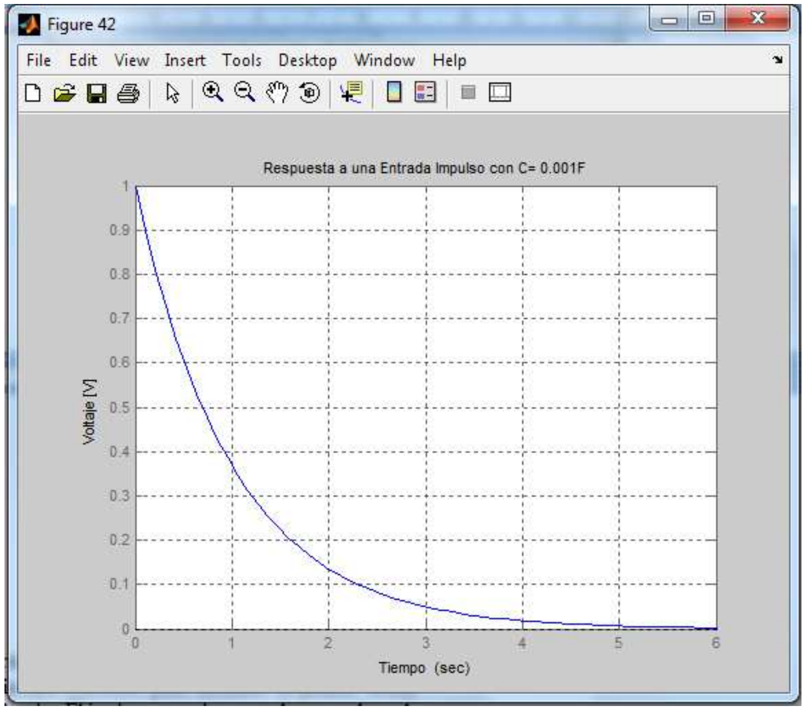


Figura # 3.45: Respuesta a una entrada impulso con un capacitor de 1mf
Fuente: Autores

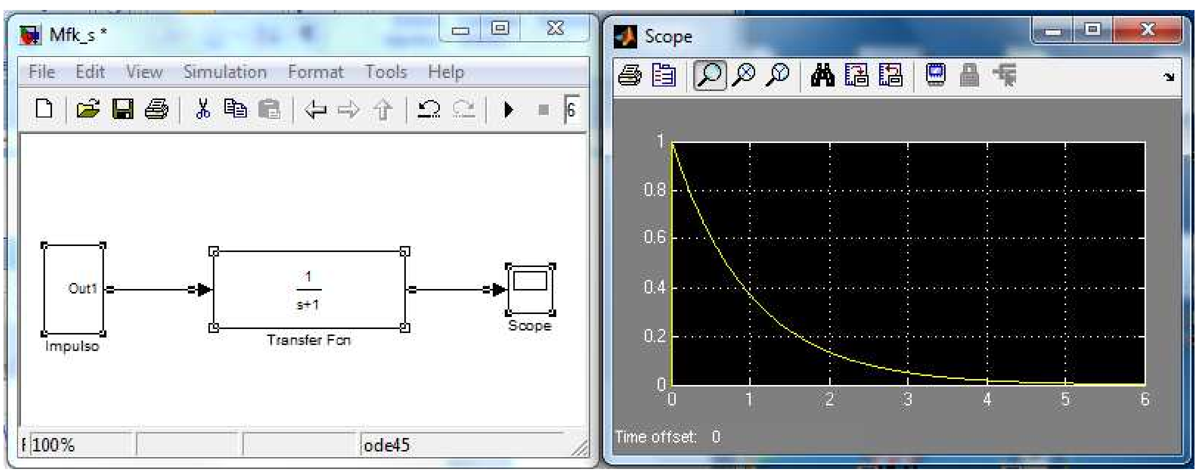


Figura # 3.46: Diagrama de bloques en Simulink con su respectiva señal a una entrada impulso con un capacitor de 1f
Fuente**: Autores**

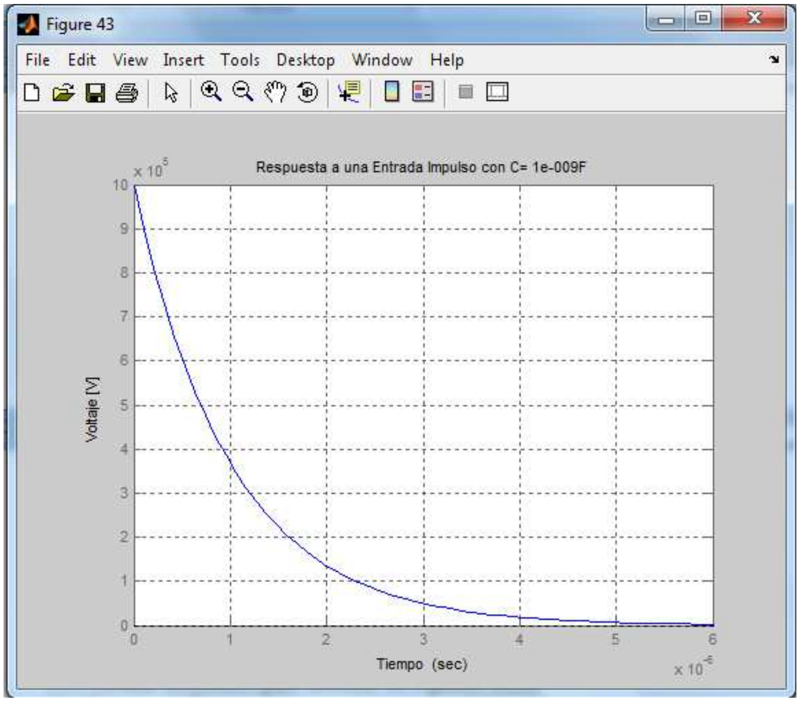


Figura # 3.47: Respuesta a una entrada impulso con un capacitor de 1nf
Fuente: Autores

4. Resultados y conclusiones

La respuesta a una entrada Impulso ante cambios de C:[1F 1mF 1nF] se la conoce como respuesta impulsional, en las gráficas nos podemos dar cuenta que aunque cambien los valores de los capacitores, el tiempo de estabilización del sistema también cambia, no importa la cantidad de tiempo que demore lo importante es que el sistema responde establemente y que tiende su respuesta a cero.

Practica # 5: Sistemas de Segundo Orden: Sistema Mecánico Fuerza M, F, K, Respuesta a una entrada paso.

1. Introducción

Los sistemas mecánicos son una parte fundamental de la vida común, ya que cualquier cuerpo físico se comporta como tal. En general los sistemas mecánicos son gobernados por la segunda ley de Newton, la cual establece para sistemas mecánicos de traslación que "la suma de fuerzas en un sistema, sean estas aplicadas o reactivas, igualan a la masa por la aceleración a que está sometida dicha masa".

$$\sum f = ma$$

Cuando se trata de sistemas mecánicos de rotación la segunda ley de Newton declara que "la suma de torques es igual al momento de inercia multiplicado por la aceleración angular".

En cualquiera de los casos anteriores se tiene diferentes elementos cuyo acoplamiento conforma al sistema mecánico completo, pudiendo además interactuar entre cada caso. A continuación se describen las generalidades de ambos tipos de sistemas mecánicos.

$$\sum T = J\alpha$$

Sistemas mecánicos de traslación.

Los sistemas mecánicos de traslación están integrados por el conjunto de elementos básicos resumidos en la siguiente tabla6.

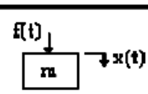
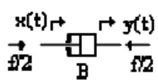
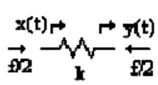
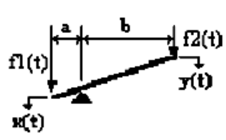
ELEMENTO	SIMBOLO	ECUACIÓN DE EQUILIBRIO	UNIDADES
MASA		$f_m = m \cdot \frac{d^2}{dt^2} x(t)$	[Kg]. ó [N.s/m]
AMORTIGUADOR		$f_B = B \cdot \frac{d}{dt}[x(t) - y(t)]$	coeficiente de fricción viscosa B = [N.seg/m]
RESORTE		$f_k = k \cdot [x(t) - y(t)]$	módulo de elasticidad k= [N/m]
PALANCA		$y = \frac{b}{a+b} \cdot x$ $f_2 = \frac{a}{a+b} \cdot f_1$	adimensional

Tabla 6: Elementos mecánicos de Translación

En este caso las variables involucradas son desplazamientos, velocidades, aceleraciones y fuerzas. La disposición que guardan estos elementos entre sí da lugar a dos configuraciones denominadas arreglos mecánicos en serie y arreglos mecánicos en paralelo.

Elementos mecánicos en serie.

En un elemento mecánico en serie, la fuerza aplicada f(t) es igual a la suma de las fuerzas actuantes en cada elemento y todos los elementos tienen el mismo desplazamiento. Tal como se muestra en la figura 3.48.

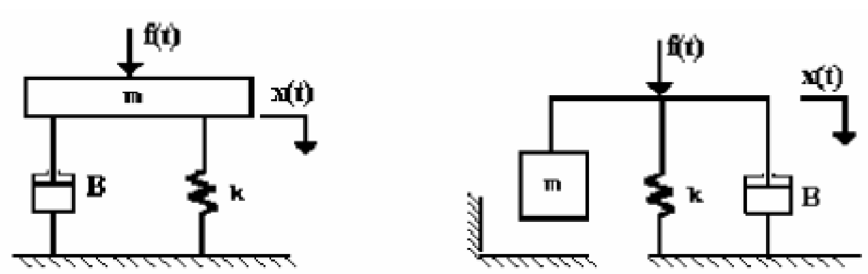


Figura # 3.48: Elementos mecánicos en Serie

La ecuación de equilibrio para el arreglo de la figura 3.48 es:

$$f(t) = m \cdot \frac{d^2}{dt^2}x(t) + B \cdot \frac{d}{dt}x(t) + k \cdot x(t)$$

Y su transformada de Laplace considerando condiciones iniciales iguales a cero es:

$$F(s) = (ms^2 + Bs + k)X(s)$$

Donde la impedancia mecánica es:

$$Z(s) = ms^2 + Bs + k$$

Elementos Mecánicos en Paralelo.

En este tipo de arreglo la fuerza aplicada f(t) se transmite a través de todos los elementos. Además, la deformación o corrimiento total es la suma de los desplazamientos de cada elemento. La figura 3.49 muestra un ejemplo de este tipo de

$$X(s) = \frac{F(s)}{k} + \frac{F(s)}{B_1 s} + \frac{F(s)}{B_2 s}$$

Arreglo en el que considerando las ecuaciones ya transformadas el desplazamiento total está dado por:

La relación fuerza a desplazamiento queda como:

$$F(s)=\frac{1}{1/k+1/B_1s+1/B_2s}X(s)$$

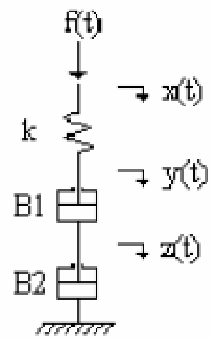


Figura # 3.49: Arreglo mecánico en paralelo

Donde la impedancia mecánica es:

$$Z(s)=\frac{1}{1/k+1/B_1s+1/B_2s}$$

Un comentario importante respecto al comportamiento de una masa es que esta no puede estar en paralelo con otros elementos a menos que sea el último de los elementos. Para ilustrar lo anterior veamos que en la figura 3.50, la masa, al ser el último elemento, participa como si estuviera en paralelo dando la ecuación que relaciona la fuerza con el desplazamiento de la forma:

$$F(s)=\frac{1}{1/k+1/Bs+1/ms}X(s)$$

Mientras que en la figura 69-b al estar la masa colocada como un elemento intermedio, y tener el mismo desplazamiento y(t) en la parte superior e inferior, la sitúa en serie tanto con k1 como con k2 y B respecto al desplazamiento y(t) mientras que no tiene nada que ver con los desplazamientos x(t) y z(t) que afectan al comportamiento de los elementos k1 y k2-B respectivamente.

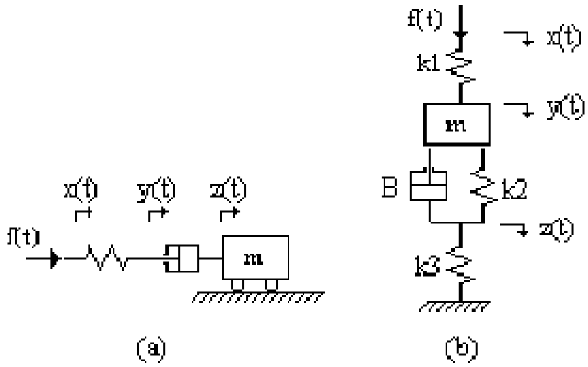


Figura # 3.50 a : Masa como elemento en paralelo b: Masa como elemento en serie

Para el caso de la figura 3.50 las ecuaciones de equilibrio en cada desplazamiento son:

En x(t):

$$f(t) = k_1 \cdot [x(t) - y(t)]$$

En y(t):

$$m \cdot \frac{d^2}{dt^2} y(t) + k_1 \cdot [y(t) - x(t)] + B \cdot \frac{d}{dt}[y(t) - z(t)] + k_2 \cdot [y(t) - z(t)] = 0$$

En z(t):

$$k_2 \cdot [z(t) - y(t)] + B \cdot \frac{d}{dt}[z(t) - y(t)] + k_3 \cdot z(t) = 0$$

La determinación de la función de transferencia sigue los pasos expuestos con anterioridad.

2. Objetivo

Analizar el comportamiento de un sistema de segundo orden mecánico MFK, utilizando Función de Transferencia con respuesta a una entrada paso.

Obtener la función de transferencia en matlab y el diagrama en bloques en Simulink

3. Equipos y software Utilizados

Computadora

Software Matlab

Software Simulink

4. Desarrollo de la Práctica

```
%ANALISIS DE SISTEMA DE SEGUNDO ORDEN
%====================================
%======== SISTEMA MECANICO =========
%========== Fuerza M f k ===========
%====================================
%UTILIZANDO FUNCION DE TRANSFERENCIA
%Ingresando las constantes.
%Valor de k=1
k=1;
%Valor de f=1[m]
f=10;
%Valor de la Masa M=10[]
M=10;
%Ingresando Numerador
num1=1;
%Ingresando Denominador
den1=[M f k];
%Definiendo la Funcion de Transferencia
G1=tf(num1,den1);
%Mostrando la Funcion de Transferencia
display(G1)



Transfer function:
        1
-----------------
10 s^2 + 10 s + 1


%Respuesta a una entrada Paso
%Asignacion a una figura
figure(10)
%Aplicando una entrada paso
step(G1)
%Definiendo la cuadricula
grid
%Titulo
title('Respuesta a una Entrada Paso')
%Etiqueta de la Accisa
xlabel('Tiempo ')
%Etiqueta de la Ordenada
ylabel('Desplzamiento [m]')
```

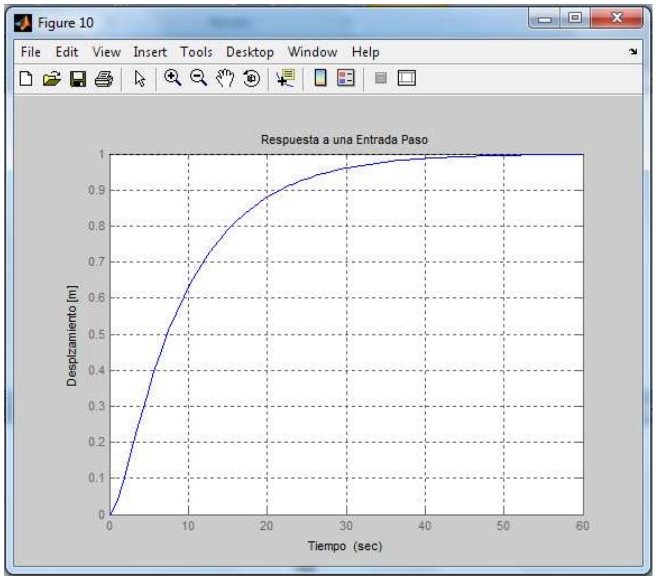


Figura # 3.51: Respuesta a una entrada paso
Fuente: **Autores**

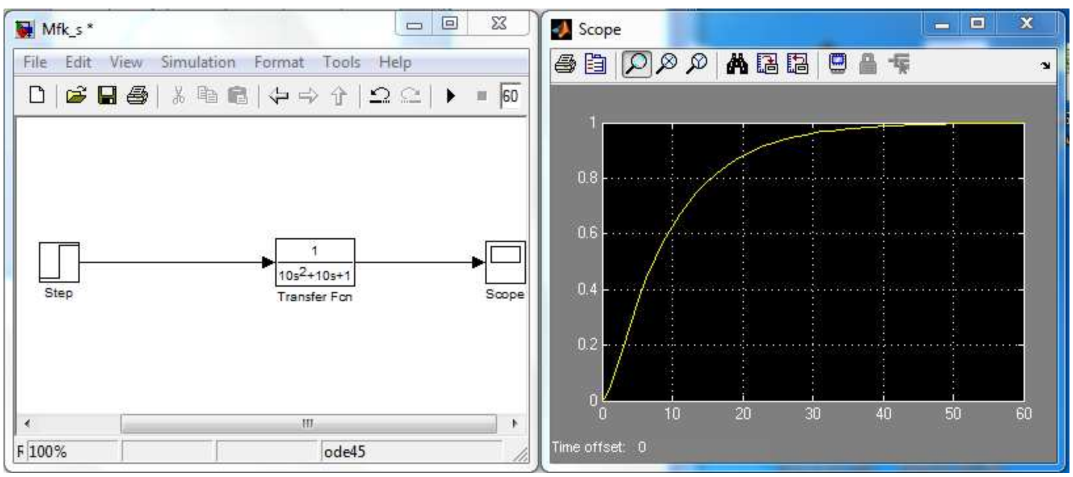


Figura # 3.52 : Diagrama de bloques en Simulink con su respectiva señal a una entrada paso
Fuente: Autores

5. Resultados y conclusiones

De la respuesta en el grafico se observa que el sistema es estable y el valor que alcanza en estado estacionario es de 1, consiguiendo estabilizarse en un tiempo de aproximadamente 50segundos.

Practica # 6: Sistemas de Segundo Orden: Sistema Mecánico Fuerza M, F, K, Respuesta a una entrada paso ante cambios de M:[1 100 1000]

1. Objetivo

Obtener el modelo y la función de transferencia del sistema Mecánico

Hallar por medio de la simulación la respuesta en el tiempo de este modelo de sistema ante cambios en la masa y así poder analizar su estabilidad.

2. Equipos y software utilizados

Computadora

Software Matlab

Software Simulink

3. Desarrollo de la practica

Lo primero que haremos es Obtener la función de transferencia en matlab y el diagrama en bloques en Simulink del sistema mecánico de los tres cambios de masa realizado.

```
%Respuesta a una entrada Paso ante
%Cambios de M:[1 100 1000]
Mv=[1 100 1000];
%Lazo para generar cambios de Mv
for i=1:3
    %Ingresando Denominador
    den=[Mv(i)  f k];
    %Definiendo la Funcion de Transferencia
    G=tf(num1,den);
    %Mostrando la Funcion de Transferencia
    display(G)
    %Respuesta a una entrada Paso
    %Asignacion a una figura
    figure(20+i)
    %Aplicando una entrada paso
    step(G)
    %Definiendo la cuadricula
    grid
    %Titulo
    title(['Respuesta a una Entrada Paso con',' M= ',num2str(Mv(i)),'
'])
    %Etiqueta de la Accisa
    xlabel('Tiempo ')
    %Etiqueta de la Ordenada
    ylabel('Desplzamiento [m]')
end
```

```
Transfer function:
      1
--------------
s^2 + 10 s + 1



Transfer function:
        1
------------------
100 s^2 + 10 s + 1
```

```
Transfer function:
         1
-------------------
1000 s^2 + 10 s + 1
```

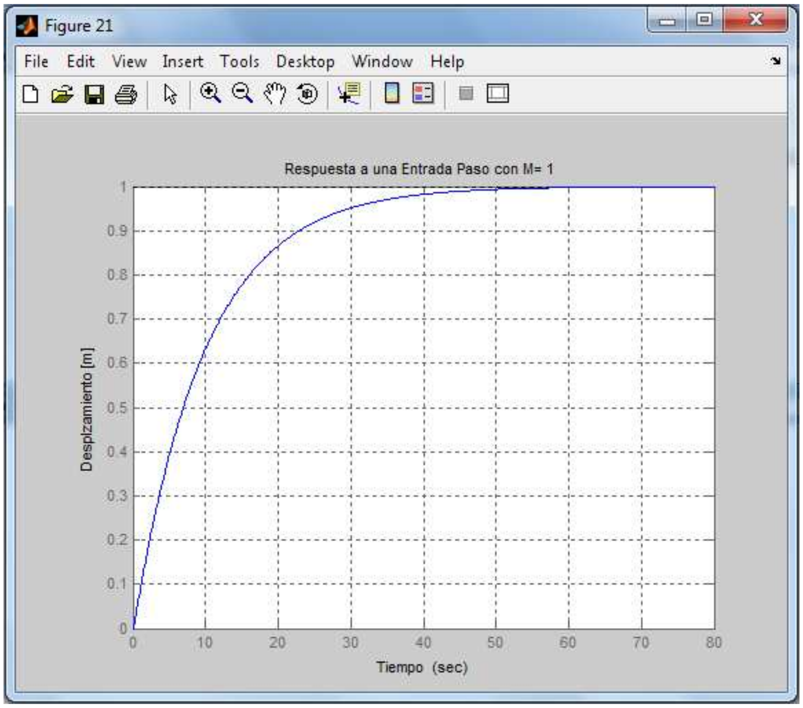


Figura # 3.53: Respuesta a una entrada paso con una masa de 1
Fuente: Autores

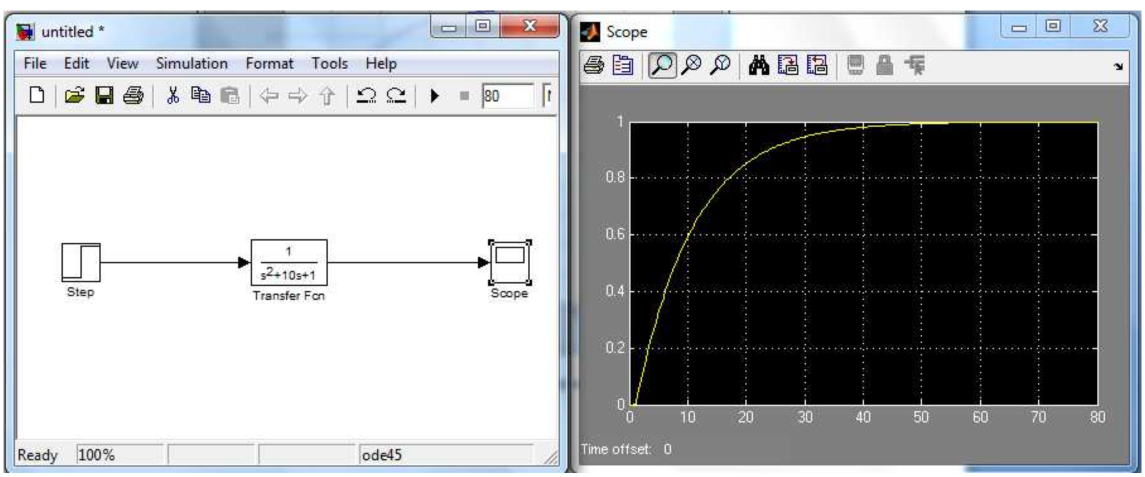


Figura # 3.54: Diagrama de bloques en Simulink con su respectiva señal a una entrada paso con una M= 1
Fuente: Autores

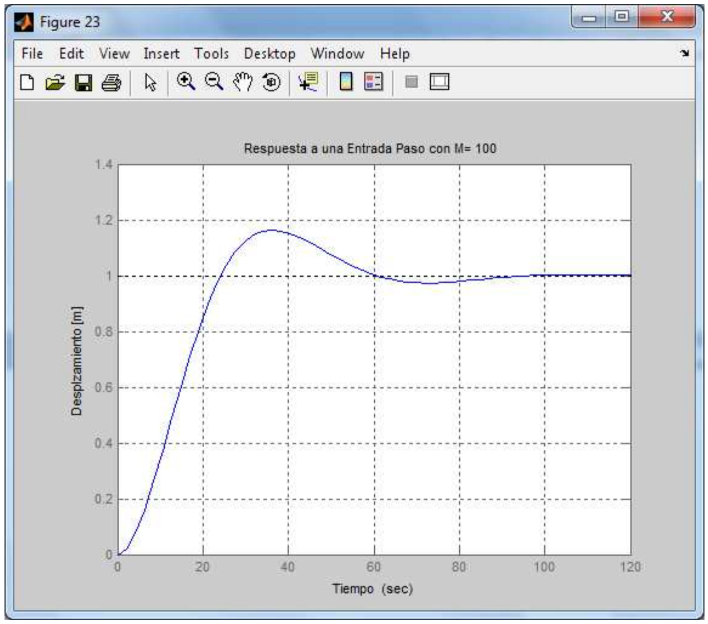


Figura # 3.55: Respuesta a una entrada paso con una masa de 100
Fuente: Autores

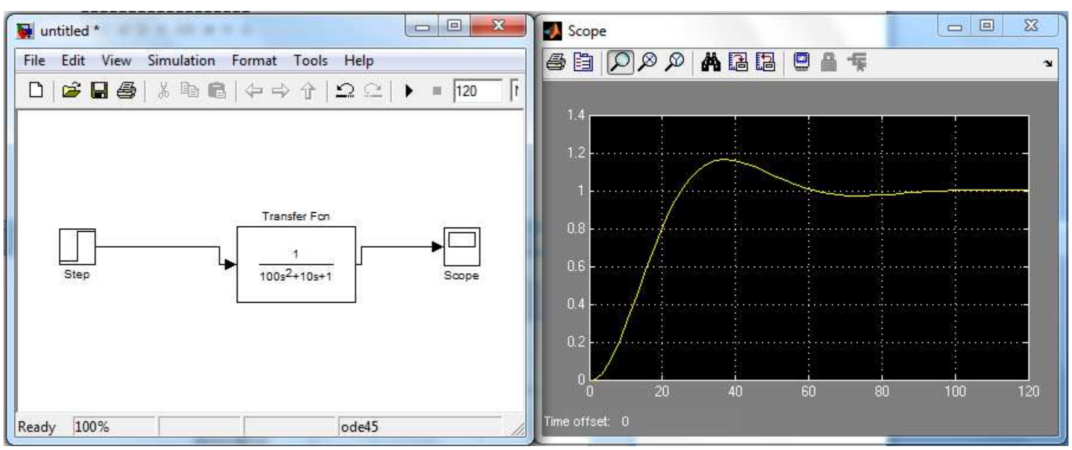


Figura # 3.56: Diagrama de bloques en Simulink con su respectiva señal a una entrada paso con una M= 1
Fuente: Autores

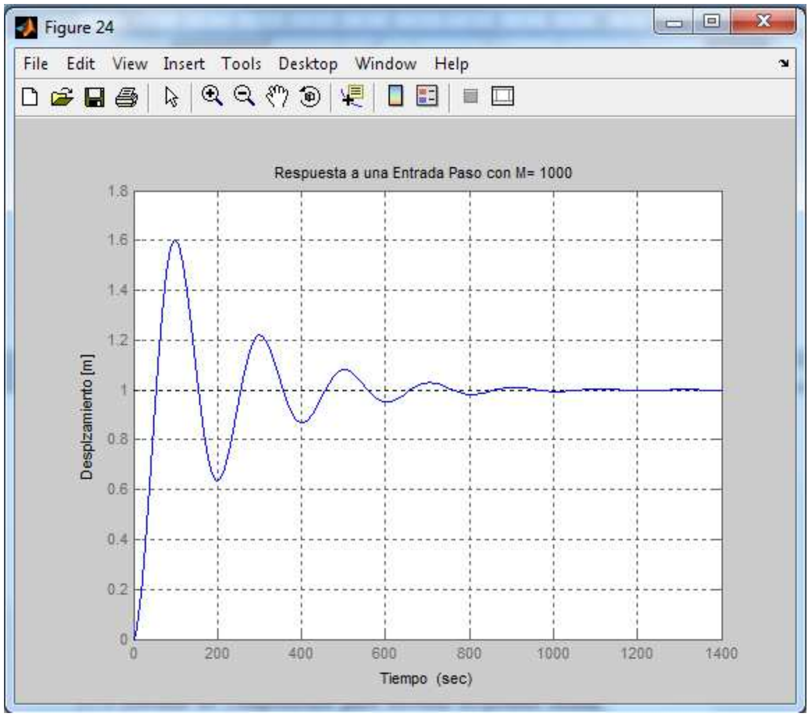


Figura # 3.57: Respuesta a una entrada paso con una masa de 1000

Fuente: Autores

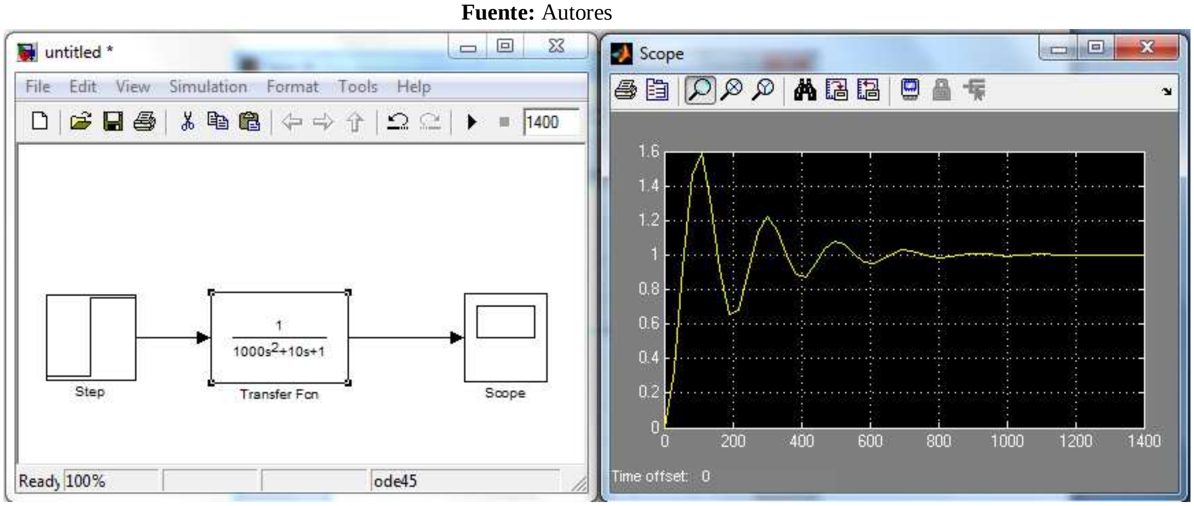


Figura # 3.58: Diagrama de bloques en Simulink con su respectiva señal a una entrada paso con una M= 100
Fuente: Autores

4. Resultados y conclusiones

El sistema responde con amortiguación a las señales de una entrada paso cambiando su masa en los diferentes valores anteriormente indicados.

Practica # 7: Sistema de Segundo Orden: Sistema Mecánico Fuerza M,F,K, respuesta a una entrada impulso.

1. **Objetivo**

Analizar el comportamiento de un sistema de segundo orden mecánico MFK, utilizando Función de Transferencia con respuesta a una entrada impulso.

Obtener la función de transferencia en matlab y el diagrama en bloques en Simulink

2. **Equipos y software Utilizados**

Computadora

Software Matlab

Software Simulink

3. **Desarrollo de la Práctica**

En Matlab la respuesta impulsional se puede obtener del mismo modo que la respuesta escalón paso, pero usando en este caso la función **impulse**, que tiene una sintaxis similar al comando **step**. , utilizado en la respuesta escalón-paso.

```
%Respuesta a una entrada Impulso
%Asignacion a una figura
figure(30)
%Aplicando una entrada Impulso
impulse(G1)
%Definiendo la cuadricula
grid
%Titulo
title('Respuesta a una Entrada Impulso')
%Etiqueta de la Accisa
xlabel('Tiempo ')
%Etiqueta de la Ordenada
ylabel('Desplzamiento [m]')
```

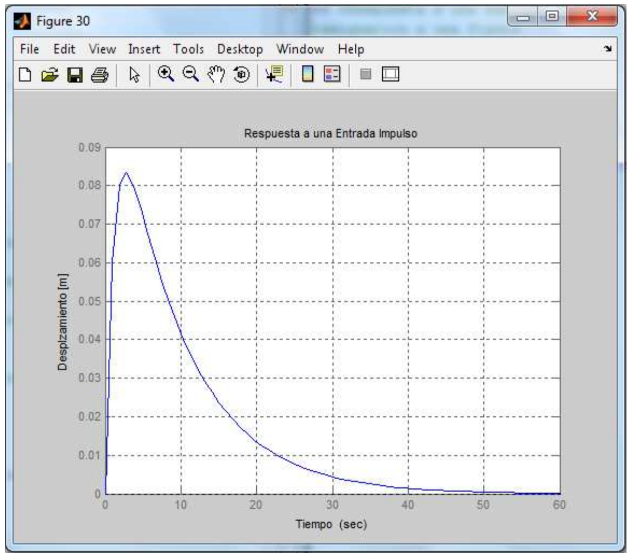


Figura # 3.59: Respuesta a una entrada impulso
Fuente: Autores

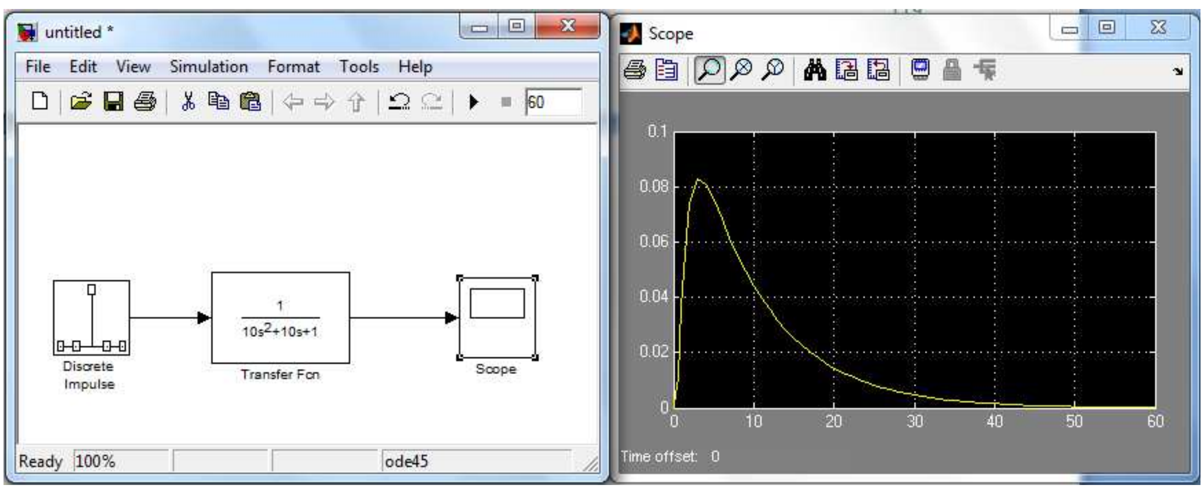


Figura # 3.60: Diagrama de bloques en Simulink con su respectiva señal a una entrada impulso
Fuente: Autores

4. Resultados y Conclusiones

En la gráfica se aprecia la respuesta del sistema de segundo orden ante una entrada impulso y se observa que efectivamente este responde ante un cambio en la entrada pero luego retorna al valor de salida cero. Esta respuesta nos indica que el sistema es estable.

Practica # 8: Sistema de Segundo Orden: Sistema Mecánico Fuerza M,F,K, respuesta a una entrada impulso ante Cambios de M=[1 100 1000]

1. Objetivo

Obtener el modelo y la función de transferencia del sistema Mecánico.

Hallar por medio de la simulación la respuesta en el tiempo de este modelo de sistema ante cambios en la masa y así poder analizar su estabilidad.

2. Equipos y software utilizados

Computadora

Software Matlab

Software Simulink

3. Desarrollo de la Práctica

Lo primero que haremos es Obtener la función de transferencia en matlab y el diagrama en bloques en Simulink del sistema mecánico de los tres cambios de masa realizado.

```
%Respuesta a una entrada Impulso ante
%Cambios de M=[1 100 1000]
Mv=[1 100 1000];
%Lazo para generar cambios de Cv
for i=1:3
    %Ingresando Denominador
    den=[Mv(i)  f k];
    %Definiendo la Funcion de Transferencia
    G=tf(num1,den);
    %Mostrando la Funcion de Transferencia
    display(G)
    %Respuesta a una entrada Impulso
    %Asignacion a una figura
    figure(40+i)
    %Aplicando una entrada Impulso
    Impulse(G)
    %Definiendo la cuadricula
    grid
    %Titulo
    title(['Respuesta a una Entrada Impulso con',' M=
',num2str(Mv(i)),' '])
    %Etiqueta de la Accisa
    xlabel('Tiempo ')
    %Etiqueta de la Ordenada
    ylabel('Desplzamiento [m]')
end
```

```
Transfer function:
      1
--------------
s^2 + 10 s + 1


Transfer function:
        1
------------------
100 s^2 + 10 s + 1


Transfer function:
         1
-------------------
1000 s^2 + 10 s + 1
```

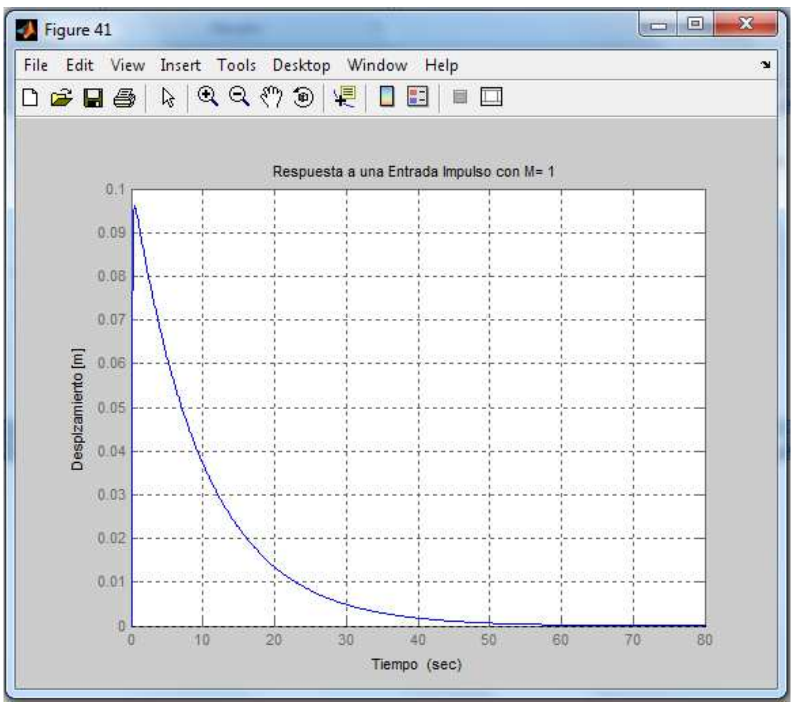


Figura # 3.61: Respuesta a una entrada impulso con masa 1
Fuente: Autores

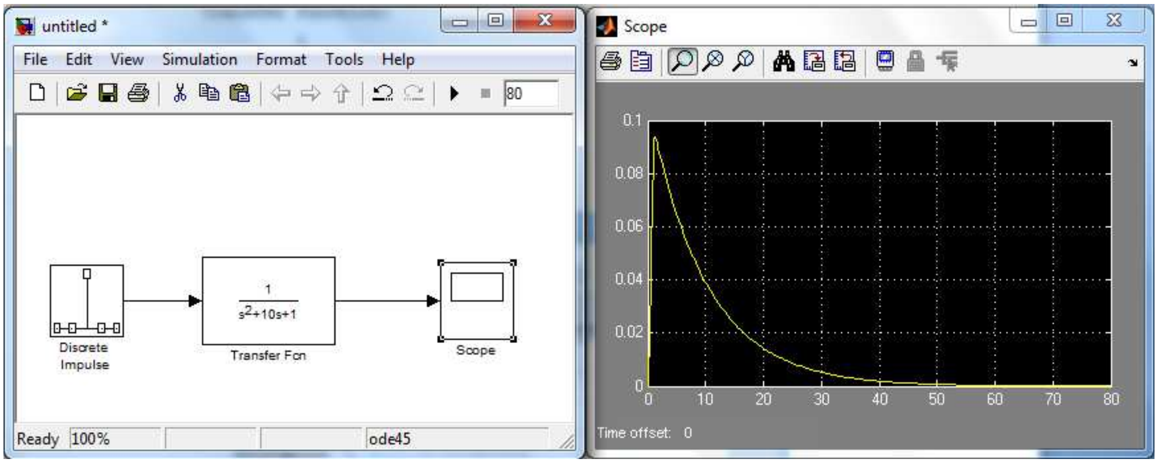


Figura # 3.62: Diagrama de bloques en Simulink con su respectiva señal a una entrada impulso con una M=1
Fuente: Autores

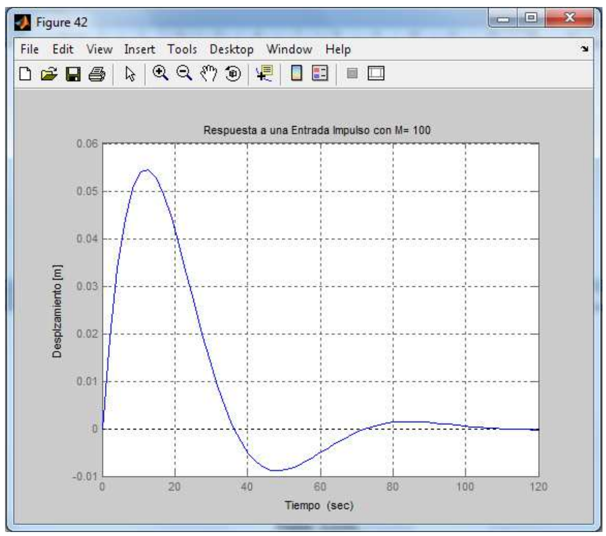


Figura # 3.63: Respuesta a una entrada impulso con masa 100
Fuente: Autores

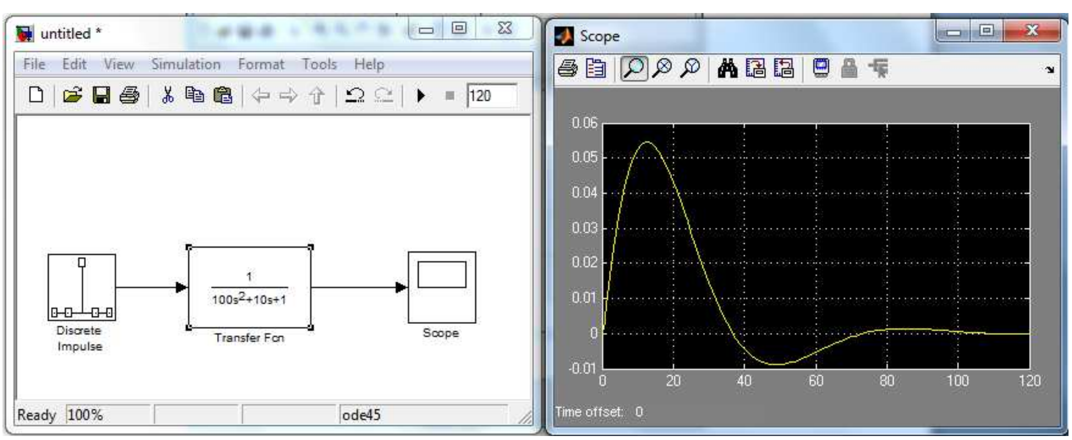


Figura # 3.64: Diagrama de bloques en Simulink con su respectiva señal a una entrada impulso con una M=100

Fuente: Autores

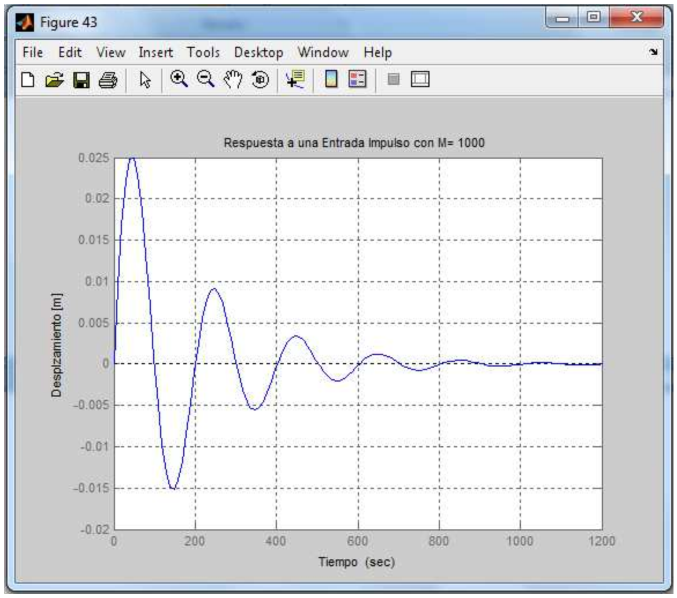


Figura # 3.65: Respuesta a una entrada impulso con masa 1000
Fuente: Autores

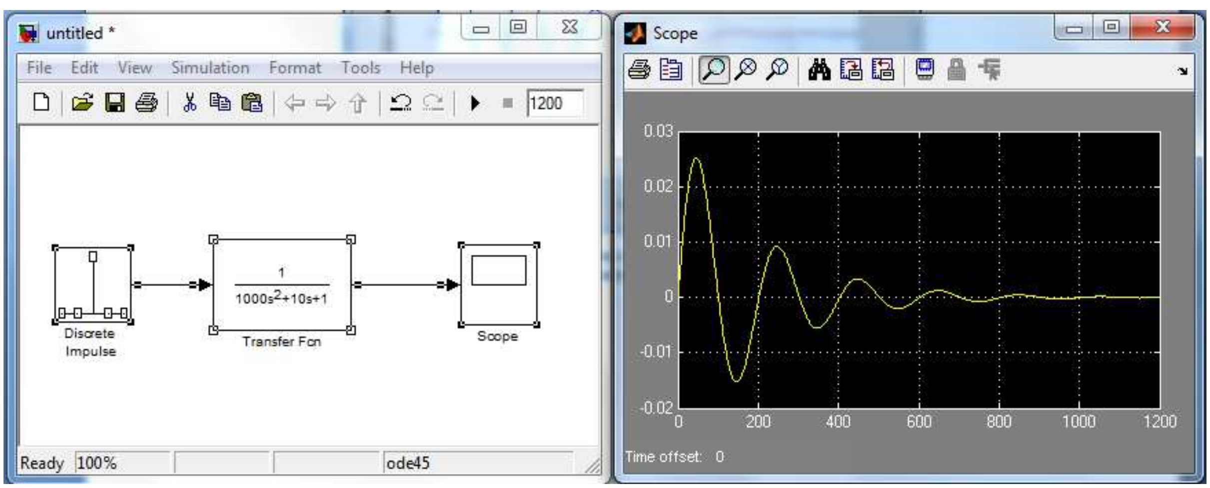


Figura # 3.66: Diagrama de bloques en Simulink con su respectiva señal a una entrada impulso con una M=1000
Fuente: Autores

4. Resultados y Conclusiones

En las gráficas se aprecia las respuestas del sistema de segundo orden ante una entrada impulso y se observa que efectivamente este responde ante un cambio en la entrada, con la única variación que al cambiar las masas aparecen algunos amortiguamientos, pero luego retorna al valor de salida cero. Esta respuesta nos indica que el sistema es estable

CAPÍTULO 4

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones.

Para obtener la respuesta de un sistema en el tiempo ante una entrada estándar, debe primero definirse el sistema a utilizar. Para ello puede definirse en MatLab la función de transferencia propia del sistema o las ecuaciones de estado.

La función de transferencia de un sistema es una relación formada por un numerador y un denominador.

Con el desarrollo de estas práctica se pudieron afianzar los conocimientos matemáticos útiles para el análisis y solución de sistemas físicos (mecánicos, eléctricos ,etc.) mediante modelado matemático.

Con ayuda de la herramienta Matlab pudimos simular la respuesta en el tiempo de los sistemas físicos propuestos en la práctica y así entendimos con ayuda de esta simulación cuando un sistema es estable o inestable.

Al desarrollar estas prácticas nos dimos cuenta de que un modelo matemático de un sistema físico puede cambiar dependiendo de la perspectiva de sus mismas variables.

Se pudo comprender en general que en Matlab con ayuda de la matemática simbólica se nos facilitan muchos cálculos matemáticos que cuando los realizamos sin una herramienta software de alta capacidad nos complican el desarrollo de los problemas, se nos hacen complejos por las cantidades de variables que se deben de tener en cuenta cuando realizamos el análisis respectivo de este problema. La matemática simbólica de Matlab es muy importante para el cálculo en general, adicionalmente Simulink está integrado en Matlab y por ello es posible tener acceso a una amplia gama de herramientas que permiten desarrollar algoritmos, analizar y visualizar simulaciones.

4.2 Recomendaciones

Se recomienda la utilización constante de las herramientas de Matlab y Simulink en los laboratorios con el fin de potencializar el aprendizaje, el conocimiento y el desarrollo integral de los alumnos, profesores y profesionales de la rama.

Se aconseja masificar el aprendizaje de las herramientas de software Matlab y Simulink con el desarrollo de prácticas de circuitos reales y tarjetas de adquisición de datos, para que los estudiantes puedan realizar sus propios diseños de acuerdo a sus necesidades y direccionados hacia aplicaciones específicas.

Se recomienda que además del Matlab y el Simulink se puedan usar otros programas de simulación y cálculos como SPICE y Labview

4.3 Bibliografía

Alicia Arce Rubio, G. V. (2009). *Manual de simulink para la asignatura de teoría de sistemas.* Sevilla: Universidad de Sevilla.

El Halabi, N. (2007). *Sistemas de control: ganancias de realimentación y observadores de estado.* Argentina: El Cid Editor - Ingeniería .

García, E. J. (2004). *Técnicas de Automatización avanzadas en procesos industriales.* Rioja: Universidad de la Rioja servicios de publicaciones.

R.P. Ñeco, O. Reinoso, N.García, R. Arcil. (2003). *Apuntes de Sistemas de Control.* Elche: Editorial Club Universitario.

Gil Rodríguez, M. (2006). *Introducción rápida a Matlab y Simulink para ciencia e ingeniería.* España: Ediciones Díaz de Santos .

Gómez Sarduy, J. R. (2008). *Temas especiales de instrumentación y control.* Cuba: Editorial Félix Varela .

Javier García de Jalón, J. I. (2005). *Aprenda Matlab 7.0 como si estuviera en primero.* Madrid: Escuela Técnica Superior de Ingenieros Industriales Universidad Politécnica de Madrid.

Kuo, B. C. (2001). *Sistemas de Control Automatico.* Mexico: Pearson Educación.

Ma Antonia Simón Rodríguez, F. D. (2009). *Regulación automática. Problemas resueltos.* Madrid: Editorial Visión Libros.

Martínez, A. A. (01 de 02 de 2000). *http://www.dea.icai.upco.es.* Recuperado el 23 de 07 de 2012, de http://www.dea.icai.upco.es/ramon/Manual_Matlab.pdf

Moreno, R. P. (01 de 09 de 2001). *automata.cps.unizar.es.* Recuperado el 23 de 07 de 2012, de automata.cps.unizar.es/regelectricos/Practicas_2345Electricos.pdf

NUÑEZ, M. J. (12 de 09 de 2012). *http://gama.fime.uanl.mx.* Recuperado el 12 de 09 de 2012, de http://gama.fime.uanl.mx/~agarcia/materias/ingco/apclas/03%20-%20Diagramas%20de%20Bloques.pdf

Ogata, K. (2003). *Ingeniería de Control Moderna.* Madrid: Pearson Educación.

Rivero, R. A. (15 de 09 de 2012). *www.edutecne.utn.edu.ar.* Recuperado el 17 de 09 de 2012, de http://www.edutecne.utn.edu.ar/tutoriales/identificacion-sistemas-segundo-orden.pdf

Rodriguez, J. Á. (01 de 01 de 2004). *http://www.esi2.us.es.* Recuperado el 23 de 07 de 2012, de http://www.esi2.us.es/~jaar/Datos/RegAuto/Practica3.pdf

Villegas, L. (2007). *Trabajo teórico práctico con Matlab.* Argentina: El Cid Editor - Informática .