

Universidad Central “Marta Abreu” de Las Villas
Facultad de Ingeniería Eléctrica
Departamento de Automática y Sistemas Computacionales



TRABAJO DE DIPLOMA

Las asignaturas Modelado y Simulación e Ingeniería de Control II sobre Scilab.

Tesis presentada en opción al grado de
Ingeniero en Automática

Autor: Julian Miranda Uría

Tutor: Msc. María del Carmen Hernández Carús
Dr.C. Ivan Santana Ching

Santa Clara

2017

“Año 59 de la Revolución”

Universidad Central “Marta Abreu” de Las Villas
Facultad de Ingeniería Eléctrica
Departamento de Automática y Sistemas Computacionales



TRABAJO DE DIPLOMA

Las asignaturas Modelado y Simulación e Ingeniería de Control II sobre Scilab.

Tesis presentada en opción al grado de
Ingeniero en Automática

Autor: Julian Miranda Uría
email: jmiranda@uclv.cu

Tutor: Msc. María del Carmen Hernández Carús Prof. Asistente
Dpto. de Automática, Facultad de Ing. Eléctrica, UCLV
email: Carmen@uclv.edu.cu

Dr.C. Ivan Santana Ching
Dpto. de Automática, Facultad de Ing. Eléctrica, UCLV
email: Ching@uclv.edu.cu

Santa Clara

2017

“Año 59 de la Revolución”



Hago constar que el presente trabajo de diploma fue realizado en la Universidad Central "Marta Abreu" de Las Villas como parte de la culminación de estudios de Ingeniería en Automática, autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

Julian Miranda Uría
Autor

Fecha

Los abajo firmantes certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Julian Miranda Uría
Autor

Fecha

Alain Martínez Laguardia, Dr.C
Jefe del Departamento

Fecha

Responsable ICT o J' de Carrera, (Dr.C., M.Sc. o Ing.)
Responsable de Información Científico-Técnica

Fecha

PENSAMIENTO

“Solo trabajo con los mejores. No se trata de los mejores curriculums, sino de la actitud correcta”.

Donald Trump.

DEDICATORIA

A mi padre,
por todo el esfuerzo y sacrificio que ha realizado para poder ser quien soy.

A mi madre
*por estar siempre pendiente del desarrollo de mi carrera, por su paciencia y
compromiso.*

A mi hermana
por su apoyo incondicional ante cualquier situación.

A mi hermano (Javier)
*por estar siempre a mi lado, a pesar de todas las adversidades que hemos pasado en
estos cinco años.*

*A ellos, por ser los responsable de mi formación y educación, por confiar
en mí, y ser mi apoyo incondicional.*

AGRADECIMIENTOS

Especialmente a mis padres, que me han apoyado en todo momento y han sabido moldear mi personalidad hasta convertirme en quien soy hoy.

A mi hermana, que siempre estuvo a mi lado para lo que fuese necesario.

A mis hermanos, Luis Javier y Pedro Pablo, cada uno por su lado regalándome los mejores momentos de mi vida.

A mi tutora María del Carmen, por su apoyo en cada momento de la investigación, por siempre haber tenido todo su tiempo a mi disposición. Pero sobre todo, por ser mi amiga.

A Ana Mayra, que estuvo presente en cada uno de los momentos que viví en la universidad. A parte, por ser mi amiga de siempre.

A Claudia, por acogerme en su vida en el peor momento de mi carrera y ayudarme a salir adelante.

A Danny, por enseñarme a ser perseverante y mantener la calma en los momentos de estudio.

A Marcel, por ser mi mano derecha para lo que mi hiciera falta.

A Adrián P. M., Randy, El Pewy, El Mene, Claudia Patricia, Beatriz Mabel, por haber estado siempre pendiente.

A todos los amigos y compañeros de estudio, que de una forma u otra hemos compartido momentos de fracaso y victoria en estos cinco años, en especial a Camilo, Francis, Oscar, Rolando, Diony, Rubén, Javier Ojeda, etc.

A todos los profesores que durante la carrera han intervenido en mi formación profesional.

En general quisiera agradecer a todo aquel que de una forma u otra, ayudó a la realización de este trabajo de diploma y a mi formación.

Santa Clara, Cuba, 2017.

RESUMEN

La utilización del software libre hoy en día es una realidad en el mundo, las aplicaciones que se crean sobre esta base son cada día más eficientes y se difunden en todos los campos del conocimiento.

El software Matlab posee las prestaciones para cumplir de manera eficiente todas las tareas que se trazan en el desarrollo de la Ingeniería Automática; pero el empleo de este software está restringido a una patente, la cual, es un impedimento para Cuba por su alto costo y por el bloqueo económico impuesto por el gobierno de los Estados Unidos de América.

Por estas razones, en el año 2015 en el Departamento de Automática y Sistemas Computacionales de la Universidad Central “Marta Abreu” de las Villas, se comenzó un proceso de migración de las asignaturas de la disciplina Sistemas de Control hacia el software libre Scilab, ya que este presenta la mayoría de las funciones y herramientas que permiten a los estudiantes cumplir de manera eficiente todas las tareas a vencer en el año académico.

Con esta investigación queda constancia de la posibilidad de realizar todas las actividades prácticas de las asignaturas Modelado y Simulación e Ingeniería de Control II en dicho software. Se deja en anexos la preparación de todos los ejercicios que actualmente se desarrollan en ambas asignaturas.

INTRODUCCIÓN

En los inicios del proceso de información, con la informática sólo se facilitaban los trabajos repetitivos y monótonos del área administrativa, pero la automatización de esos procesos trajo como consecuencia directa una disminución de los costos y un incremento en la productividad. Más adelante, la computación se colocó como la principal ciencia que impulsaba el desarrollo tecnológico, tal es el caso de Ingeniería en Automática, la cual ha logrado su gran avance gracias al progreso del software. Sin embargo, el software usado en esta rama de la ingeniería es muy costoso y esto impide a muchos países desarrollar investigaciones que tributen a la automatización de industrias. Por este motivo muchos países se ven obligados a buscar alternativas económicamente viables para impulsar el desarrollo de sus principales sectores.

El movimiento de software libre tuvo su gran impulso con el surgimiento de *Free Software Foundation* (FSF), organización creada por Richard Stallman en el año 1985. Según Stallman, las universidades como fuente de creación y divulgación de conocimiento deben asumir, entre sus compromisos sociales, la promoción y el desarrollo del software libre como herramienta liberadora, ya que es innegable los aportes del software propietario en la creación del conocimiento ([Stallman, 2004](#)).

La poderosa herramienta de software Matlab posee las prestaciones para cumplir de manera eficiente todas las tareas que se trazan en el desarrollo de la Ingeniería Automática; pero el empleo de este software esta restringido a una patente, la cual, es un impedimento para Cuba por su alto costo y por el bloqueo económico impuesto por el gobierno de los Estados Unidos de América.

Por estas razones, en el año 2015 en el Departamento de Automática y Sistemas Computacionales de la Universidad Central “Marta Abreu” de las Villas, se comenzó un proceso de migración de las asignaturas de la disciplina Sistemas de Control hacia el software libre Scilab, ya que este presenta la mayoría de las funciones y herramientas que permiten a los estudiantes cumplir de manera eficiente todas las tareas a vencer en el año académico.

Pérez Lanza ([Lanza, 2015](#)) realizó un estudio acerca del software propietario Matlab, donde expone las desventajas de usar el mismo y evalúa el uso de un software libre

que tenga iguales prestaciones como solución. En esta investigación se determinó que el software libre Scilab es el más apropiado para apoyar el aprendizaje de las asignaturas de Sistema de Control.

La migración comenzó con la investigación de Giovanni Landaburo ([Landaburo, 2016](#)). En este trabajo se prepararon las actividades prácticas de la asignatura Ingeniería de Control I, utilizando las funciones que ya trae Scilab e incorporando algunas nuevas creadas por él, que garantizan el cumplimiento de los objetivos a alcanzar por esa asignatura.

En estos momentos restan para la total migración de la disciplina las asignaturas: Modelado y Simulación, Ingeniería de Control II, Procesos, Máquinas Eléctricas, Accionamiento Eléctrico y Control de Procesos.

En consecuencia se plantea el siguiente **problema científico**: Las asignaturas Modelado y Simulación e Ingeniería de Control II no han sido implementadas en Scilab. Además el software Matlab al ser propietario no tiene acceso a su código fuente impidiendo modificarlo.

La hipótesis de esta investigación se sustenta en los resultados de la revisión bibliográfica presentada en el Capítulo 1.

Hipótesis:

La utilización de las herramientas de simulación que forman parte del software libre Scilab garantiza la realización de las actividades prácticas de las asignaturas Modelado y Simulación e Ingeniería de Control II que se imparten en la carrera Ingeniería Automática de la Universidad Central “Marta Abreu” de las Villas.

Con esta investigación se pretende cumplir los siguientes objetivos:

Objetivo general: Desarrollar las actividades prácticas de las asignaturas Modelado y Simulación e Ingeniería de Control II sobre Scilab, que contribuya a la migración paulatina de la disciplina Sistema de Control a software libre.

Objetivos específicos:

1. Analizar la bibliografía existente y trabajos previos en el Departamento sobre las posibilidades de sustituir Matlab por Scilab como software de apoyo a la impartición de las asignaturas de la disciplina Sistemas de Control.
2. Confeccionar las actividades prácticas de la asignatura con una posible solución y sus resultados.
3. Utilizar las funciones del software Scilab que permitan conseguir resultados semejantes a los que se obtienen en la actualidad con el Matlab.

Para lograr el cumplimiento de los objetivos anteriormente expuestos se propone ejecutar las siguientes tareas:

- Revisión en la literatura especializada de los aspectos teóricos relacionados con la temática que aparecen reportados en la literatura.
- Selección de las actividades prácticas de las asignaturas Modelado y Simulación e Ingeniería de Control II a implementar en Scilab.
- Estudio del software Scilab.
- Identificación de las herramientas de Scilab necesarias para la implementación de las actividades prácticas.
- Elaboración del informe científico de la investigación.

Contenido de la Tesis:

La investigación incluye tres capítulos, además de las conclusiones, recomendaciones, referencias bibliográficas y anexos correspondientes. Los temas que se abordan en cada capítulo se encuentran estructurados de la forma siguiente:

Capítulo I: En el primer capítulo se realiza un estudio de los trabajos precedentes, así como, la literatura especializada sobre el software libre Scilab. Se presenta además, el impacto que ha alcanzado el software libre a nivel mundial. Se analizan los temas de las asignaturas Modelado y Simulación e Ingeniería de Control II a tener en cuenta para este proceso migratorio, así como las herramientas que son necesarias emplear en Scilab para la confección de las actividades prácticas.

Capítulo II: En el segundo capítulo se presentan las actividades prácticas desarrolladas sobre Scilab para la asignatura Ingeniería de Control II. Se dan a conocer los resultados gráficos y numéricos obtenidos en cada una de ellas.

Capítulo III: En el tercer capítulo se abordan las actividades prácticas correspondientes a la asignatura Modelado y Simulación. Se muestran los resultados numéricos y gráficos obtenidos. Por último, se presenta el análisis económico y medioambiental de la investigación.

CAPÍTULO 1

ESTUDIO DEL SOFTWARE LIBRE EN LA ENSEÑANZA

1.1. Introducción

En este capítulo se hace una breve descripción de la enseñanza de Ingeniería en Automática en la Universidad Central “Marta Abreu” de Las Villas. Se realiza un análisis sobre la situación del software privado usado en las prácticas de la disciplina Sistemas de Control y las desventajas que trae consigo, además se propone la migración hacia software libre en las asignaturas de la disciplina, específicamente Modelado y Simulación e Ingeniería en Control II.

1.2. Definición del software libre

Las ideas de software libre nacen en el laboratorio de Inteligencia Artificial (AI Lab) del Instituto Tecnológico de Massachusetts (MIT) en 1971. El software que escribían era compartido a otras universidades y cuando se les pedía el código fuente, lo compartían para leerlo o modificarlo para formar uno nuevo ([Stallman, 2004](#)). Uno de los miembros y programadores era Richard Stallman, y estas experiencias le sirvieron para fundar el movimiento llamado Software Libre. Entonces, software libre es aquel que respeta la libertad de sus usuarios y de la comunidad por lo que Stallman creó en 1985 la *Fundación del Software Libre* (FSF) y definió cuatro libertades básicas que tiene que cumplir un software libre ([López, 2005](#)):

1. Usar el programa, con cualquier propósito.
2. Estudiar cómo funciona el programa y adaptarlo a las necesidades específicas del usuario mediante el acceso al código fuente.
3. Distribuir copias, de las que puedan beneficiarse otros usuarios.
4. Mejorar el programa y hacer público esos desarrollos, de modo que toda la comunidad se beneficie de ello.

Estas libertades traen como consecuencia muchas ventajas, entre las cuales se destaca la económica, cuestión que debe ser aprovechada por las instituciones educativas para

incorporarlas al aula de clases. En la educación las principales ventajas de usar software libre son ([Heinz, 2017](#)):

- Los costos para las instituciones son menores.
- Poder acceder al código fuente permite la innovación y la apropiación de nuevas tecnologías.
- Se puede adaptar a las necesidades locales y de cualquier persona.
- Permite que el usuario copie y difunda el software sin incurrir a copias ilegales.
- Difunda el conocimiento, permite modificar, construir y cooperar con los compañeros.
- Que el estudiante pueda contribuir a mejorar el software.
- Sobre todo en investigación, que es una de las tareas principales de toda institución educativa, el uso de software libre es básico ya que el software privado evita conocer cómo funciona el programa.

Para amparar jurídicamente estos cuatro propósitos, se redactó y se utiliza la denominada licencia GNU GPL (Licencia Pública de GNU). Los términos y condiciones de GPL deben estar disponibles para cualquiera que reciba una copia de la obra al cual ha sido aplicada esta licencia. Cualquier licencia que se le apliquen dichos términos da permiso a realizar modificaciones a una obra, realizar copias y distribuirlas o distribuir cualquiera de sus versiones derivadas. Con esta licencia está permitido cobrar por la distribución de cada copia, o no cobrar nada. Este último punto distingue las licencias GPL de las licencias de software que prohíben la distribución comercial. La FSF argumenta que en el software libre no debe haber cabida para las restricciones comerciales y que las obras bajo este tipo de licencias pueden ser vendidas a cualquier precio ([Nelson, 2007](#)).

1.2.1. Software Scilab

SciLab es un “lenguaje interpretado” para cálculo científico, que fue desarrollado en 1990 por el instituto francés de investigación INRIA (Institut National de Recherche en Informatique et Automatique) ([Enterprises et al., 2012](#)). Posteriormente colaboró en su desarrollo, la escuela de ingenieros ENPC (Ecole Nationale de Ponts et Chaussées). Desde mayo del 2003, es mantenido y desarrollado por “SciLab Consortium”, un consorcio de empresas colaboradoras como Renault, Peugeot-Citroën, CEA (Commissariat à l’Energie Atomique), CNES (Centre National d’Etudes spatiales), Dassault Aviation, EDF (Electricité de France), entre otras.

La herramienta Scilab, que cumple con los requisitos de licencia GNU GPL establecidos por la FSF ([Stallman, 2004](#)), es una de las más usadas y conocidas en el ámbito matemático, por sus ventajas explicadas anteriormente y sobretodo, por ser gratuita. Pero también posee gran documentación y ayuda on-line (solo en inglés y francés) que

está disponible para todos sus usuarios. Scilab dispone de un manual de usuario que se puede consultar en una ventana de ayuda (Help Browser).

Otra ventaja que ofrece esta completa herramienta, es la posibilidad de proporcionar dentro de su documentación en su página oficial, un apartado para los usuarios que antes usaron Matlab (Freire, 2016). Este beneficio hace que la herramienta pueda ser utilizada por más personas, ya que al ser gratuita, el usuario que antes usaba Matlab, puede realizar lo mismo en Scilab y aprender de manera rápida y sencilla gracias a los ejemplos.

Scilab incluye cientos de funciones especializadas para computación numérica (Santos, 2013), organizadas en librerías llamadas Toolboxes que cubren muchas áreas como: simulación, sistemas y control, optimización y procesamiento de señales. Además posee un simulador de diagramas en bloques (Xcos) el cual se usa para sistemas dinámicos híbridos, control clásico, control robusto, optimización LMI, optimización diferenciable y no diferenciable, tratamiento de señales, grafos y redes (Brewster, 2011; Sharma, 2010). Cuenta además con Scilab paralelo que emplea PVM (máquina virtual paralela), procesamiento estadístico, creación de GUIs, una interfaz para el cálculo simbólico usando Maple y MuPAD4, así como una interfaz con lenguaje de herramientas de comando TCL5 y TK6. Además se pueden agregar numerosas herramientas o toolboxes hechas por los usuarios, por ejemplo de ellos la herramienta Open FEM (una caja de Herramientas para Elementos Finitos) hecha por INRIA.

1.2.2. Software libre en la educación

Los programas educativos pueden utilizarse para apoyar y ampliar las experiencias de aprendizaje dentro del enfoque en la educación (Squires, 1997). Y es que, la utilización del software educativo se ha vuelto una necesidad para el apoyo que pueden tener los estudiantes en las diferentes áreas del conocimiento.

El software libre facilita la apropiación del conocimiento y por ello tiene una gran cantidad de beneficios para la educación comparados con el software privativo, entre ellos se encuentran:

- Se reducen costos de equipos: La ventaja del software libre en relación con el hardware es que no requiere equipos tan potentes y es posible que funcione en la mayoría de las computadoras que existen en el mercado.
- Se reduce costos de licencias: Las licencias del software libre son completamente gratuitas, por lo que las escuelas pueden ahorrar dinero e invertirlo en otras áreas.

- Permite compartir conocimiento: Al compartir material educativo libre se comparte conocimiento y forman comunidades de aprendizaje en las que se valoran las experiencias de éxito.
- Independencia: el uso de programas con licencia gratuita elimina la dependencia y ofrece libertad para elegir el programa que más agrade al usuario y que mejor se adapte a sus necesidades.

A partir de los beneficios que trae consigo el uso de software libre para la educación muchas son las experiencias que se pueden citar, entre ellas se encuentran:

- Gleducar: es una comunidad educativa libre que apoya la idea de compartir conocimiento. A través de su página web (<http://gleducar.org>) se dan a conocer algunas actividades relacionadas con la educación y software libre en general. En esta comunidad se utilizan y promueven las tecnologías libres (herramientas tecnológicas que puedan emplearse sin ninguna restricción y ser desarrolladas conjuntamente por la comunidad libre, como el código abierto, hardware y electrónica libre, formatos abiertos y protocolos libres) dentro de un marco de Innovación Pedagógica, que se refiere a cambiar el modelo de aprendizaje desde algunos puntos específicos como la teoría del aprendizaje constructivista, la pedagogía crítica y la Educación Popular (Gómez, 2012).
- Qimo, Edubuntu, Skolelinux, TuxPaint: distribuciones de sistemas operativos GNU/Linux diseñados especialmente para niños y niñas, así como herramientas de software para el aprendizaje de las matemáticas a través del juego, elaborar dibujos con el uso de la computadora, entre otras.
- Edusol: encuentro de Educación, Cultura y Software Libres, realizado anualmente en línea, con participación de docentes, especialistas y activistas de toda América Latina y España (Vargas, 2011).
- Celestia: programa de simulación de un planetario, de fácil comprensión. Se estimula la construcción de conocimiento acerca del sistema solar y es ideal para los trabajos de educación científica que se les proporciona a los niños en las escuelas (Peña, 2013).

Lo que puede aportar el software libre en las escuelas y en la sociedad, según como se introduzca, es otro modelo cultural y tecnológico, otra manera de entender la relación entre las personas, el conocimiento y las herramientas digitales. El software libre puede adaptarse a las necesidades docentes; solamente con modificarse para ofrecer a los alumnos una versión simplificada, o dar una apariencia adecuada a los conocimientos del estudiante.

1.2.3. Software libre en las universidades

Cada vez son más las universidades, tanto públicas como privadas, que utilizan y promueven el empleo de software libre entre la comunidad universitaria y en la sociedad en general. Además el desarrollo en la última década de las computadoras con gran capacidad de cálculo, trae consigo, la integración de programas matemáticos al aula que sirven para resolver problemas que demandan mucho tiempo si se realiza de forma manual. Algunos antecedentes en cuanto a la implementación del uso del software matemático en educación superior sugieren muchas ventajas ([Ortega, 2004](#)), algunos ejemplos son:

- El software Scilab ayuda a mejorar la participación, invita al trabajo individual y colectivo, ha dado espacio para que los alumnos piensen, aunque también ha provocado la disminución en las habilidades para hacer cálculos ([Rosales, 2012](#)).
- El uso del programa mejora la participación activa de los estudiantes, aumenta el interés, la motivación y el rendimiento académico con respecto a la enseñanza tradicional ([Pizarro, 2009](#)).
- El uso del software Octave en las asignaturas de Cálculo numérico logró que los estudiantes trabajaran de manera activa en las diferentes actividades propuestas como apoyo a la docencia ([Costa, 2010](#)).

Las universidades son las principales fuentes de investigación y divulgación del conocimiento, además tienen una responsabilidad y unos intereses más amplios que una empresa o un particular cuando adquiere tecnologías de la información ([Stallman, 2004](#)); sin embargo se enfrentan a problemas económicos para el uso y mantenimiento de los softwares al ser estos de carácter privativo.

Una solución a esta problemática es el uso de software libre, ya que los proyectos donde este se use, su costo inicial es muy bajo, se puede construir sobre el trabajo de otros proyectos y explorar sus aplicaciones educativas. ([Adell, 2007](#)).

1.2.4. Software libre en la carrera Ingeniería en Automática de la UCLV

La carrera Ingeniería en Automática de la Universidad Central “Marta Abreu” de Las Villas siempre ha estado a la vanguardia en la investigación y divulgación del conocimiento, pero presenta la limitante de los problemas legales que trae consigo el software privativo, por lo que es necesaria la adopción del software libre como vía de eliminar las posibles trabas que puedan existir para la transferencia de conocimiento ([Lanza, 2015](#)). Por este motivo en abril de 2004, el Consejo de Ministros adoptó el Acuerdo 084/2004 donde indicaba al Ministerio de la Informática y las Comunicaciones (MIC) ([Paumier, 2015](#)) ordenar el proceso paulatino de migración de Cuba a Software Libre. Aunque no

se haya llevado a cabo completamente dicha propuesta de migración, ya son varios los software libres que hoy en día se usan en algunas disciplinas, por ejemplo: Eclipse y Qt para las disciplinas de Computación y Programación respectivamente.

1.2.5. Software libre en la Disciplina Sistemas de Control

El Departamento de Automática y Sistemas Computacionales ha realizado trabajos previos relacionados con la migración hacia el software libre Scilab de todas las actividades prácticas de la docencia que se imparte a la carrera de Ing. Automática, en la UCLV, dentro de la disciplina Sistemas de Control. Una vez que se decidió este software por sus similitudes con el Matlab, el curso anterior comenzaron las tareas propias de la migración en la asignatura Ingeniería de Control I, por ser esta el tronco teórico de la disciplina de Sistemas de Control. El presente trabajo continúa el proceso de migración en las asignaturas Modelado y Simulación e Ingeniería de Control II.

1.3. Asignatura Modelado y Simulación

La asignatura Modelado y Simulación permite contribuir a la concepción científica del mundo y al interés por la ciencia, la técnica, la investigación, la experimentación; que permita desarrollar la capacidad de razonamiento y el pensamiento lógico. Como principales objetivos se plantea: aplicar los principios de balance de masa y energía para la obtención de modelos analíticos de los procesos, identificar experimentalmente sistemas sencillos, utilizar paquetes profesionales para la simulación y análisis de sistemas, además analizar el comportamiento dinámico de los sistemas. La asignatura pretende que los estudiantes adquieran los conceptos básicos de modelado y análisis de sistemas lineales en el dominio del tiempo, definiciones fundamentales de los sistemas de control, propiedades dinámicas de los objetos controlados, utilización de paquetes profesionales orientados para la simulación, principios básicos de la simulación analógica de sistemas y métodos de identificación de sistemas en el dominio del tiempo y de la frecuencia.

1.3.1. Temas de la asignatura Modelado y Simulación

Esta asignatura es la primera de la disciplina Sistemas de Control, por lo que en ella se introducen los conceptos preliminares para el análisis y diseño de los sistemas de control y en ella se apoyan varias de las asignaturas de la disciplina. Con el objetivo de contribuir a la mejor explotación de los procesos tecnológicos industriales, se le brindan a los egresados de la carrera, los conocimientos necesarios y el desarrollo habilidades para la obtención de modelos y la simulación de sistemas, exigiendo dominio de métodos y técnicas analíticas experimentales de identificación, algoritmos y productos de computación para la simulación digital de sistemas, básicamente en aspectos relacionados con el control de procesos mecánicos, eléctricos y electromecánicos ([Martínez, 2015-2016](#)).

La asignatura cuenta con tres temas fundamentales:

1. Fundamentos de la teoría de control. Modelación de sistemas físicos. Obtención de modelos de sistemas de diferente naturaleza: eléctricos, mecánicos, electromecánicos, hidráulicos, térmicos, etc. Respuesta temporal.

El estudiante debe familiarizarse con las diferentes representaciones de sistemas que se abordarán (FT, E-S, Modelo de Estado), así como diagramas en bloques y las formas para reducirlos. Se debe trabajar en la obtención de modelos de sistemas físicos sencillos y linealizar los no lineales; analizar la respuesta temporal de sistemas lineales e invariantes en el tiempo y las señales de prueba típicas que se usan para el análisis de estos sistemas.

2. Simulación de sistemas. Fundamentos del Matlab-Simulink. Simulación de sistemas representados por funciones transferenciales, ecuaciones de estado y ecuaciones diferenciales. Solución de ecuaciones diferenciales.

En este tema se plantea la simulación de sistemas y el empleo del paquete profesional orientado a la simulación Matlab-Simulink para la solución de ecuaciones diferenciales.

3. Identificación Experimental de Sistemas. Métodos convencionales de identificación.

El objetivo principal es la obtención de modelos sencillos de procesos a partir de datos experimentales. Se aplicarán métodos convencionales de identificación de sistemas en el dominio del tiempo y de la frecuencia y para la obtención de modelos PORT y SORT mediante la respuesta al paso.

En esta asignatura, las actividades prácticas se orientan al uso de la red de computadoras de la facultad y el software de simulación disponible en la misma, en particular, el Matlab-Simulink, el cual se presenta de forma general como un paquete de software profesional que contiene todos los bloques imprescindibles para la simulación de sistemas de control.

Por otra parte, para las actividades prácticas del tema Identificación experimental de sistemas, se usa la herramienta Ident, la cual forma parte del Toolbox llamado System Identification y consiste en un GUI que provee un ambiente interactivo para el análisis de datos, estimación de modelos y visualización de la respuesta.

En el caso de Scilab, existe una herramienta llamada Xcos orientada a la simulación de sistemas de control. Esta herramienta contiene todos los bloques necesarios para esta asignatura; además, la nomenclatura y la forma de funcionar los mismos es muy similar a la del Simulink.

1.4. Asignatura Ingeniería de Control II

La asignatura Ingeniería de Control II pretende como principales objetivos: el análisis y diseño de sistemas lineales representados en el espacio de estado de mediana complejidad, caracterizar los sistemas no lineales y las no linealidades típicas, aplicar técnicas de análisis y diseño de sistemas de control con datos muestreados, diseñar reguladores digitales aplicando algoritmos convencionales y no convencionales. El estudiante al terminar la asignatura debe ser capaz de dominar conceptos básicos vinculados al método de espacio de estado aplicados a sistemas SISO y MIMO; diseñar en el espacio estado sistemas SISO, observadores de estado y sistemas no lineales de control automático. Además el conocimiento del método de la función descriptiva, plano de fase y segundo método de Liapunov. Por último el principio de funcionamiento y rasgos fundamentales de sistemas de control óptimos, adaptables e inteligentes.

1.4.1. Temas de la asignatura Ingeniería de Control II

En las actividades prácticas de esta asignatura, Matlab permite definir la función de transferencia de sistemas discretos o el modelo de espacio de estados de los mismos y además cuenta con una función para transformar una en otra. Posee también funciones para analizar y diseñar sistemas discretos muy similares a las usadas en el análisis de sistemas continuos. Además de lo anterior, Matlab contiene las herramientas necesarias para analizar estabilidad absoluta y relativa, diseñar reguladores digitales, permite realizar operaciones con matrices y funciones para obtener la matriz de realimentación de los estados en el método de ubicación de polos, determinar si un sistema es controlable y/o observable; calcular la matriz de realimentación de los estados para minimizar un índice cuadrático, entre otras funciones que son muy útiles en esta asignatura. Para la simulación, Simulink posee los bloques con las no linealidades típicas y la variante discreta de todos los bloques que se usan en el análisis y diseño de sistemas continuos.

En el caso de Scilab, cuenta con funciones para: declarar funciones transferenciales discretas, declarar el modelo de espacio de estados de dichas funciones, convertir una función transferencial de continua a discreta, transformándola primero en la forma del modelo de espacio de estados, obtener el lugar geométrico de las raíces de sistemas discretos, mostrar el mapa de polos y ceros de sistemas discretos con la información necesaria del mismo. Por otro lado, Scilab cuenta también con las funciones necesarias para las operaciones con matrices, entre ellas están: multiplicar, invertir, rango y determinante, entre otras. Este programa también cuenta con funciones específicas para obtener la matriz de ganancia de realimentación de los estados a partir de la ubicación de polos deseados, determinar si un sistema es observable y/o controlable, así como

calcular la matriz de realimentación de los estados para minimizar un índice cuadrático. En el caso del Xcos, el mismo posee bloques con alinealidades típicas como son: backlash, zona muerta, saturación y relé.

1.5. Simulación de sistemas, Xcos

Scilab tiene un entorno similar a Simulink de Matlab para la simulación de sistemas dinámicos y resolución de sistemas de ecuaciones diferenciales llamado Xcos. Este entorno posee varios paquetes que incluye algunas herramientas para simulación sencilla de circuitos eléctricos, sistemas de control, termo hidráulica, etc; necesarios para la preparación de las actividades prácticas de las asignaturas Modelado y Simulación e Ingeniería de Control II.

1.6. Consideraciones finales

Luego de realizar un análisis de la literatura consultada, se arriban a las siguientes consideraciones.

El uso de algún software como medio auxiliar de enseñanza en las asignaturas de la disciplina Sistemas de Control, es imprescindible por la complejidad de los cálculos, las simulaciones, los métodos de diseño que requieren de gráficos en su mayoría difíciles de obtener a mano.

Durante años se ha utilizado Matlab por su potencia, siendo el más popular y útil, pero es de suma importancia la migración hacia software libre para darle solución a la problemática económica que trae consigo el uso del software privado.

Además se presentaron los principales temas que tratan las asignaturas Modelado y Simulación e Ingeniería de Control II, y en los capítulos siguientes se abordarán las posibles soluciones que brinda Scilab para alcanzar los objetivos propuestos.

CAPÍTULO 2

ACTIVIDADES PRÁCTICAS DE ING. DE CONTROL II SOBRE SCILAB

2.1. Introducción

En el presente capítulo se presentan las actividades prácticas de la asignatura Ingeniería de Control II sobre el software Scilab, como herramienta de apoyo para impartir la asignatura. Se demuestra que Scilab es capaz de sustituir al potente Matlab, en el cual se desarrollan las prácticas en la actualidad, con resultados semejantes y fácil desempeño de sus usuarios.

La asignatura que se aborda en este capítulo realiza seis actividades prácticas, en las cuales se abarcan los principales temas a estudiar.

2.2. Solución de las actividades prácticas de la asignatura Ingeniería de Control II

La asignatura Ingeniería en Control II constituye una continuación de Ingeniería de Control I, por lo que los conocimientos adquiridos en la primera son fundamentales para la segunda y sirven de apoyo para entender otros temas más complejos.

2.2.1. Actividad Práctica 1

El primer tema de la asignatura corresponde a *Sistemas alineales*; este laboratorio está conformado por 3 ejercicios con el objetivo de estudiar el efecto de diferentes alinealidades sobre el comportamiento de sistemas de control sencillos mediante simulación.

Ejercicio 1

Conforme un esquema en el que pueda comprobar la relación Salida-Entrada de diferentes alinealidades. Para ello:

1. Tome una fuente sinusoidal, un graficador xy y las alinealidades juego libre de engranes, saturación, zona muerta, y relay.

2. Colocando como entrada la onda sinusoidal y ajustando debidamente los parámetros de cada alinealidad simulada podrá observar en el graficador xy la forma de la curva que caracteriza la alinealidad.
3. Añada al esquema una pantalla en la que pueda obtener en el tiempo la entrada y la salida del elemento no lineal correspondiente.

Colocando como entrada la onda sinusoidal y ajustando debidamente los parámetros de cada alinealidad simulada se observa en el graficador xy, la forma de la curva que caracteriza cada alinealidad, ver Figura 2-1.

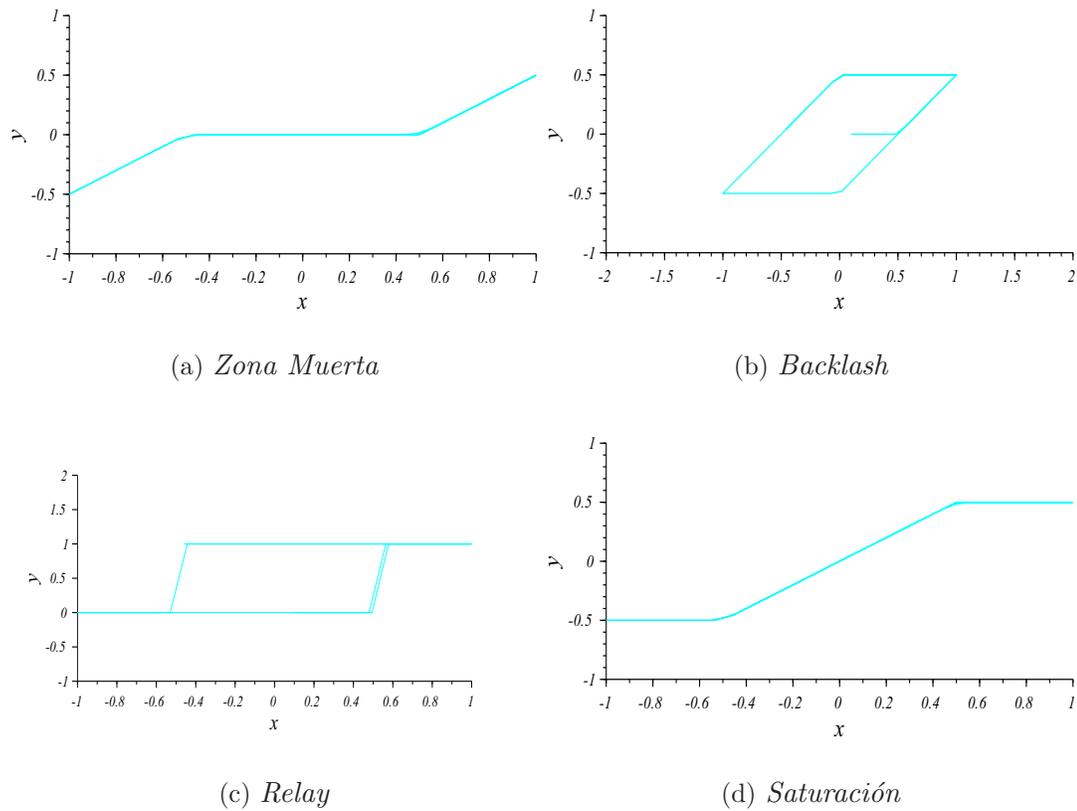
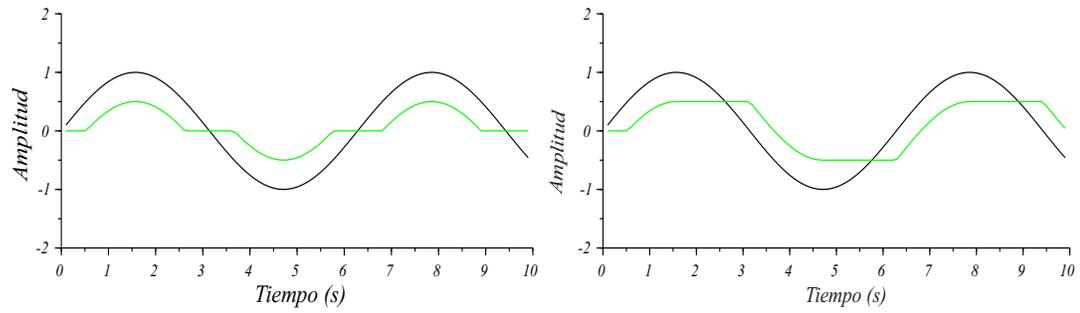


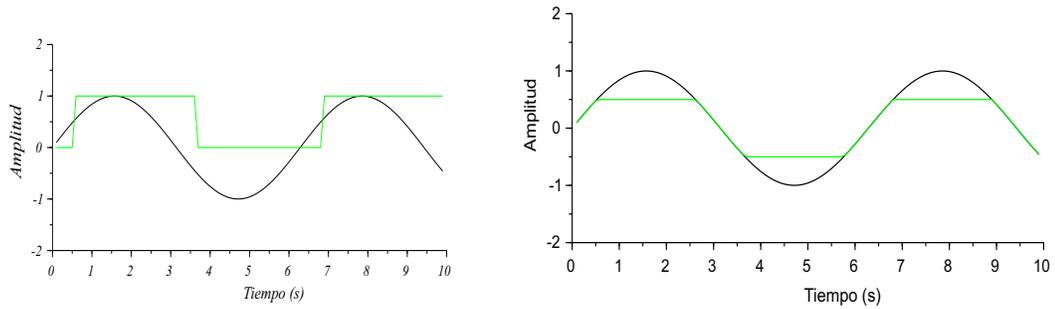
Figura 2-1: Forma de curva de cada alinealidad

Al esquema se le añadió una pantalla para obtener en el tiempo, la entrada y la salida del elemento no lineal correspondiente, tal como se muestra en la Figura 2-3.



(a) Zona Muerta

(b) Backlash



(c) Relay

(d) Saturación

Figura 2-2: Forma de curva de cada alinealidad

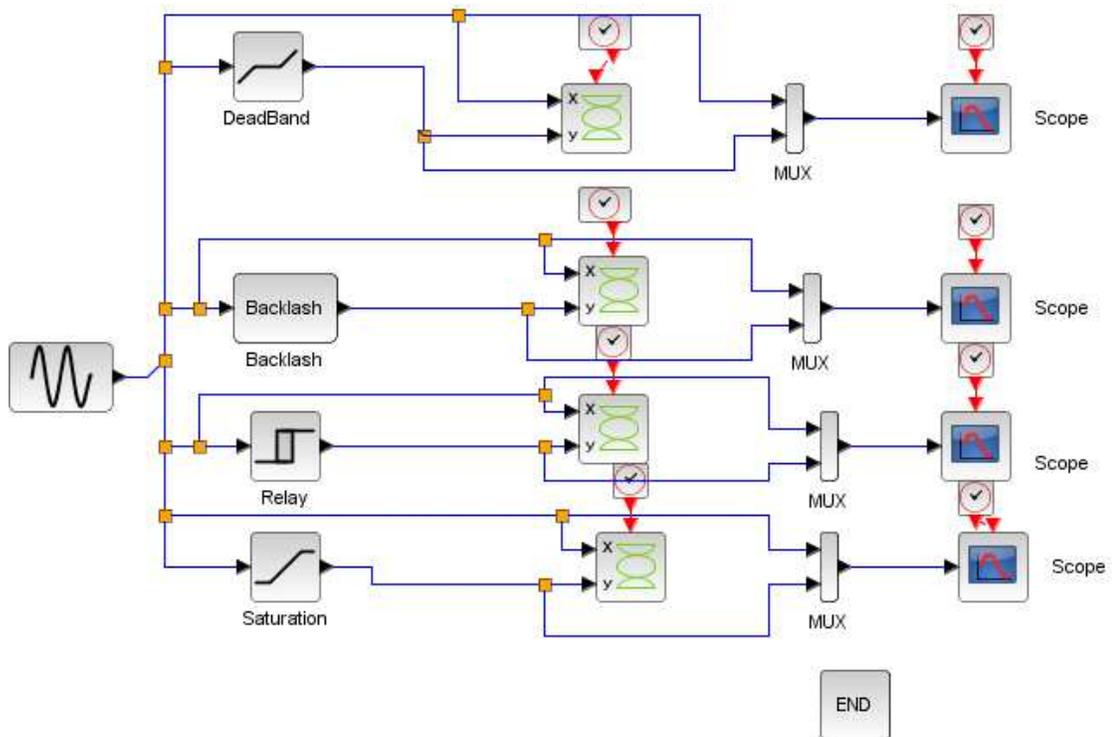


Figura 2-3: Esquema Salida-Entrada de diferentes alinealidades.

La Figura 2-4 muestra el comportamiento en el tiempo la entrada y salida de la alinealidad zona muerta donde se observa que no hay salida para entradas que caen dentro de la amplitud de zona muerta

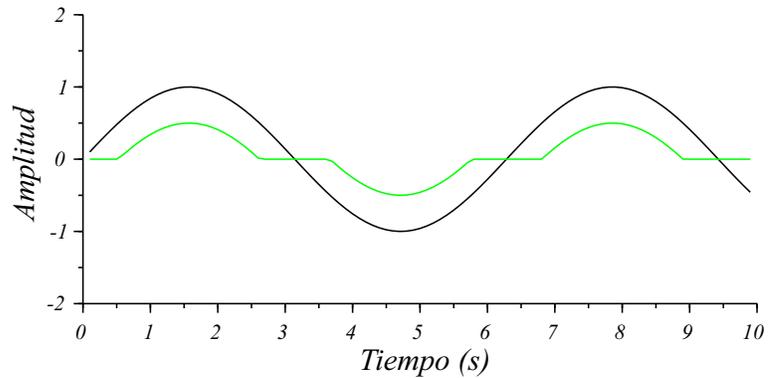


Figura 2-4: Gráfica de entrada-salida para zona muerta en el tiempo.

En la Figura 2-5 se observa en el tiempo la entrada y salida de la alinealidad Backlash. Se muestra que si la entrada cambia de dirección, el cambio inicial no tiene efecto en la salida; este fenómeno se presenta en engranajes con desgaste.

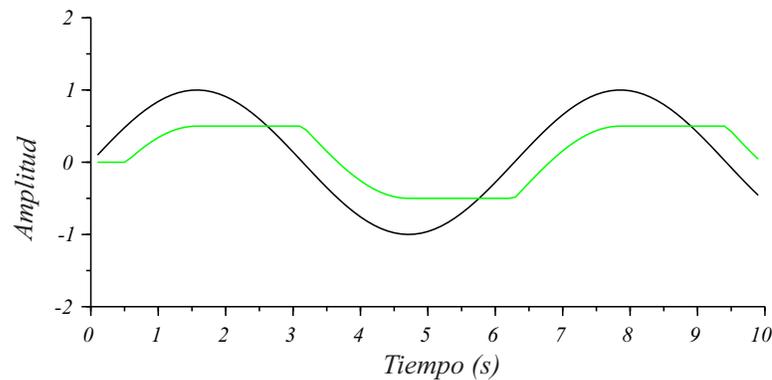


Figura 2-5: Gráfica de entrada-salida para backlash en el tiempo.

La Figura 2-6 muestra el comportamiento en el tiempo de la entrada y salida de la alinealidad Relay. Permite cambiar la salida entre dos valores específicos; cuando el relay está encendido se mantiene de esta manera hasta que la entrada caiga por debajo del valor del switch point.

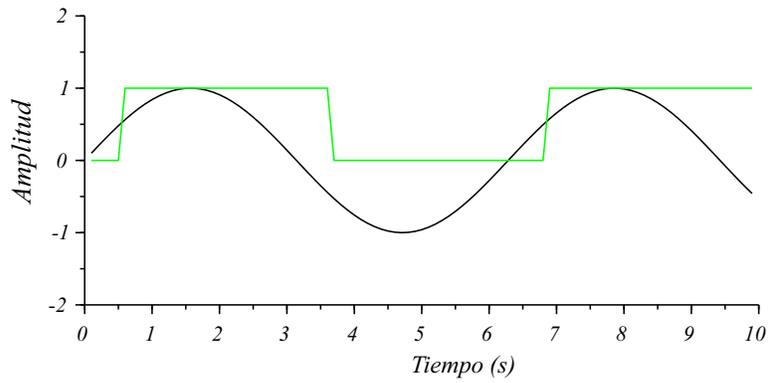


Figura 2-6: Gráfica de entrada-salida para relay en el tiempo.

La Figura 2-7 muestra el comportamiento en el tiempo de la entrada y salida de la alineidad saturación. Como se observa en la figura es para señales de entrada pequeñas, la salida de un elemento de saturación es proporcional a la entrada. Para señales mayores de entrada, la salida no se incrementa proporcionalmente y finalmente para valores muy elevados de las señales de entrada, la salida se mantiene constante.

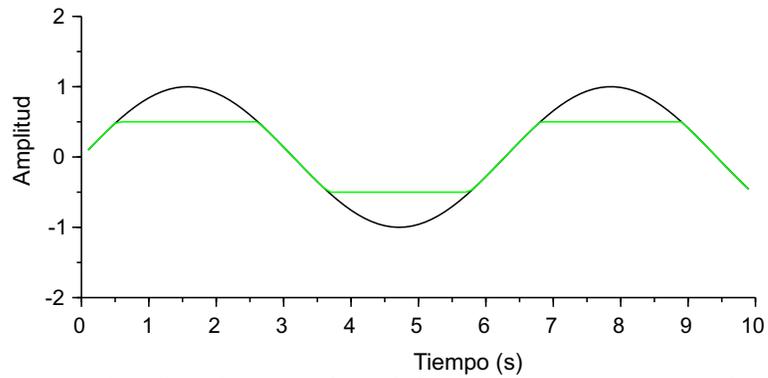


Figura 2-7: Gráfica de entrada-salida para saturación en el tiempo.

Ejercicio 2

Análisis del efecto de las alinealidades sobre sistemas lineales.

Se desea ver la respuesta en el plano cc si la entrada se conforma por la suma de tres funciones paso:

$$r(t) = u(t) + 2u(t - 2) + 3u(t - 3)$$

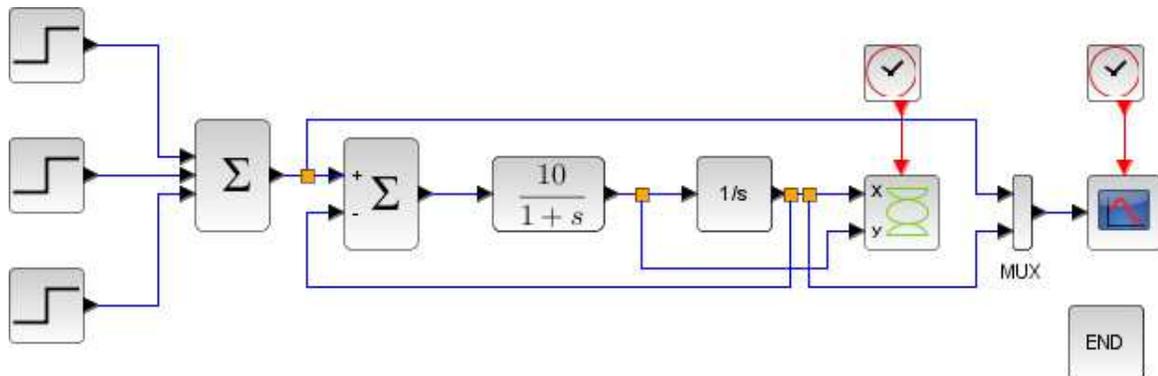


Figura 2–8: Diagrama en bloques el sistema ante entrada de tres funciones tipo paso.

En la Figura 2–8, se observa el diagrama en bloques del sistema al cual se le da 3 entradas tipo paso para desfasarlo en el tiempo, cambiando solo el inicio y la magnitud del paso.

Como se observa en la Figura 2–9, el sistema es capaz de seguir la entrada con buena respuesta transitoria y cero error en estado estable.

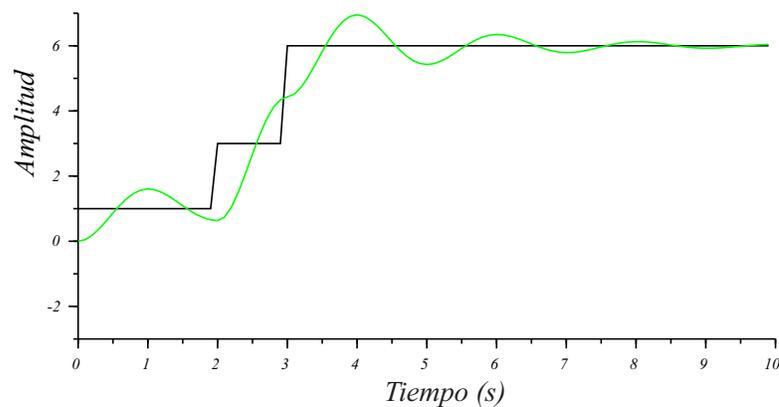


Figura 2–9: Respuesta del sistema.

En la gráfica xy , Figura 2–10, se observa la respuesta en el plano cc cuando la entrada se conforma por la suma de tres funciones tipo paso, propio de un foco estable.

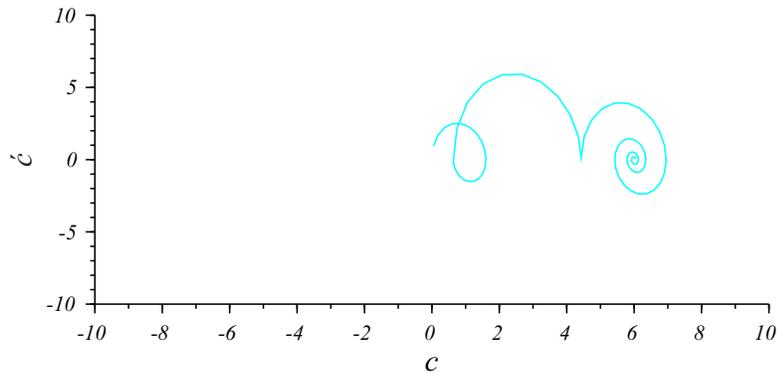


Figura 2-10: Respuesta en el plano cc' .

Ejercicio 3

Análisis utilizando bloques alineales. Con el mismo sistema alineal introduzcamos con una sola entrada escalar un bloque de saturación, Figura 2-11. El paso de entrada de 5 unidades y la saturación de $\pm 0,5$.

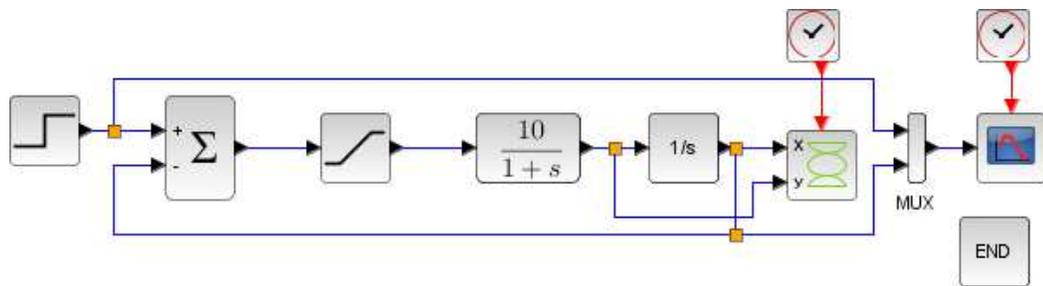


Figura 2-11: Diagrama en bloques de sistema alineal con una entrada paso y alinealidad saturación.

En la Figura 2-12 se observa la respuesta transitoria ante una entada paso de 5 unidades y un bloque saturación.

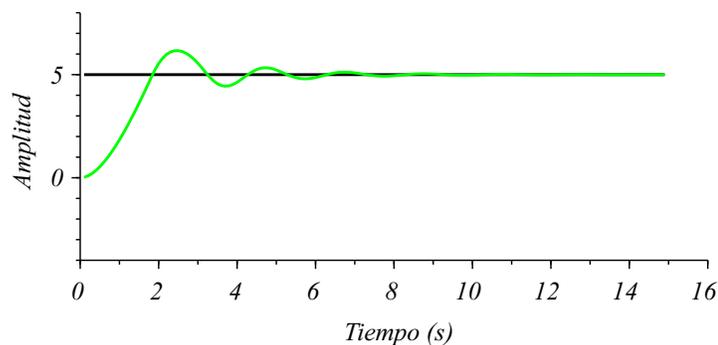


Figura 2-12: Respuesta del sistema ante una entrada paso y alinealidad saturación.

2.2.2. Actividad Práctica 2

Análisis de sistemas discretos. Cuenta de dos ejercicios con el objetivo de adquirir habilidades en el uso de medios computacionales para el análisis de Sistemas Discretos.

Para obtener funciones discretas equivalentes de funciones continuas se requiere, además de la función continua, el tiempo de muestreo y el método de aproximación.

Ejercicio 1

Obtener las equivalencias discretas del sistema mostrado utilizando diferentes tiempos de muestreo y métodos. Analice las diferencias obtenidas al utilizar diferentes tiempos de muestreo.

$$S_{ysc}(s) = \frac{s + 10}{(s + 1)(s + 2)(s + 3)}$$

Para obtener el sistema discreto equivalente de un sistema continuo por el método ZOH utilizamos en Scilab la función *dscr*, ver Anexo [A.0.1](#), la cual tiene como argumentos la función transferencial y el período de muestreo.

$$[sld] = dscr(sl, dt)$$

Parámetros:

sl : sistema lineal

dt : período de muestreo

sld : sistema lineal discretizado

Al variar los tiempos de muestreo en 0,1, 0,5, 1 y 2; como se muestra en el Anexo [A.0.1](#) no se ve afectado el orden del sistema, solo los coeficientes.

Utilizando Tustin como método de muestreo para el mismo sistema se observa que no aumenta el orden del sistema Anexo [A.0.1](#), solo el orden del numerador con respecto al método anterior.

La función que permite discretizar una función de transferencia continua mediante el método de Tustin es : *cls2dls*, que tiene como argumentos el sistema continuo y el período de muestreo.

$$[sl1] = cls2dls(sl, dt)$$

Parámetros:

sl : sistema lineal

dt : período de muestreo

sl1 : sistema lineal discretizado

Al igual que en el método anterior al variar el tiempo de muestreo no cambia el orden del sistema, solo se ven afectados los coeficientes, ver Anexo [A.0.1](#).

Ejercicio 2

Obtenga la respuesta del sistema discreto con la equivalencia “Tustin”, ante impulso y paso.

Scilab da la posibilidad de obtener la respuesta de sistemas discretos a entradas específicas mediante el comando *dsimul*, tanto para entrada tipo paso como para impulso.

Al aplicar estas entradas al sistema discreto por el método de Tustin, ver Anexo A.0.2, obtenemos un seguimiento bastante aproximado de la señal continua, tanto para la entrada paso, Figura 2–13, como para el impulso, Figura 2–14.

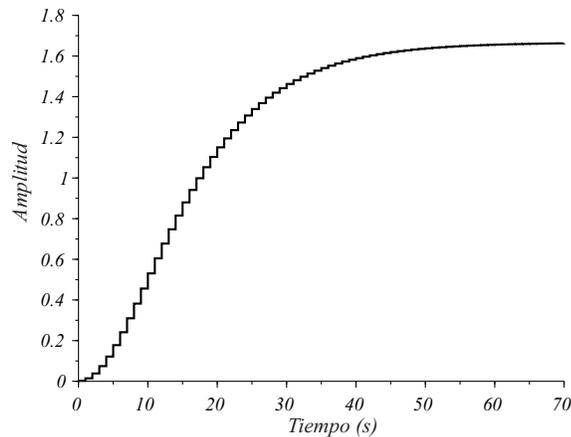


Figura 2–13: Respuesta del sistema discreto ante una entrada paso.

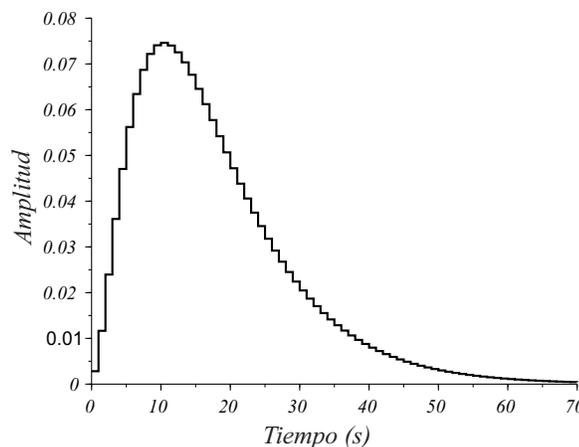


Figura 2–14: Respuesta del sistema discreto ante una entrada tipo impulso.

2.2.3. Actividad Práctica 3

La Actividad Práctica No.3 tiene como objetivo adquirir habilidades en el diseño y simulación de controladores digitales con diferentes algoritmos de control. Este laboratorio consta de un ejercicio que abarca los métodos de control digital más conocidos que definen la relación entrada-salida, los cuales son: Deadbeat, Dahlin y Kalman.

Ejercicio 1

Tomando en cuenta la función de transferencia de un motor de corriente directa:

$$G(s) = \frac{1550}{0,024s + 1} \quad T = 0,1s$$

Obtenga el controlador digital por los métodos estudiados.

Para establecer la forma deseada de la respuesta en lazo cerrado $T(z)$, existen restricciones que permiten que $D(z)$ sea físicamente realizable, estable y para que el sistema responda a una entrada específica con cero error en estado estable.

Estas restricciones se definen en 4 reglas:

1. El exceso de polos en $T(z)$ debe ser al menos igual al exceso de polos de $G(z)$.
2. $T(z)$ debe contener entre sus ceros, los ceros de $G(z)$, ubicados sobre o fuera del círculo de radio unidad.
3. $[1 - T(z)]$ debe contener entre sus ceros los polos de $G(z)$ que se ubiquen sobre o fuera del círculo de radio unidad.
4. $[1 - T(z)]$ debe tener un factor $(1 - Z^{-1})^{q+1}$. Siendo:
 - $q=1$ para entrada paso.
 - $q=2$ para entrada rampa.

Para el algoritmo Deadbeat

El algoritmo Deadbeat establece como principio que la respuesta a una entrada tipo paso escalón se estabiliza en el mínimo número posible de períodos de muestro, sin error ni oscilaciones entre un período y el próximo. En el Anexo [A.1.1](#) se calcula el controlador digital por el método Deadbeat.

Para el algoritmo Kalman

Kalman estableció un algoritmo que ofrece una respuesta sin oscilaciones y por su estructura es más sencillo.

Note que el controlador digital obtenido Anexo [A.1.2](#) por el algoritmo de Kalman coincide con el Deadbeat ya que no existe retardo de transporte.

Para el algoritmo Dahlin

Si $\Theta = 0,5s$, $\mu = 1s$ y $T = 0,1s$. $k = 5$

Dahlin estableció su algoritmo planteando que la respuesta a una entrada paso sea similar a la de un sistema de primer orden con retardo y constante de tiempo prefijados.

Donde Θ es el retardo de transporte en segundos y μ es la constante de tiempo, también en segundos.

El controlador obtenido se muestra en el Anexo [A.1.3](#).

2.2.4. Actividad Práctica 4

Esta actividad práctica está destinada al *Diseño ubicando polos*. Presenta un ejercicio que tiene como objetivo adquirir habilidades para la determinación de la controlabilidad y la observabilidad de un sistema, así como, las ganancias del controlador y observador necesarios para ubicar los polos en la posición deseada.

Ejercicio 1

Sea el sistema descrito por:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v \quad (2.1)$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (2.2)$$

Determine:

1. Si el sistema es o no controlable
2. Si el sistema es o no observable
3. La ganancia de retroalimentación de los estados que ubica los polos dominantes en $-0,5 \pm j0,8$
4. Determine el observador de estado
5. Simule un sistema de segundo orden con los polos indicados y compárelo con el resultado de ubicar los polos.

Posible solución:

1 Matriz de controlabilidad: $M = cont_mat(A, B)$

Para determinar el rango de M: $rank(M) = 3$

El sistema es controlable ya que orden es igual al rango de la matriz.

2 Para la observabilidad: $N = obsv_mat(A, C)$

Rango de N: $rank(N) = 3$

El sistema es observable ya que el orden es igual al rango de la matriz.

3 $P = [-0.5 + j0.8, \quad -0.5 - j0.8, \quad -5]$

Se coloca el tercer polo 6 veces alejado de los polos dominantes. Para obtener la ganancia podemos usar *acker* Anexo [A.2](#).

Obteniéndose la matriz de ganancia: $K = [2.67 \quad 1.89 \quad 0]$

4 Para obtener el observador de estado podemos ubicar sus polos alejados 8 veces.

$$P = [-4, -4, -4]$$

Como la dinámica es $(A - KeC)$ se puede aplicar ackerman con A' y C' Anexo A.2.

Obteniéndose la matriz de ganancia del observador:

$$K_e = \begin{bmatrix} 8 \\ 14 \\ -8 \end{bmatrix} \tag{2.3}$$

5 La Figura 2-15 muestra el sistema de segundo orden, con las ganancias.

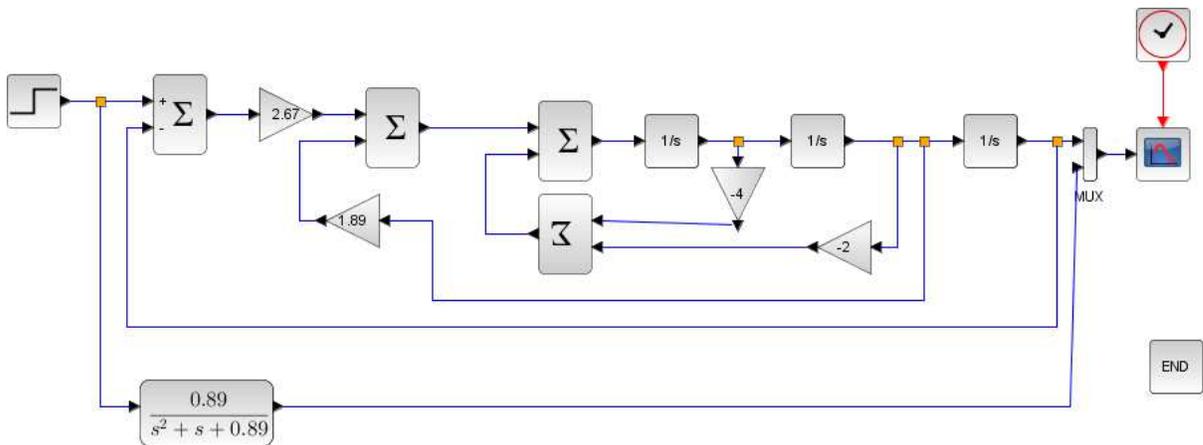


Figura 2-15: Sistema de segundo orden.

Mientras que la Figura 2-16 muestra el sistema con el observador de estado.

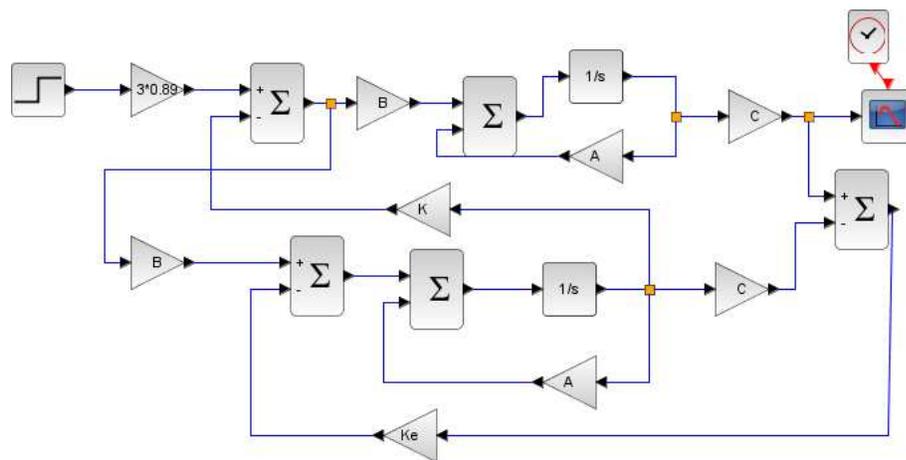


Figura 2-16: Sistema con observador de estado.

2.2.5. Actividad Práctica 5

Esta clase cuenta con un ejercicio, donde se estudiará el diseño de control óptimo con índices cuadráticos. Se comprobará mediante simulación las características que presenta la optimización de sistemas de control con índices cuadráticos.

Ejercicio 1

Determine la ley de control.

Dado:

$$A = \begin{bmatrix} -1 & 1 \\ 0 & -2 \end{bmatrix}; \quad B = \begin{bmatrix} -1 & 1 \\ 0 & -2 \end{bmatrix}$$

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad R = 1$$

El problema general de control óptimo consiste en lograr que la combinación de variables se mantenga en su valor extremo (óptimo), mínimo o máximo, independientemente de las variaciones causadas por los disturbios y cambios de parámetros.

Scilab permite obtener la matriz de ganancia K que optimiza el sistema a través de la función lqr , ver respuesta del ejercicio en Anexo [A.3](#), donde:

$$[K, X] = lqr(P12)$$

Parámetros:

P12 : Representación en espacio estado del sistema lineal dinámico

K : ganancia óptima

X : Solución de la ecuación de Riccati

2.2.6. Actividad Práctica 6

Los sistemas de control adaptativo por modelo de referencia permiten obtener respuestas deseadas, aún en procesos inestables o no lineales. Este actividad cuenta con 3 ejercicios sobre diseño de control adaptativo.

Ejercicio 1

La Figura [2-17](#) muestra un control adaptable por modelo de referencia *MRAC*. El sistema opera sobre el principio de ajustar los parámetros de control, con el objetivo de que la salida de la planta siga la salida del modelo a variaciones en la entrada de referencia, como se observa en Figura [2-18](#).

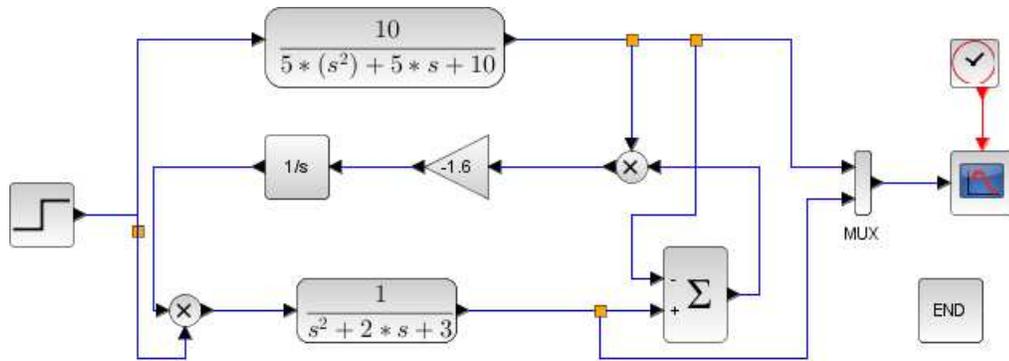


Figura 2-17: Sistema con control adaptable por modelo de referencia MRAC.

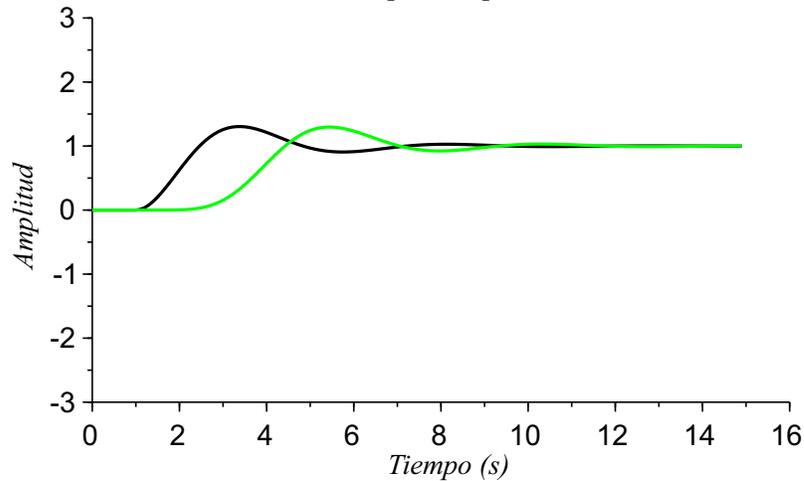


Figura 2-18: Respuesta para control por modelo de referencia MRAC.

Ejercicio 2

Para el sistema inestable descrito por:

$$G(s) = \frac{3}{s - 1}$$

Comenzar por una ganancia de adaptación igual a 10 y variar para analizar la estabilidad y velocidad de la respuesta, la salida se observa en Figura 2-20.

El diagrama en bloques esta dado por la Figura 2-19:

Al variar la ganancia de adaptación del modelo de -5 a -50 se obtiene valores aceptables de tiempos de subida y establecimiento de acuerdo con la respuesta esperada.

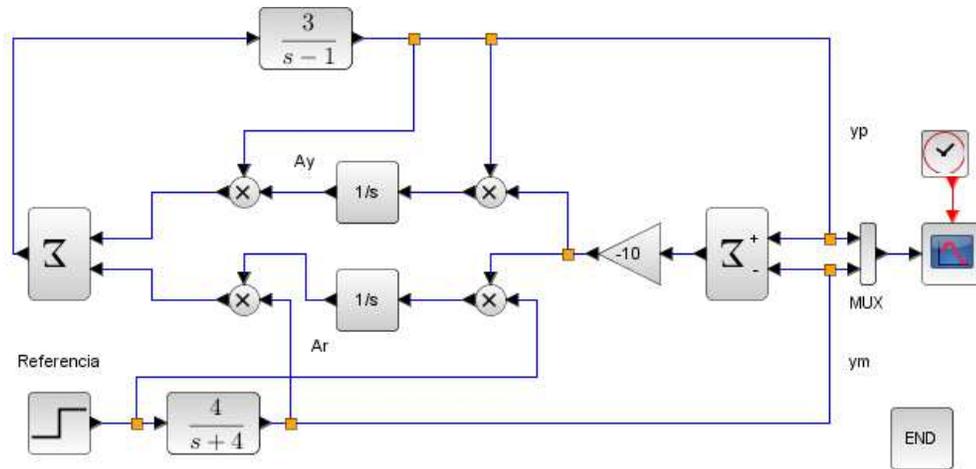


Figura 2-19: Control adaptable para sistema inestable.

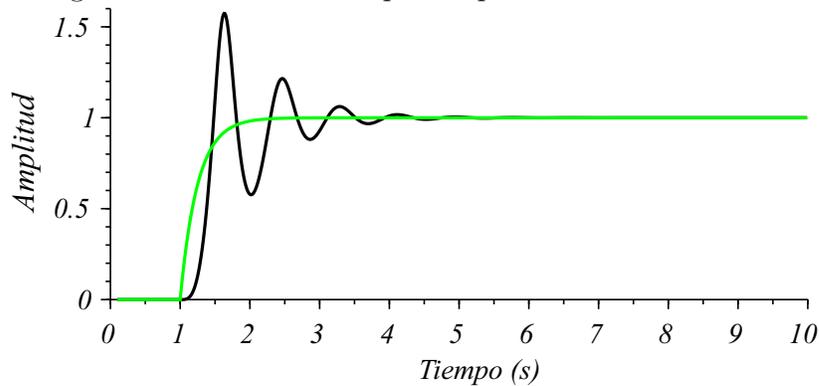


Figura 2-20: Sistema inestable.

Ejercicio 3

Modele el sistema alineal descrito por:

$$\dot{y} = y + (y^2) + 3u \tag{2.4}$$

La respuesta del sistema alineal se observa en Figura 2-22.

2.3. Consideraciones finales del capítulo

Todos los ejercicios que se incluyen en las actividades prácticas de la asignatura Ingeniería de Control II pueden realizarse con Scilab y aunque no se incluye aquí, la comparación porque no es objetivo del trabajo, se comparó con los resultados que hoy se obtienen con Matlab y son semejantes.

A los profesores de la asignatura les corresponderá, ya sea a través de orientaciones previas o en las propias actividades, las explicaciones pertinentes sobre la implementación de las instrucciones y los esquemas utilizados.

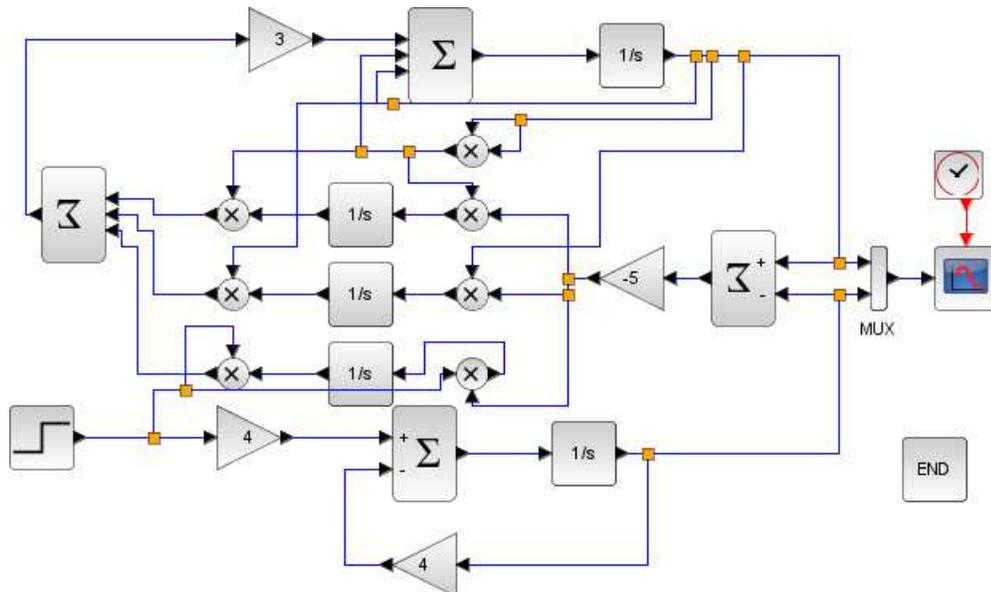


Figura 2-21: Control adaptable para sistema alineal.

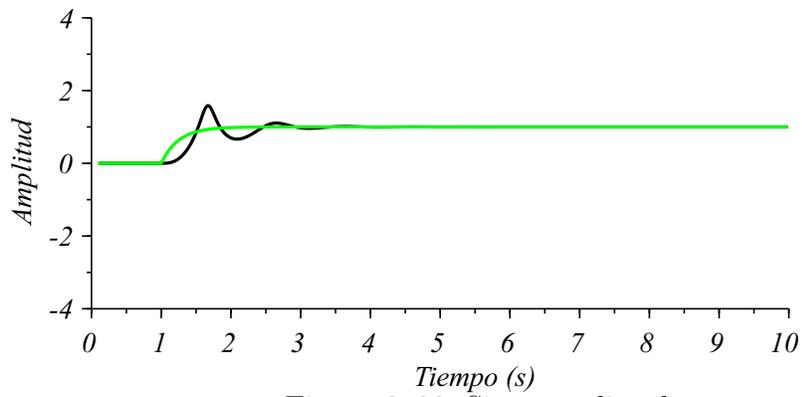


Figura 2-22: Sistema alineal.

CAPÍTULO 3

ACTIVIDADES PRÁCTICAS DE MODELADO Y SIMULACIÓN SOBRE SCILAB

3.1. Introducción

En este capítulo se elaboran las actividades prácticas de la asignatura Modelado y Simulación mediante el apoyo del software Scilab. Se plantean en caso de ser necesario las modificaciones que hayan sido implementadas.

3.2. Solución de las actividades prácticas de la asignatura Modelado y Simulación

En esta asignatura las prácticas de laboratorio se orientan al uso de la red de computadoras de la facultad y el software de simulación disponible en la misma, en particular, el Matlab-Simulink, el cual se presenta de forma general como un paquete de software profesional orientado a la simulación. Este paquete contiene bloques imprescindibles para la simulación de sistemas de control, entre los principales bloques están: función transferencial, integrador, derivador, controlador PID, operaciones lógicas y aritméticas, discontinuidades, entradas tipo paso y rampa, retardos, bloques para analizar los resultados de la simulación, entre otros ([Lanza, 2015](#)).

3.2.1. Actividad Práctica 1

La primera actividad práctica de la asignatura es una introducción al “Control System Toolbox”. Esta actividad está comprendida por cuatro ejercicios donde se pretende obtener modelos en espacio estado y funciones de transferencia utilizando la ayuda de un software matemático.

Los principales comandos que forman parte del Toolbox Control System son:

- Para representación de modelos: *syslin*, *zpk*, *tf2ss*.
- Para el análisis del comportamiento de modelos: *csim*, *g1/.g2* para realimentación.

Ejercicio 1

Declare las siguientes funciones de transferencia:

1. $G(s) = \frac{2}{20s + 2}$
2. $G(s) = \frac{5}{(s^2 + 2)(5s + 3)}$
3. $G(s) = \frac{4s + 1}{s^3 + 10s^2 + 20}$
4. $G(s) = \frac{5}{s(s + 3)(s + 2)}$

En la teoría de control, se usan las funciones de transferencia para caracterizar las relaciones de entrada-salida de sistemas que se describen mediante ecuaciones diferenciales lineales invariantes con el tiempo. En Scilab para crear una función de transferencia es necesario escribir la siguiente línea de comando:

```
g = syslin('c', poly(n,'s','c'), poly(conv(m1, m2),'s','c'))
```

O se pueden escribir varias líneas de comandos como se muestra a continuación.

```
num = poly(n,'s','c');
```

```
den = poly(conv(m1, m2),'s','c');
```

```
g = syslin('c', num, den);
```

Donde:

n, *m1* y *m2*: Son vectores numéricos que contienen los coeficientes de los polinomios respectivamente, en potencias ascendentes de la variable simbólica 's'.

poly: Es la palabra predefinida por el software que indica la creación de un polinomio.

syslin: Indica que el sistema es lineal

La función de transferencia que da respuesta a estos incisos puede verse en el Anexo [B.0.1](#).

Aplicando el comando *zpk* permite llevar la función de transferencia a forma de ceros, polos y ganancias; ver Anexo [B.0.1](#).

Por último el comando *roots* nos permite obtener las raíces del sistema, ver Anexo [B.0.1](#).

Ejercicio 2

A partir de las funciones de transferencia creadas previamente en el ejercicio anterior, almacene en dos variables distintas el numerador y el denominador de las mismas.

Para extraer el numerador de una función de transferencia, Anexo [B.0.2](#), en Scilab basta con utilizar el comando:

```
numerador = sl.num
```

Donde:

numerador : variable que almacena el numerador de la función de transferencia.

sl : función de transferencia.

Para extraer el denominador se utiliza:

denominador = sl.den

2.1*Representación de modelos en espacio estado*

Sea el sistema en espacio estado:

$$\frac{dx}{dt} = Ax + Bu \quad y = Cx + Du \quad (3.1)$$

$$\dot{x}_1 = x_2 \quad \dot{x}_2 = -4x_1 - 2x_2 + 2u \quad y = x_1 \quad (3.2)$$

Representa en Scilab el modelo en espacio estado.

El comando *syslin* permite declarar tanto sistemas lineales, como sistemas en espacio estado, como se define en el Anexo B.

2.2

Obtenga la función de transferencia de los sistemas:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 4 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.3)$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 \end{bmatrix} \quad (3.4)$$

Para obtener la función de transferencia de un sistema en espacio estado se utiliza la función:

$$[h] = ss2tf(sl)$$

Donde:

sl : sistema en espacio estado

h : función de transferencia

La solución a este ejercicio puede verse en el Anexo [B.0.3](#).

Ejercicio 3

Encuentre la FTLC de los siguientes sistemas:

$$a) FTD1 = \frac{2}{20s + 2} \text{ realimentación} = 4.$$

$$b) FTD2 = \frac{2}{20s + 2} \text{ realimentación} = \frac{5}{2s + 1}.$$

Mediante la sintaxis $Sl = Sl1/.Sl2$ se obtiene la función de transferencia de lazo cerrado, donde:

Sl1: ft de lazo directo.

Sl2: ft de realimentación.

Obteniéndose las funciones de transferencia de lazo cerrado:

$$FTLC1 = \frac{2}{20s + 10}$$

$$FTLC2 = \frac{4s + 2}{40s^2 + 24s + 12}$$

Ejercicio 4

Grafique la respuesta al paso de:

a) Sistema en lazo cerrado del inciso anterior.

$$b) FT1 = \frac{20}{s^2 + 2s + 1} \text{ para } \varphi > 1$$

$$c) FT2 = \frac{20}{s^2 + 0,5s + 1} \text{ para } \varphi < 1$$

Este ejercicio se realiza para demostrar como graficar varios sistemas en una misma gráfica, ver Figura [3-1](#). Para ello:

3.2.2. Actividad Práctica 2

En esta clase se realiza una introducción a la biblioteca Xcos, Toolbox Continuos. Se introducen las principales herramientas que se utilizarán en la confección de las actividades prácticas, como lo son:

- Continuos:
 - Función de transferencia
 - Integrador
- Sources:
 - Seno
 - Paso
 - Rampa
- Comunmente usados:

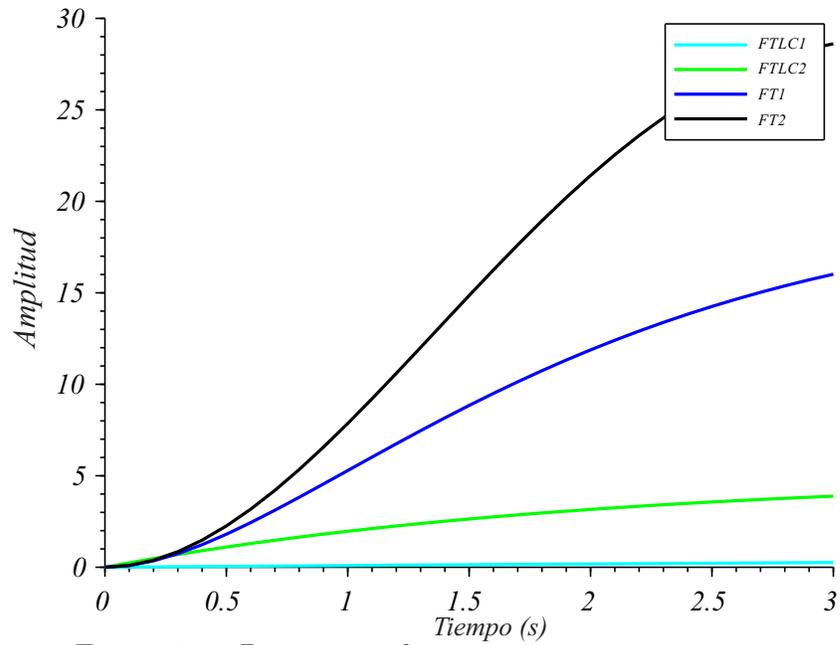


Figura 3-1: Respuesta al paso para varios sistemas.

- Ganancia
- Punto de suma
- Mux
- Demux
- Sinks:
 - Scope

Ejercicio 1

Implemente en Xcos el siguiente sistema de control, Figura 3-2. Vea el comportamiento del sistema ante entradas:

- paso
- rampa
- seno

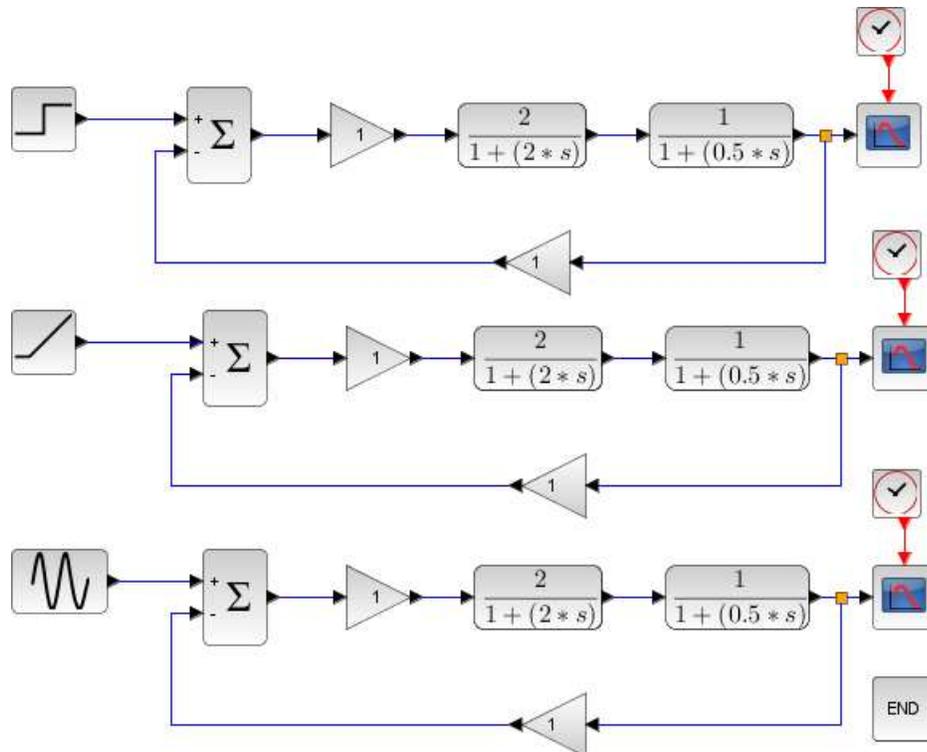


Figura 3-2: Diagrama en bloques para distintas entradas.

Este ejercicio pretende que los estudiantes se familiaricen con el ambiente de Xcos, aprendan a simular sistemas de control sencillos, además de plotear la salida del sistema ante diferentes entradas.

A pesar que la dinámica de la planta $G(s) = \frac{1}{0,5s + 1}$ es de primer orden, el sistema en su conjunto es de segundo orden y en efecto, la respuesta transitoria como se observa en la Figura 3-3.

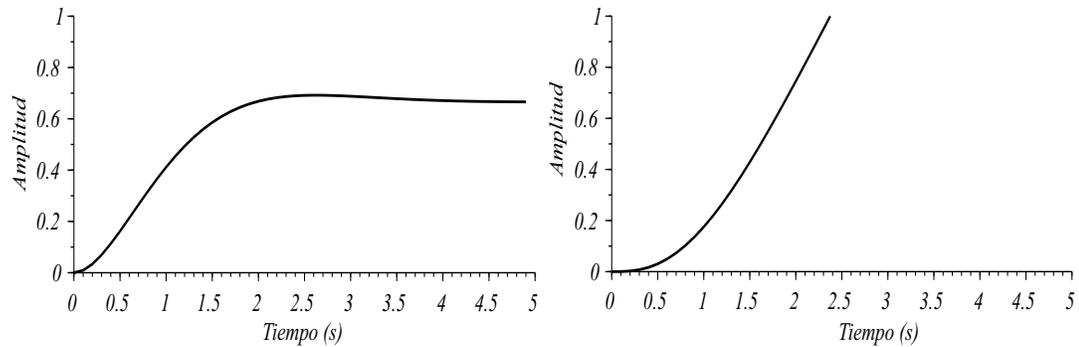
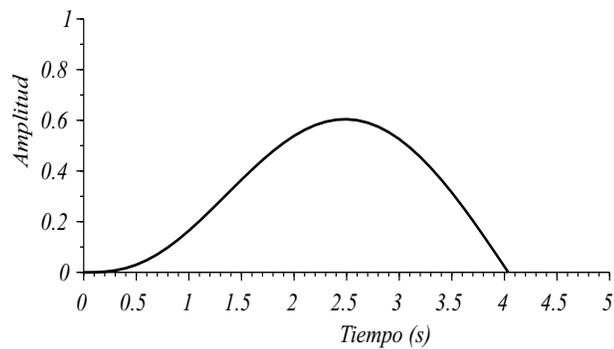
(a) *Entrada paso*(b) *Entrada rampa*(c) *Entrada sinusoidal*

Figura 3-3: Respuesta del sistema ante varias entradas

Ejercicio 2

Simule el comportamiento de los siguientes sistemas ante entrada paso de magnitud 2.

$$a) TF1 = \frac{25}{s^2 + 20s + 25}$$

$$b) TF2 = \frac{25}{s^2 + 5s + 25}$$

$$c) TF3 = \frac{25}{5s + 1}$$

Este ejercicio continua el trabajo en el ambiente Xcos, se simulan tres sistemas ante una entrada paso y se observa la respuesta en una pantalla (CSCOPE), ver Figura 3-4.

1. Plotee las respuestas en la misma gráfica.

En la Figura 3-5 se muestra el comportamiento de los tres sistemas ante una entrada paso en una misma gráfica.

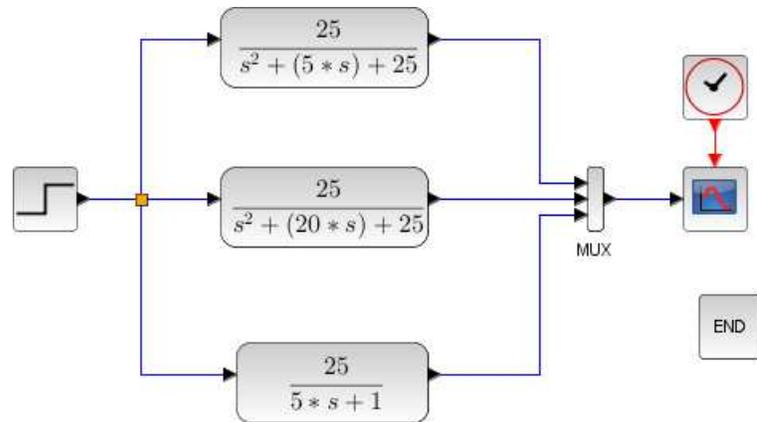


Figura 3-4: Diagrama en bloques para tres sistemas ante una entrada paso.

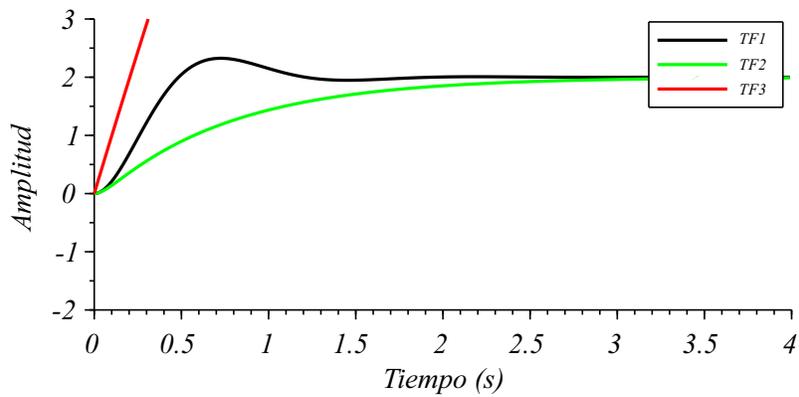


Figura 3-5: Respuesta ante entrada paso de los tres sistemas.

Ejercicio 3

Implemente en Xcos el siguiente sistema en espacio estado.

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -25 & -5 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 25 \\ -121 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}; \quad D = 0$$

- a) Observe el comportamiento de la salida ante entrada paso unitario.
- Realice la misma operación utilizando la función *ss2tf*.
- b) Repita el inciso a) para el sistema:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -25 & -5 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 25 \\ -121 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix};$$

Possible Solución

Tanto por Xcos como por *ss2tf* el sistema responde igual ante una entrada paso, como se observa en la Figura 3-6

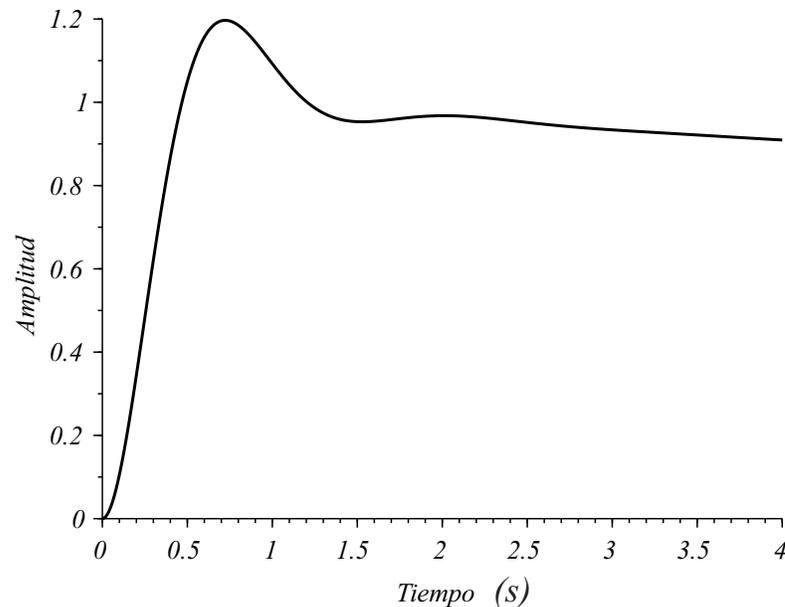


Figura 3-6: Respuesta ante entrada paso.

Para el inciso b) la matriz *C* cambia a una matriz 3 con 3, la respuesta transitoria ante entrada paso se observa en la Figura 3-7.

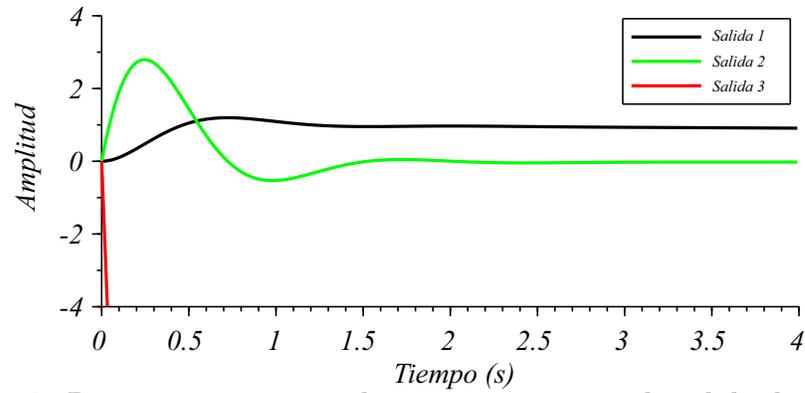


Figura 3–7: Respuesta ante entrada paso para matriz de salida de orden 3.

3.2.3. Actividad Práctica 3

Ejercicio 1

Obtener gráficamente la respuesta transitoria del sistema descrito por Figura 3–8:

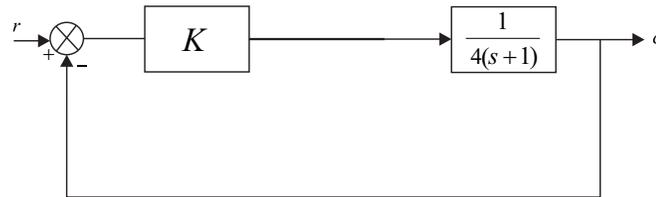


Figura 3–8: Sistema de primer orden.

En este ejercicio se trabaja con la respuesta transitoria de sistemas, además se enseña como crear entradas tipo paso, rampa e impulso, ver Anexo B.0.4 . Para ver la respuesta transitoria del sistema ver Figura 3–9.

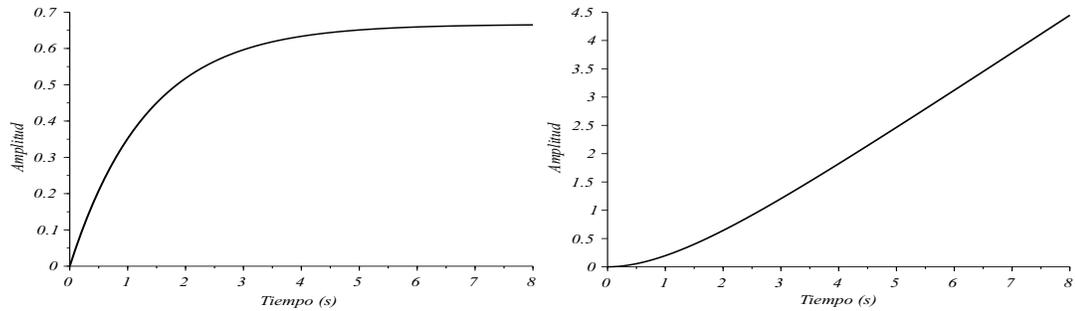
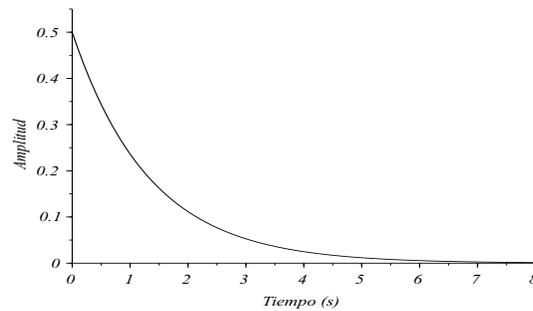
(a) *Respuesta ante entrada paso*(b) *Respuesta ante entrada rampa*(c) *Respuesta ante entrada impulso*

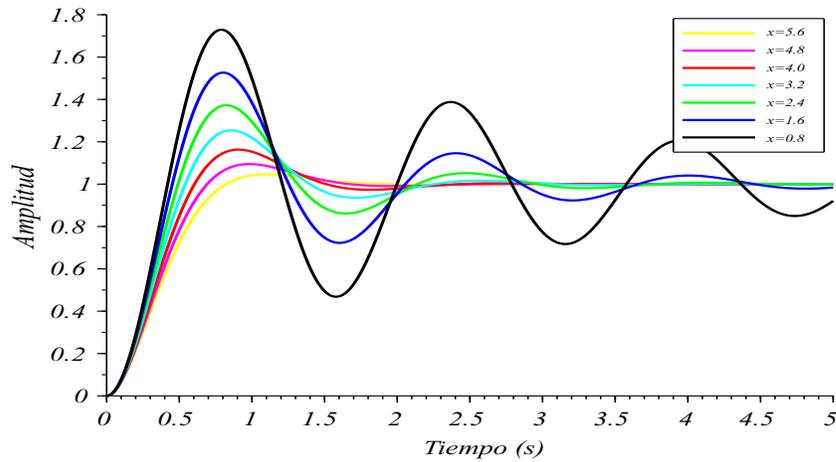
Figura 3–9: Respuesta del sistema ante varias entradas

Ejercicio 2

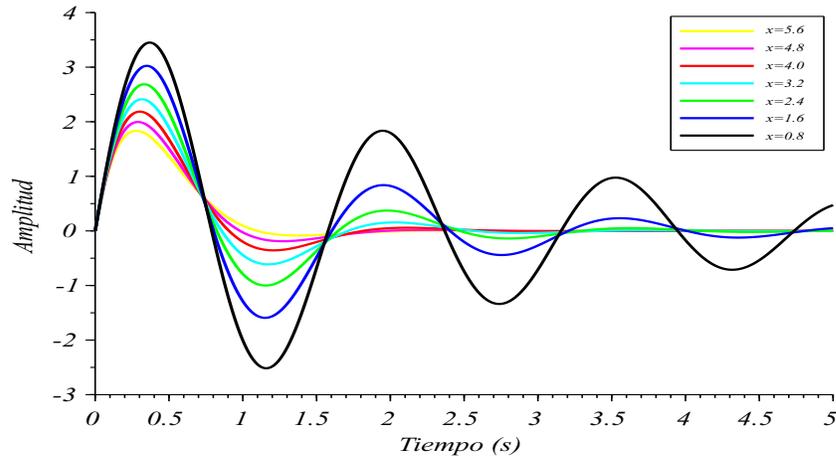
Obtenga la respuesta transitoria de un sistema de segundo orden cuya función de transferencia esta dada por la Ecuación 3.5, si x varía entre 5.6 y 0.8 a intervalos de 0.8.

$$\frac{Y(s)}{X(s)} = \frac{16}{s^2 + xs + 16} \quad (3.5)$$

En el ejercicio a medida que cambia el denominador del sistema, varía la ubicación del par de polos complejos conjugados y así va variando la respuesta transitoria del sistema, ver Figura 3–10.



(a) Respuesta ante entrada paso



(b) Respuesta ante entrada impulso

Figura 3-10: Respuesta del sistema ante entrada paso e impulso

Ejercicio 3

Para el sistema que se muestra en la Figura 3-11:

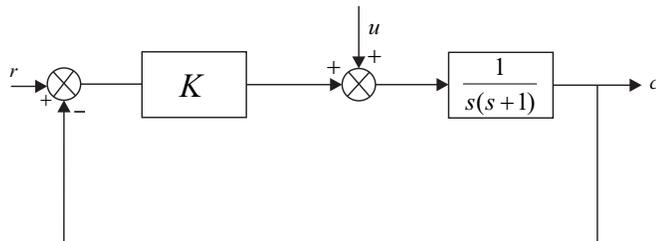


Figura 3-11: Diagrama en bloques del sistema.

- Obtenga en un mismo gráfico la respuesta ante entrada paso en la referencia.
- Obtenga la respuesta transitoria a una entrada paso en la señal perturbadora.

En la Figura 3–12 se muestra como, al variar la ganancia ante entrada de referencia, varía la respuesta transitoria del sistema, siendo para $K=0.1$ sobreamortiguado, $K=0.25$ críticamente amortiguado y para $K=10$ subamortiguado, ver Anexo B.0.6.

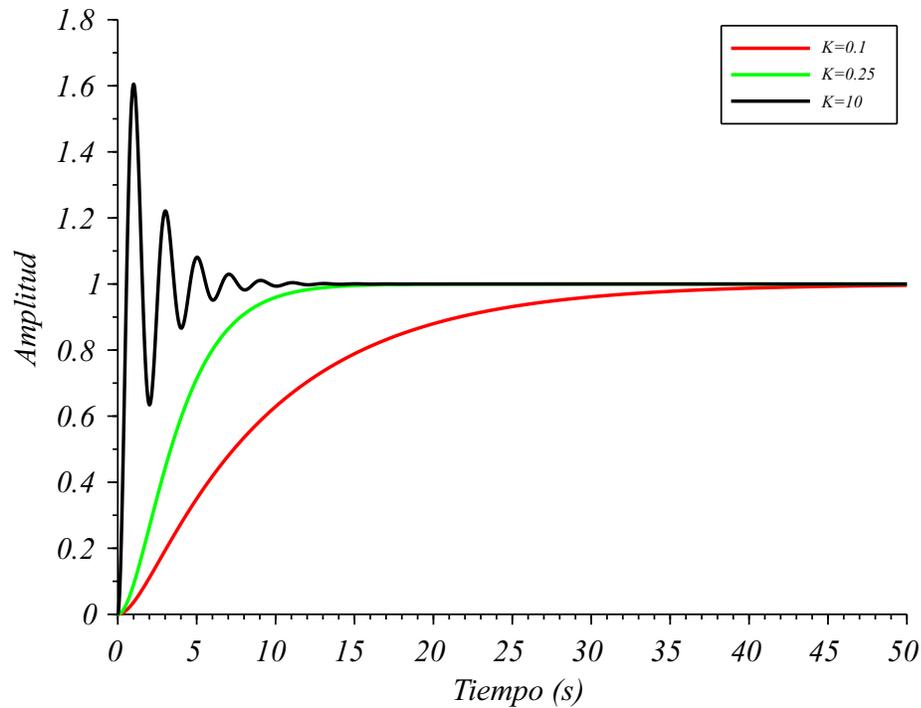


Figura 3–12: Respuesta ante entrada paso por referencia.

En la Figura 3–13 se observa la respuesta del sistema ante una entrada paso por la perturbación.

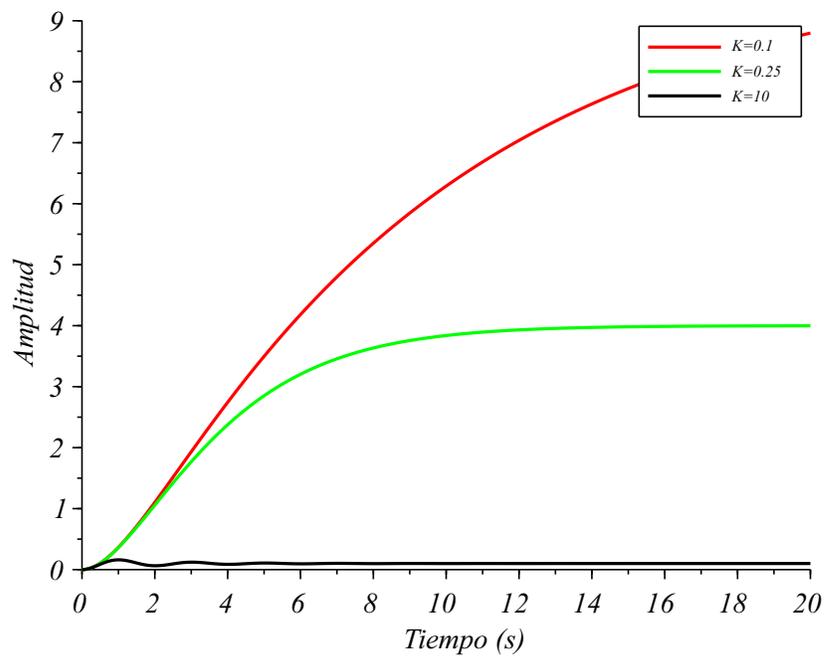


Figura 3–13: Respuesta paso por la señal perturbadora.

Para observar la respuesta para $K=10$, es necesario aplicar *Zoom* a la imagen, como se observa en la Figura 3-14.

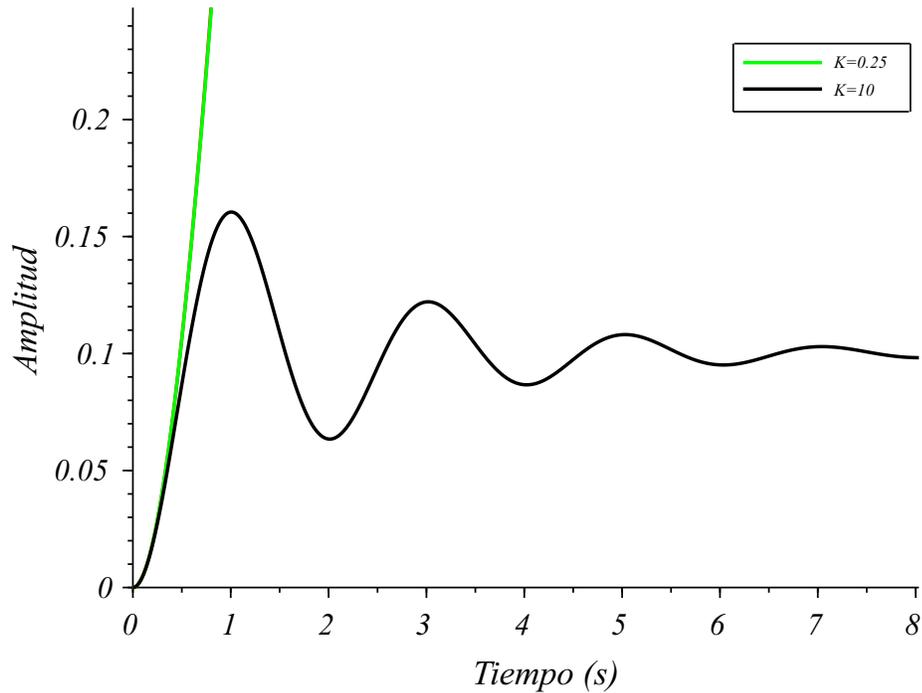


Figura 3-14: Zoom para poder observar la salida para $K=10$.

3.2.4. Actividad Práctica 4

La actividad práctica número 4 tiene como objetivo sistematizar el uso de la herramienta Xcos en la representación de sistemas continuos.

Ejercicio 1

Implemente el siguiente sistema 3.6 en espacio estado con bloques básicos.

$$\dot{x}_1 = x_1 + x_2 \quad \dot{x}_2 = x_3 + u_1 \quad \dot{x}_3 = x_2 + x_3 \quad y = x_1 + x_2 + x_3 \quad (3.6)$$

El objetivo de este ejercicio es aprender a montar en Xcos sistemas en Espacio Estado con bloques básicos, como se observa en la Figura 3-15.

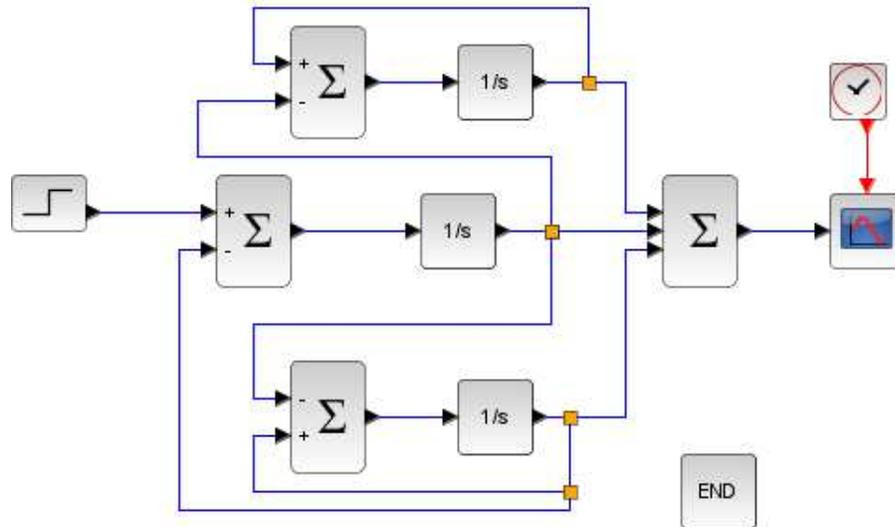


Figura 3-15: Diagrama en bloques del sistema en Espacio Estado.

3.2.5. Clase Práctica 11

Esta actividad práctica cuenta con dos ejercicios que tienen como objetivo desarrollar habilidades en la identificación paramétrica y no paramétrica.

La identificación paramétrica sirve para obtener un modelo aproximado a emplearse en el control; mientras que la no paramétrica se encarga de estimar estructuras paramétricas tales como: process model, ARX, ARMAX, OE ...

Ejercicio 1

Obtenga un modelo de primer orden con retardo utilizando el Scilab y aplicando la regla práctica para el sistema:

$$G(s) = \frac{10}{(s + 10)(s + 1)(s + 0,1)}$$

Considere la amplitud del paso igual a 5.

En este ejercicio se obtiene la función de transferencia y se le aplica un paso de magnitud 5 mediante la función *csim*, ver Anexo B.0.7, incluyendo las modificaciones realizadas en la tesis previa (Landaburo, 2016).

Posteriormente se calcula la ganancia del modelo PORT, variación de la salida sobre variación del mando: $k = \frac{\Delta C}{\Delta M}$

La constante de tiempo se calcula: $\tau = \frac{3}{2}(t_2 - t_1)$

Y el retardo de transporte $t_0 = t_2 - \tau$

Quedando el modelo PORT: $G(s) = \frac{10e^{-1,04s}}{10,15s + 1}$

Ejercicio 2

A partir del diagrama de Bode del sistema anterior encuentre la función de transferencia correspondiente utilizando la respuesta de frecuencia.

Primeramente se representa el diagrama de Bode mediante la función $bode(sys)$.

Donde:

sys : Función de transferencia.

Para la ganancia del sistema:

$$20\log k = 20$$

$$k = 10$$

Posteriormente las frecuencias de esquina: $W_1 = 0,1$, $W_2 = 1$ y $W_3 = 10$.

En cada lugar que cambia la pendiente, el inverso de cada W son las constante de tiempo del sistema. Quedando el sistema descrito por:

$$G(s) = \frac{10}{(s + 10)(s + 1)(s + 0,1)}$$

3.3. Análisis económico

Es conocido que en la actualidad existe un avance en la normalización de las relaciones entre Cuba y Estados Unidos. Aunque actualmente no se paga ningún tipo de licencia por el uso de Matlab, en un futuro si continúa el avance de esta normalización, Cuba tendría que reconocer leyes de mercado y de propiedad intelectual, lo que traería consigo tener que invertir gran cantidad de dinero por concepto de propiedad, licencia de software y actualizaciones. El costo actual de una versión de Matlab está valorada en \$ 2 150, por otra parte los diferentes Toolboxes que sean necesarios instalar tienen un elevado costo, como se observa en la Tabla 3-1, los datos fueron tomados de MathWorks. El laboratorio de la facultad de Ingeniería Automática cuenta con un total de 12 computadoras, con Matlab instalado en cada una de ellas, lo cual produciría un elevado costo para la Facultad, el cual se evitaría si se tomara la decisión de comenzar el uso del software libre Scilab.

Tabla 3-1: cambiar

| Producto | Costo en USD |
|----------------------------------|------------------|
| Matlab | \$ 2 150 |
| Control System Toolbox | \$ 1 000 |
| System Identification Toolbox | \$ 1 000 |
| SimPowerSystems | \$ 3 000 |
| Simulink Control Desing | \$ 1 000 |
| Robust Control Toolbox | \$ 1 850 |
| Model Predictive Control Toolbox | \$ 2 150 |
| Total del Costo | \$ 12 150 |

3.4. Consideraciones finales del capítulo

Con las actividades prácticas presentadas, queda en evidencia que el software Scilab es adecuado para la realización de todos los ejercicios que se incluyen en las actividades prácticas de la asignatura Modelado y Simulación.

Queda por parte de los profesores de la asignatura las explicaciones pertinentes de la implementación de las instrucciones y esquemas realizados.

CONCLUSIONES

Como resultado final de esta investigación se logra desarrollar las actividades prácticas de las asignaturas Modelado y Simulación e Ingeniería de Control II sobre Scilab. A partir de este resultado, se plantean las siguientes conclusiones generales:

- Es posible sustituir el Matlab, por el software libre Scilab, en las asignaturas Modelado y Simulación e Ingeniería de Control II sin afectar el cumplimiento de los objetivos de las actividades docentes que necesitan un software de apoyo.
- Todos los ejercicios que en este momento se realizan en las actividades prácticas de las asignaturas Modelado y Simulación e Ingeniería de Control II pueden realizarse en Scilab con resultados similares a los que se obtienen en Matlab.
- Se dispone de los códigos y esquemas necesarios para cada ejercicio de las actividades prácticas de las asignaturas Modelado y Simulación e Ingeniería de Control II sobre Scilab.

A partir de las conclusiones presentadas, se justifica plenamente la necesidad de la investigación y por tanto se satisfacen los objetivos del trabajo, quedando corroborada la hipótesis inicial establecida.

RECOMENDACIONES

Para establecer la necesaria continuidad que debe tener este trabajo se recomienda lo siguiente:

- Establecer un programa de migrar a Scilab, partiendo de la capacitación del personal docente y al mismo tiempo comenzar desde los primeros pasos que se dan con los estudiantes desde el primer año. Si un alumno ya domina Matlab es muy difícil en medio de una asignatura cambiarlo, ya que el uso de una herramienta no es el elemento fundamental sino los resultados que con ella se logren. Por otra parte como bien se señala, Matlab es el software más utilizado mundialmente, que tengamos que migrar no nos debe incapacitar para entender lo que se seguirá haciendo en el mundo sobre la base de Matlab.
- Continuar desarrollando las herramientas y funciones necesarias para completar la migración de la Disciplina Sistemas de Control hacia Scilab.

Conclusiones y recomendaciones.

REFERENCIAS BIBLIOGRÁFICAS

- Adell, J.; Bernabé, Y. (2007). Software libre en educación. *Tecnología educativa. Madrid: McGraw-Hill* pp. 173–195.
- Brewster, M. ; Gobbert, M. K. (2011). A comparative evaluation of Matlab, Octave, FreeMat, and Scilab on tara. Trabajo de maestría. University of Maryland, Baltimore County High Performance Computing Facility. Maryland, Estados Unidos.
- Costa, V. A. ; Scarabino, A. ; Idiart M. I. ; Knoblauch M. (2010). Enseñanza del álgebra lineal para alumnos de ingeniería aeronáutica: Experiencia motivadora. In: *World Congress and Exhibition Engineering 2010-Argentina*. IJCI. Buenos Aires, Argentina. pp. 173–195.
- Enterprises, Scilab et al. (2012). Scilab: Free and open source software for numerical computation. *Scilab Enterprises, Orsay, France* p. 3.
- Freire, J.; Brunet, K. S. (2016). Políticas y prácticas para la construcción de una universidad digital. *La cuestión universitaria* **4**(6), 85–94.
- Gómez, M. T.; Fernández-Delgado, F. C. (2012). El conocimiento libre: Una responsabilidad educativa. *Teoría de la Educación. Educación y Cultura en la Sociedad de la Información* **13**(2), 324–342.
- Heinz, F.; Da Rosa, F. (2017). *Guía práctica sobre software libre: su selección y aplicación local en Latinoamérica y el Caribe*. Montevideo: UNESCO, 2007.. Montevideo, Uruguay.
- Landaburo, G. (2016). La asignatura Ingeniería de Control I sobre SCILAB.. Trabajo de diploma. Universidad Central "Marta Abreu" de las Villas. Santa Clara, Cuba.
- Lanza, J. L. (2015). Evaluación de uso de software libre en la disciplina de Sistemas de Control.. Trabajo de diploma. Universidad Central "Marta Abreu" de las Villas. Santa Clara, Cuba.
- López, J. R. (2005). Ciencia y comunicación científica: edición digital y otros fundamentos del libre acceso al conocimiento. *El profesional de la información* **14**(4), 246–254.
- Martínez, B. L. (2015-2016). Programa analítico.
- Nelson, D. R.; Barber, D. B.; McLain T. W.; Beard R. W. (2007). Vector field path following for miniature air vehicles. *IEEE Transactions on Robotics* **23**(3), 519–529.
- Ortega, P. (2004). *La enseñanza del Álgebra Lineal mediante sistemas informáticos de cálculo algebraico*. Universidad Complutense de Madrid, Servicio de Publicaciones. Madrid, España.

- Paumier, R.; Pérez, Y.; Meneses A. (2015). Guía cubana de migración a software libre. Tesis de maestría. Universidad de las Ciencias Informáticas. La Habana, Cuba.
- Peña, J. C. (2013). Uso didáctico de recursos tic y software libre en la enseñanza de la observación lunar. Tesis doctoral. Universidad Nacional de Colombia. Bogotá, Colombia.
- Pizarro, R. A.; Ascheri, M. E. (2009). Diseño e implementación de un software educativo en cálculo numérico. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*.
- Rosales, G. R. (2012). Design and implementation of workshops for the teaching and learning of matrix algebra and solution of systems of linear equations with Scilab. Tesis doctoral. Universidad Nacional de Colombia-Sede Manizales. Manizales, Colombia.
- Santos, D. P. (2013). Avaliao da expansao de células estromais mesenquimais em bioreator de fibra oca. Trabajo de diploma. Universidad Federal de San Carlos. Sao Carlos, Brasil.
- Sharma, N.; Gobbert, M. K. (2010). A comparative evaluation of matlab, octave, freemat, and scilab for research and teaching.
- Squires, D.; McDougall, A. (1997). *Cómo elegir y utilizar software educativo: guía para el profesorado*. Ediciones Morata. Madrid, España.
- Stallman, R. (2004). *Software libre para una sociedad libre*. Traficantes de Sueños. Madrid, España.
- Vargas, M.; Ferrer, O. L.; Batista I. (2011). Una metodología para la elaboración de tareas docentes integradoras en las asignaturas técnicas. *EduSol*.

REFERENCIAS BIBLIOGRÁFICAS
ANEXOS

ANEXOS

ANEXO A

CÓDIGOS PARA LAS ACTIVIDADES PRÁCTICAS DE LA ASIGNATURA INGENIERÍA DE CONTROL II

Actividad Práctica 2

A.0.1. Ejercicio 1

Código para obtener sistema discreto por método de aproximación *ZOH*

//Obtener la función de transferencia

```
sisc=syslin('c',poly([10 1], 's', 'c'),poly([6 11 6 1], 's', 'c'));
```

//La función *dscr* discretiza el sistema por el método Zero Order Hold

```
sisd=ss2tf(dscr(tf2ss(sisc),0.1));
```

```
sisd2=ss2tf(dscr(tf2ss(sisc),0.5));
```

```
sisd3=ss2tf(dscr(tf2ss(sisc),1));
```

```
sisd4=ss2tf(dscr(tf2ss(sisc),2));
```

Código para obtener sistema discreto por método de aproximación *Tustin*

```
sisc=syslin('c',poly([10 1], 's', 'c'),poly([6 11 6 1], 's', 'c'));
```

```
sisd=ss2tf(cls2dls(tf2ss(sisc),0.1));
```

```
sisd2=ss2tf(cls2dls(tf2ss(sisc),0.5));
```

```
sisd3=ss2tf(cls2dls(tf2ss(sisc),1));
```

```
sisd4=ss2tf(cls2dls(tf2ss(sisc),2));
```

```
sisdzpk=zpk(sisd); //Representación del sistema en cero, ganancia y polos
```

```
plzr(sisd);
```

```
zgrid
```

A.0.2. Ejercicio 2

Código para obtener la respuesta del sistema en *Z* a entrada paso e impulso.

```

sisc=syslin('c',poly([10 1], 's', 'c'),poly([6 11 6 1], 's', 'c'));
sisd=ss2tf(cls2dls(tf2ss(sisc),0.1));
sl=tf2ss(sisd);

u=zeros(1,71);u(1)=1;
x1=dsimul(sl,u) //Respuesta Impulso

u=ones(1,71);
x2=dsimul(sl,u); //Respuesta a paso

xset('window',1)
t=[0:1:70]
plot2d2(t,x2)

xset('window',2)
t1=[0:1:70]
plot2d2(t1,x1)

```

A.1. Actividad Práctica 3

Obtención del controlador digital

```

n=1550
d=poly([1 0.024], 's', 'c')
G=syslin('c',n,d);
Gz=ss2tf(dscr(tf2ss(G),0.1));

```

A.1.1. Deadbeat

```

num=1
den=poly([0 1], 'z', 'c')
Tz=syslin('d',num,den) //Define la función en tiempo discreto
Dz=Tz/(Gz*(1-Tz))

```

A.1.2. Kalman

```

// Gd=Pz/Qz con Pz normalizado //  $Gd = \frac{1526z^{-1}}{1 - 0,0155z^{-1}}$ 
//Dividimos entre 1526
Pz=1526/1526
Qz=([1, -0.0155])/1526

```

```
Tz=syslin('d',Pz,Qz)
Gz=ss2tf(dscr(tf2ss(Tz),0.1));
```

A.1.3. Dahlin

```
cita=0.5
m=1
//miu t=0.1
//cita=k*t
k=cita/t

DzD = [z-(k+1) * [1-exp(-t/m)]]/Gd * [1-exp(-t/m) * z - z-(k+1) * 1-exp(-t/m)]
```

A.2. Actividad Práctica 4

```
A=[0 1 0;0 0 1;0 -2 -4];
B=[0;0;1];
C=[1 0 0];
```

```
M = cont_mat(A, B) //Permite obtener la matriz de controlabilidad
rank(M)
```

```
N = obsv_mat(A, C) //Permite obtener la matriz de observabilidad
rank(N) //Para obtener el rango de la matriz
```

```
P=[-0.5+i*0.8;-0.5-i*0.8;-5]
//Buscar funcion acker
K2=ppol(A,B,P)
PO=[-4;-4;-4]
```

```
//Para obtener la ganancia del observador
Ke=ppol(A',C',PO)
Ke=Ke'
```

A.3. Actividad Práctica 5

```
A11=[-1 1; 0 2];
B11=[1;1];
C11=[1 0];
D11=0;
R=1;
```

```
Q=[1 0;0 1];  
Big=sysdiag(Q,R); // Para hacer R una matriz 2 con 2  
[w,wp]=fullrf(Big);  
C12=wp(:,1:2);  
D12=wp(:,3:s);  
ss=syslin('c',A11,B11,C12,D12);  
K1=lqr(ss);
```

ANEXO B

CÓDIGOS PARA LAS ACTIVIDADES PRÁCTICAS DE LA ASIGNATURA MODELADO Y SIMULACIÓN.

Laboratorio 1

B.0.1. Ejercicio 1

```
g1=syslin('c',poly([2],'s','c'),poly([2 20],'s','c')); //permite crear funciones de transfe-  
rencia
```

```
g2=syslin('c',poly([5],'s','c'),poly(conv([2 0 1],[5 3]),'s','c'));
```

```
g3=syslin('c',poly([1 4],'s','c'),poly([20 0 10 1],'s','c'));
```

```
g4=syslin('c',poly([5],'s','c'),poly(conv([0 3 1],[2 1]),'s','c'));
```

```
gzpk=zpk(g1) //Para obtener el sistema en forma de ceros, polos y ganancia
```

```
groots=roots(g1)//Para obtener las raíces de la función de transferencia
```

B.0.2. Ejercicio 2

```
numg1=g1.num//Se obtiene el numerador de g1
```

```
deng1=g1.den//Se obtiene el denominador de g1
```

```
numg2=g2.num
```

```
deng2=g2.den
```

```
numg3=g3.num
```

```
deng3=g3.den
```

```
numg4=g4.num
```

```
deng4=g4.den
```

B.0.3. Ejercicio 2.1

```
a=[0 1;-4 -2]
```

```
b=[0;2]
```

```
c=[1 0]
d=0
ss1=syslin('c',a,b,c,d)
```

Laboratorio 3

B.0.4. Ejercicio 1

```
//Función de transferencia
sisc=syslin('c',poly([2],'s','c'),poly([3 4],'s','c'));

//Se crea un vector t para plotear la entrada, ya sea paso, rampa o impulso, con respecto
al tiempo
t=0:0.1:8;

xset('window',1);
step_res=csim('step',t,sisc);//El comando csim permite dar la entrada deseada
plot2d(t,step_res);
xtitle('Respuesta Paso','tiempo (s)','Respuesta');

xset('window',2);
imp_res=csim('imp',t,sisc);
plot2d(t,imp_res);
xtitle('Respuesta Impulso','tiempo (s)','Respuesta');

xset('window',3);
ramp_res=csim(t,t,sisc);
plot2d(t,ramp_res);
xtitle('Respuesta Rampa','tiempo (s)','Respuesta');
```

B.0.5. Ejercicio 2

```
for x=[5.6:-0.8:0.8]
sisc=syslin('c',poly([16],'s','c'),poly([16 x 1],'s','c'));
t=0:0.01:5;

xset('window',1);
step_res=csim('step',t,sisc);
plot2d(t,step_res);
xtitle('Respuesta Paso','Tiempo (s)','Amplitud');

xset('window',2);
imp_res=csim('imp',t,sisc);
```

```

plot2d(t,imp_res);
xlabel('Respuesta Impulso','Tiempo (s)','Amplitud');
end

```

B.0.6. Ejercicio 3

```

k1=0.1 //Sobre
k2=0.25 //Crítico
k3=10 //Sub
h=1

for k=[0.1 0.25 10]
g=syslin('c',poly([k],'s','c'),poly([0 1 1],'s','c'));
lc=g/. h//En lc se obtiene el lazo cerrado de g,h

t=0:0.01:50;
xset('window',1);
step_res=csim('step',t,lc);
plot2d(t,step_res);
xlabel('Respuesta Paso','Tiempo (s)','Amplitud');
end

```

Clase Práctica 11

B.0.7. Ejercicio 1

```

g=syslin('c',poly([10],'s','c'),poly(conv([10 11 1],[0.1 1]),'s','c'));
t=0:0.01:5;
xset('window',1);
step_res=csim('step',t,g);
plot2d(t,step_res);
xlabel('Step response','Tiempo (s)','Amplitud');

```