



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE INGENIERÍA ELECTRÓNICA
PROYECTO TÉCNICO

Previa la obtención del Título de:

INGENIERO ELECTRÓNICO

TEMA

**“Diseño de un algoritmo de control para el péndulo invertido simple marca
Feedback modelo 33-936s usando variables de estado”**

AUTOR

Carlos Luis Cifuentes Briones

TUTOR: Ing. Vicente Peñaranda.

GUAYAQUIL

2018

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA

Yo, CARLOS LUIS CIFUENTES BRIONES, autorizo a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, declaro que los conceptos, análisis desarrollados y las conclusiones del presente trabajo son exclusiva responsabilidad del autor.

Guayaquil, 23 de noviembre del 2018

Carlos Cifuentes B.

C.I. 092335753-7

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR

Yo, **CARLOS LUIS CIFUENTES BRIONES**, con documento de identificación N° **0923357537**, manifiesto mi voluntad y cedo a la **UNIVERSIDAD POLITÉCNICA SALESIANA** la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de grado titulado **“DISEÑO DE UN ALGORITMO DE CONTROL PARA EL PÉNDULO INVERTIDO SIMPLE MARCA FEEDBACK MODELO 33-936s USANDO VARIABLES DE ESTADO”** mismo que ha sido desarrollado para optar por el título de **INGENIERO ELECTRÓNICO CON MENCIÓN EN TELECOMUNICACIONES**, en la Universidad Politécnica Salesiana, quedando la universidad facultada para ejercer plenamente los derechos antes cedidos.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscrito este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 17 de diciembre del 2018

(f) Carlos Cifuentes Briones

C.I: 092335753-7

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

En calidad de tutor del trabajo de titulación proyecto técnico **“DISEÑO DE UN ALGORITMO DE CONTROL PARA EL PÉNDULO INVERTIDO SIMPLE MARCA FEEDBACK MODELO 33-936s USANDO VARIABLES DE ESTADO”** realizado por el señor Carlos Luis Cifuentes Briones, para optar por el título de ingeniero electrónico, certifico que el mencionado proyecto fue realizado bajo mi dirección.

(f) Ing. Vicente Peñaranda

DEDICATORIA

Dedico esta tesis a mi familia que siempre me apoya incondicionalmente. A mi Tío y a mi abuelo dos pilares fundamentales en mi vida que siempre impulsaron en mí valores y responsabilidades que nunca las voy a olvidar.

Carlos Cifuentes B.

AGRADECIMIENTO

Agradezco a Dios por brindarme salud y bendiciones, a mis padres por su apoyo moral y económico y le agradezco a mi Universidad, a mi tutor Ing. Vicente Peñaranda y a mis maestros por sus esfuerzos para que pueda lograr mi meta y graduarme como un profesional en Ingeniería Electrónica con mención en Telecomunicaciones.

Carlos Cifuentes B.

RESUMEN

AÑO	ALUMNO	TUTOR DEL PROYECTO	TEMA DEL PROYECTO
2018	Carlos Luis Cifuentes Briones	Ing. Vicente Peñaranda	“DISEÑO DE UN ALGORITMO DE CONTROL PARA EL PÉNDULO INVERTIDO SIMPLE MARCA FEEDBACK MODELO 33-936s USANDO VARIABLES DE ESTADO”

El péndulo invertido es un sistema no-lineal e inestable que está compuesto por un carro que se desplaza libremente en dirección horizontal en el cual se encuentra un pivote con libertad de oscilar. En este proyecto se presenta el diseño de tres alternativas de control: Algoritmo de control por Variables de estado, Algoritmo de control PID y Algoritmo MPC (Control Predictivo basado en modelo). Antes de diseñar los algoritmos de control se debe hacer un modelo matemático del sistema servo-mecánico del péndulo invertido. Para después linealizar el sistema en su punto de equilibrio. Una vez linealizado el sistema los controladores son diseñados en el software de Matlab/ Simulink. El análisis, modelado y simulación del sistema nos conduce a concretar los objetivos planteados. El desempeño de estos controladores va a hacer comparado por análisis en su estado estacionario.

Palabras claves: Péndulo Invertido, Variables de estado, PID, MPC (Control Predictivo Basado en modelo)

ABSTRACT

YEAR	STUDENT	PROJECT TUTOR	PROJECT TITLE
2018	Carlos Luis Cifuentes Briones	Ing. Vicente Peñaranda	“DISEÑO DE UN ALGORITMO DE CONTROL PARA EL PÉNDULO INVERTIDO SIMPLE MARCA FEEDBACK MODELO 33-936s USANDO VARIABLES DE ESTADO”

The inverted pendulum is a non-linear and unstable system which is composed of a freely moving car in the horizontal axis with a pivot with the freedom to oscillate 360°. This project designs three control alternatives: The Control algorithm for State variables, PID control algorithm and MPC algorithm (Predictive control based on model). Before the stage of designing the control algorithms, the mathematical model of the servo-mechanical system of the inverted pendulum must be made. This is the main factor to the linearization of the system at its equilibrium point. Once the system is linearized the model is presented in the Matlab software at the platform Simulink. The analysis, model and simulation of the system allow us to conduct the objectives of this project. The analysis of the performance of these controllers is compared between them at its steady state.

Palabras claves: Inverted Pendulum, State Variables, PID, MPC (Predictive control based on model).

ÍNDICE DE CONTENIDOS

TEMA.....	i
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA	ii
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR	iii
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
RESUMEN.....	vii
ABSTRACT	viii
ÍNDICE DE CONTENIDOS	ix
ÍNDICE DE FIGURAS.....	xvi
1. EL PROBLEMA.....	2
1.1 Antecedentes	2
1.2 Importancia y Alcance	2
1.3 Delimitación.....	3
1.3.1 Delimitación Temporal	3
El tiempo de la elaboración del proyecto se realizó en el año 2017-2018	3
1.3.2 Delimitación Espacial	3

El proyecto técnico de titulación está dirigido y enfocado hacia la materia Teoría de Control de la Universidad Politécnica Salesiana, Sede de Guayaquil.	3
1.3.3 Delimitación Académica.....	3
1.4 Objetivos	3
1.4.1. Objetivo General	3
1.4.2. Objetivos Específico	4
1.5 Metodología	4
1.6 Beneficiarios	5
2. ESTADO DEL ARTE.....	6
2.1. Péndulo invertido.	6
2.2. Sistemas de Control.	6
2.3. Representación en variables de estado.....	8
2.3.1. Definiciones de espacios de estados o variables de estado	8
2.4. Representación en Control Predictivo.....	12
3.1. Modelo Matemático del Péndulo Invertido Simple	13
3.1.1. Análisis Descriptivo de la Planta del Péndulo Digital modelo 33-936s marca Feedback.....	13
3.1.2. Modelamiento Matemático del Péndulo Digital modelo 33-936s marca Feedback	15
3.1.3. Linealización del Modelo.....	18

3.1.4.	Representación mediante Variables de estado	18
3.2.	Análisis del modelo del Péndulo Invertido en el espacio de estados	21
3.2.1.	Estabilidad del Péndulo	22
3.2.2.	Controlabilidad del Péndulo	24
3.2.3.	Observabilidad del Péndulo	26
3.3.	Respuesta del sistema en lazo abierto	27
3.4.	Respuesta del sistema en lazo cerrado	31
3.5.	Diseño del controlador para la estabilización del Péndulo	32
3.5.1.	Controlador por Variables de estado	32
3.5.1.1	Polos deseados	34
3.5.1.2	Diseño de controladores en variables de estado	34
3.5.2.	Controlador PID	37
3.5.3.	Diseño del controlador Predictivo MPC	39
4.	RESULTADOS	44
4.1	Análisis de resultados de la simulación de controladores en Matlab	44
4.2	Análisis comparativo de los algoritmos de control	44
4.2.1	Análisis comparativo de los algoritmos de control usando Variables de estado, PID y Predictivo	44
4.2.2	Análisis comparativo de los algoritmos de control usando Variables de Estado, PID y Predictivo	44

5. CONCLUSIONES	50
RECOMENDACIONES	51
REFERENCIAS	52
PRÁCTICAS DE LABORATORIO.....	65
“Linealizar Modelado Matemático del sistema dinámico del Péndulo Invertido en variables de estado”	65
1.1.1. Objetivos.	65
1.1.2. Recursos utilizados.....	65
1.1.3. Tiempo estimado.	65
1.1.4. Desarrollo.	65
1.1.5. Conclusiones de la práctica.	73
“Simulación y Análisis del sistema en lazo abierto en el espacio de estados”	74
1.3.1. Objetivos.	74
1.3.2. Recursos utilizados.....	74
1.3.3. Tiempo estimado.	74
1.3.4. Desarrollo.	74
1.3.5. Conclusiones de la práctica.	76
“Simulación y Análisis del sistema en lazo cerrado en el espacio de estados”	77
1.4.1. Objetivos.	77
1.4.2. Recursos utilizados.....	77

1.4.3.	Tiempo estimado.....	77
1.4.4.	Desarrollo.....	77
1.4.5.	Conclusiones de la práctica.....	79
“Diseño de un algoritmo de control de variables de estado realimentado”.....		80
1.5.1.	Objetivos.....	80
1.5.2.	Recursos utilizados.....	80
1.5.3.	Tiempo estimado.....	80
1.5.4.	Desarrollo.....	80
1.5.5.	Conclusiones de la práctica.....	83
“Diseño de un algoritmo de control variables de estado mediante Ubicación de Polos Deseados”.....		84
1.6.1.	Objetivos.....	84
1.6.2.	Recursos utilizados.....	84
1.6.3.	Tiempo estimado.....	84
1.6.4.	Desarrollo.....	84
1.6.5.	Conclusiones de la práctica.....	87
“Análisis de la respuesta transitoria y estacionaria del sistema del Péndulo Invertido”.....		88
1.7.1.	Objetivos.....	88
1.7.2.	Recursos utilizados.....	88

1.7.3.	Tiempo estimado.....	88
1.7.4.	Desarrollo.....	88
1.7.5.	Conclusiones de la práctica.....	90
“Diseño de observador de estados para el Péndulo Invertido”		91
1.8.1.	Objetivos.	91
1.8.2.	Recursos utilizados.....	91
1.8.3.	Tiempo estimado.....	91
1.8.4.	Desarrollo.....	91
1.8.5.	Conclusiones de la practica.....	93
1.9.1.	Objetivos.	94
1.9.2.	Recursos utilizados.....	94
1.9.3.	Tiempo estimado.....	94
1.9.4.	Desarrollo.....	94
1.9.5.	Conclusiones de la practica.....	96
1.10.1.	Objetivos.	97
1.10.2.	Recursos utilizados.....	97
1.10.3.	Tiempo estimado.....	97
1.10.4.	Desarrollo.....	97
1.10.5.	Conclusiones de la practica.....	102

“Análisis comparativo los algoritmos de control PID, Variables de estado, MPC”

.....	103
1.11.1. Objetivos.	103
1.11.2. Recursos utilizados.....	103
1.11.3. Tiempo estimado.....	103
1.11.4. Desarrollo.....	103
1.11.5. Conclusiones de la Práctica.....	106

ÍNDICE DE FIGURAS

FIGURA 1: DIAGRAMA DE CUERPO LIBRE DEL PÉNDULO INVERTIDO. [2]	6
FIGURA 2: SISTEMA CON MÚLTIPLES ENTRADAS Y SALIDAS. [7]	7
FIGURA 3 : SISTEMA DE CONTROL. [9]	7
FIGURA 4: ESTRUCTURA MECÁNICA DEL PÉNDULO INVERTIDO. [6]	14
FIGURA 5: DIAGRAMA MECÁNICO DEL PÉNDULO INVERTIDO. [6]	15
FIGURA 6 : DIAGRAMA CUERPO LIBRE DEL PÉNDULO INVERTIDO. [7]	16
FIGURA 7: VARIABLES DE ESTADO DEL PÉNDULO INVERTIDO	22
FIGURA 8: PLANO ESTABILIDAD EN EL DOMINIO DE LAPLACE [9]	23
FIGURA 9: POLOS DEL SISTEMA PÉNDULO INVERTIDO	24
FIGURA 10: COMPROBACIÓN DE CONTROLABILIDAD DEL SISTEMA (PÉNDULO INVERTIDO) EN MATLAB	25
FIGURA 11: COMPROBACIÓN DE OBSERVABILIDAD DEL SISTEMA (PÉNDULO INVERTIDO) EN MATLAB	26
FIGURA 12: RESPUESTA DEL SISTEMA A ENTRADA TIPO IMPULSO	27
FIGURA 13 RESPUESTA DEL SISTEMA A ENTRADA TIPO PASO	28
FIGURA 14: MODELO DEL PÉNDULO INVERTIDO. [6]	29
FIGURA 15: SISTEMA LAZO ABIERTO DEL PÉNDULO INVERTIDO EN SIMULINK	29
FIGURA 16 GRÁFICA DE POSICIÓN DEL CARRO EN LAZO ABIERTO DEL SISTEMA PÉNDULO INVERTIDO	30
FIGURA 17: GRÁFICA DE ÁNGULO DEL PÉNDULO EN LAZO ABIERTO DEL SISTEMA PÉNDULO INVERTIDO	30
FIGURA 18: SISTEMA REALIMENTADO EN VARIABLES ESTADO	31
FIGURA 19: RESPUESTA DEL SISTEMA A ENTRADA TIPO PASO	31
FIGURA 20: UBICACIÓN DE POLOS EN EL DOMINIO DE LAPLACE	33
FIGURA 22: DISEÑO DEL ALGORITMO DE CONTROL EN VARIABLES DE ESTADO EN SIMULINK	35
FIGURA 23: RESPUESTA DEL ALGORITMO DE CONTROL EN VARIABLES DE ESTADO SIMULADO EN SIMULINK	36
FIGURA 24: CONTROLADOR PID CONSIDERADO PARA LA PLANTA PÉNDULO INVERTIDO. [6]	37
FIGURA 25: SALIDAS CONTROLADOR PID PARA LA PLANTA DEL PÉNDULO INVERTIDO	38
FIGURA 26: IDEA DE CONTROL PREDICTIVO. [11]	39

FIGURA 27: MPC DESIGNER TOOLBOX	40
FIGURA 28: VENTANA MPC STRUCTURE	40
FIGURA 29: DISEÑO DEL CONTROLADOR MPC.....	41
FIGURA 30: RESPUESTA DEL CONTROLADOR MPC, SEÑAL DE ENTRADA U	42
FIGURA 31: RESPUESTA DEL CONTROLADOR MPC, SEÑALES DE SALIDA.....	42
FIGURA 32: CONTROLADOR PREDICTIVO DEL SISTEMA DEL PÉNDULO INVERTIDO.....	43
FIGURA 33: SIMULACIÓN DE MATLAB CORRESPONDIENTE A LA POSICIÓN DEL CARRO ANTE EL CONTROLADOR DE VARIABLES DE ESTADO.....	45
FIGURA 34: SIMULACIÓN DE MATLAB CORRESPONDIENTE A LA POSICIÓN DEL CARRO ANTE EL CONTROLADOR PID	45
FIGURA 35: SIMULACIÓN DE MATLAB CORRESPONDIENTE A LA POSICIÓN DEL CARRO ANTE EL CONTROLADOR MPC.....	46
FIGURA 36: COMPARACIÓN DE ESTABILIZACIÓN DE POSICIÓN DEL CARRO ENTRE LOS CONTROLADORES DISEÑADOS.	46
FIGURA 37: SIMULACIÓN DE MATLAB CORRESPONDIENTE AL ÁNGULO DEL PÉNDULO ANTE EL CONTROLADOR DE VARIABLES DE ESTADO.....	47
FIGURA 38: SIMULACIÓN DE MATLAB CORRESPONDIENTE AL ÁNGULO DEL PÉNDULO ANTE EL CONTROLADOR PID.....	47
FIGURA 39: SIMULACIÓN DE MATLAB CORRESPONDIENTE AL ÁNGULO DEL PÉNDULO ANTE EL CONTROLADOR MPC.	48
FIGURA 40: COMPARACIÓN DE ESTABILIZACIÓN DEL ANGULO DEL PENDULO ENTRE LOS CONTROLADORES DISEÑADOS.	48

ÍNDICE DE TABLAS

TABLA 1 PARÁMETROS DEL PÉNDULO. [6]	15
TABLA 2: VALORES PARÁMETROS DEL PÉNDULO [6]	17
TABLA 3 GANANCIAS PID [6]	38
TABLA 4 ÍNDICES DE DESEMPEÑO DE LOS CONTROLADORES PID, VARIABLES DE ESTADO Y MPC CORRESPONDIENTE A LA POSICIÓN DEL CARRO.	49
TABLA 5: ÍNDICES DE DESEMPEÑO DE LOS CONTROLADORES PID, VARIABLES DE ESTADO Y MPC CORRESPONDIENTE AL ÁNGULO DEL PÉNDULO.....	49

ÍNDICE DE ANEXOS

ANEXO 1 CÓDIGO DE LA SIMULACIÓN	55
ANEXO 2 PRÁCTICA#1:.....	65
ANEXO 3 PRÁCTICA#2:.....	74
ANEXO 4: PRÁCTICA#3:.....	77
ANEXO 5: PRÁCTICA#4:.....	80
ANEXO 6 PRÁCTICA#5:.....	84
ANEXO 7 PRÁCTICA#6:	88
ANEXO 8: PRÁCTICA#7:	91
ANEXO 9 PRACTICA #8:.....	94
ANEXO 10 PRACTICA #9:	97
ANEXO 11:PRÁCTICA #10:	103

INTRODUCCIÓN

En este proyecto de fin de carrera se detalla el diseño de tres algoritmos de control para el Péndulo invertido utilizando la herramienta de Simulink del software MATLAB.

La metodología a seguir será en primer lugar el estudio y análisis de la unidad Digital del Péndulo Invertido 33-936s marca Feedback, para representarlo con un modelo matemático. Segundo lugar linealizar el sistema en sus puntos de equilibrio. Tercer lugar Diseño de los algoritmos de control (Variables de estado, PID, MPC). Cuarto lugar se muestran sus resultados por simulación en Matlab.

En el segundo capítulo se encuentra la parte teórica de este proyecto en donde se detalla los principios utilizados para desarrollar el proyecto con el objetivo de entender las bases teóricas para poder diseñar los algoritmos de control en el software de Matlab descrito en el tercer capítulo.

Una vez diseñado los algoritmos de control se realiza un análisis comparativo entre los controladores MPC (control Predictivo basado en modelo), PID y Variables de estado con el fin de analizar el desempeño de cada algoritmo lo que permitirá establecer que controlador tiene mejores prestaciones para este sistema.

Considérese entonces a este trabajo como una útil guía académica, para los interesados en el campo de la ingeniería de control.

1. EL PROBLEMA

1.1 Antecedentes

Actualmente la Universidad Politécnica Salesiana cuenta con equipos de laboratorios en los cuales los estudiantes puedan realizar prácticas y enlazar sus conocimientos teóricos con los prácticos, el caso del laboratorio de control automático el mismo que consta entre otros equipos con un péndulo invertido simple marca Feedback el cual opera con un solo algoritmo de control por lo cual limita a no experimentar e obstaculiza su aprendizaje. Dada esta problemática se lleva a la necesidad de plantear alternativas que contribuyan a mejorar los procesos de la enseñanza-aprendizaje y se plantea a diseñar prácticas de entrenamiento con nuevos algoritmos de control que permitiría ampliar los conocimientos, mejorar las destrezas y dar apertura a investigaciones futuras en las áreas de instrumentación, automatización y control. El presente proyecto va a hacer solventado en el software de MATLAB con la herramienta de Simulink donde se verifica el modelo de la planta del péndulo y se lleva a cabo tareas de diseño análisis y resultados.

1.2 Importancia y Alcance

Este proyecto tiene como fin elaborar algoritmos para prácticas de futuros investigadores de control de la Universidad Politécnica Salesiana que permita dar inicio a futuras investigaciones de sistemas de control. Este estudio sirve como una primera aproximación a los problemas que plantean una mayor complejidad en la vida real, tanto así que se considera el péndulo invertido como una perfecta analogía para el diseño de un controlador de procesos reales como lanzamientos de misiles, estabilidad de grúas entre otros.

La estabilización de un péndulo invertido simple es uno de los sistemas más utilizados en el uso de diseño de controladores porque se aplican algunos de los conceptos adquiridos en las diferentes asignaturas por lo que este proyecto brinda una orientación y aprendizaje a futuros investigadores de control.

El presente trabajo aborda 3 algoritmos de control (controlador en Variables de Estado, controlador predictivo basado en modelo y PID) y se realiza un análisis comparativo entre estos donde se evalúa las eficiencias de cada uno.

1.3 Delimitación

1.3.1 Delimitación Temporal

El tiempo de la elaboración del proyecto se realizó en el año 2017-2018

1.3.2 Delimitación Espacial

El proyecto técnico de titulación está dirigido y enfocado hacia la materia Teoría de Control de la Universidad Politécnica Salesiana, Sede de Guayaquil.

1.3.3 Delimitación Académica

El Péndulo Invertido es uno de los problemas de control más utilizados en el ámbito educativo ya que representa un sistema no-lineal mecánico sub-actuado, con dos grados de libertad. La experimentación de distintas estrategias de control para el balanceo del Péndulo Invertido través del movimiento del carro con objetivo de llevar al péndulo a su posición erecta hace perfecta analogía para aplicaciones de procesos reales como lanzamiento de cohetes espaciales, pruebas de resistencias de edificios y lanzamiento de cohetes. Este proyecto estará disponible para estudiantes de la Universidad Politécnica Salesiana en las carreras de Ingeniería.

1.4 Objetivos

1.4.1. Objetivo General

Diseñar un algoritmo de control para el péndulo invertido simple de la unidad Péndulo Digital modelo 33-936s marca FEEDBACK usando un controlador de variables de estado para el laboratorio de control automático de Universidad Politécnica Salesiana sede Guayaquil

1.4.2. Objetivos Específico

1. Hacer un análisis descriptivo de los componentes que conforman el sistema interno del péndulo simple que se encuentra en el laboratorio de control automático
2. Estudiar la operación del sistema del PID del péndulo invertido.
3. Diseñar un algoritmo de control usando variables de estado para la operación del péndulo simple en el software Matlab con el Toolbox de Simulink
4. Agrupar y etiquetar los diferentes bloques de programa para facilitar su entendimiento y dar apertura para futuras investigaciones
5. Diseñar 10 practicas con el péndulo invertido simple con algoritmos de variable de estado
6. Realizar un análisis comparativo entre los controladores PID y variables de estado aplicadas en el módulo
7. Diseñar un algoritmo de control Predictivo para la operación del Péndulo Simple en el software Matlab usando la herramienta MPC Toolbox simulada en Simulink

1.5 Metodología

La metodología utilizada en el proceso de desarrollo del proyecto de titulación, está constituida por el método Observación, el método deductivo.

Método Observación

El levantamiento de información previo al desarrollo del proyecto. Se realiza un análisis descriptivo de las diferentes funciones que realiza cada componente del módulo 33-936s para conocer el funcionamiento de la planta y así poder hacer un modelado adecuado del sistema.

Método Deductivo

Se inició el proyecto con los conocimientos adquiridos durante la carrera universitaria, lo cual permitió diseñar, simular y analizar el sistema del péndulo invertido.

En la elaboración del diseño de un algoritmo por variables de estado se aplican conocimientos adquiridos en Teoría de Control I y es necesario tener conocimientos del software MATLAB con su debido Toolbox de Simulink. Para el desarrollo del algoritmo se debe tener en claro su modelo matemático en el espacio de estados (variables de estado) el péndulo invertido.

1.6 Beneficiarios

Los principales beneficiarios de este proyecto de simulación que se ha implementado en la Universidad Politécnica Salesiana de la Ciudad de Guayaquil son principalmente todos los alumnos que cursan la materia de Teoría de Control.

2. ESTADO DEL ARTE

2.1. Péndulo invertido.

El péndulo invertido es un servomecanismo en cual consiste en un pivote o barra con una masa m despreciable con libertad de oscilar sobre su eje adaptado en un carro de masa M . El carro que puede ser desplazado a lo largo de un eje horizontal con distancia finita cuyo movimiento genera una oscilación en el péndulo. [1]

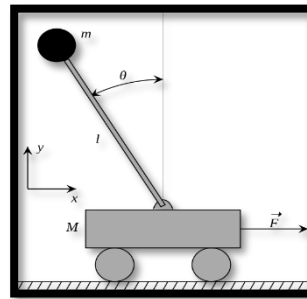


FIGURA 1: DIAGRAMA DE CUERPO LIBRE DEL PÉNDULO INVERTIDO. [2]

El balanceo del péndulo invertido es un clásico problema en el área de teoría de control que explica un punto de referencia a los ingenieros a probar y verificar el funcionamiento de sus algoritmos de control [3]. El péndulo invertido encuentra aplicaciones en varios procesos en la vida real como el levantamiento de grúas, lanzamiento de cohetes, estabilización de satélites y muchos mas. En lo que lo convierte en un problema interesante para su análisis. [4][5].

2.2. Sistemas de Control.

Para entender un sistema de control se debe tener claro una serie de conceptos para comprender el comportamiento interno reflejado en su modelado de la relación entre entrada y salida de los sistemas lineales o no lineales.

Sistema: se define como una combinación de componentes que actúan conjuntamente y que cumplen determinado objetivo. [7].

Sistema de control: es una interconexión de componentes que forman una configuración del sistema que proporcionará una respuesta deseada. [8].

Sistema lineal: Es el que cumple el Teorema de superposición.

Sistema SISO: Sistema de control que tiene una entrada una salida.

Sistema SIMO: Sistema de control que tiene una entrada y múltiples salidas.

Sistema MISO: Sistema de control que tiene múltiple entrada y una salida.

Sistema MIMO: Sistema de control que tiene múltiple entrada y múltiples salidas.

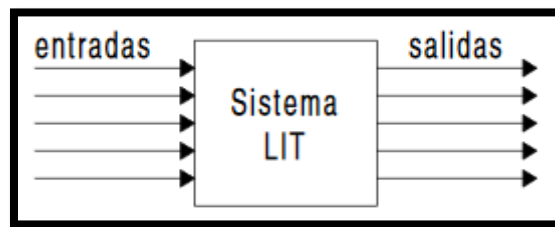


FIGURA 2: SISTEMA CON MÚLTIPLES ENTRADAS Y SALIDAS. [7]

Planta: conjunto de elementos de un equipo o dispositivo que cumplen una función de efectuar una operación.

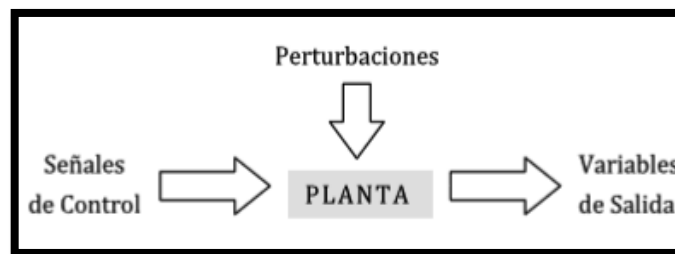


FIGURA 3 : SISTEMA DE CONTROL. [9]

Variable controlada: Variable que se puede medir y controlar. [7].

Perturbación: Ruido externo o interno q afecta al sistema

Sistema de control en lazo abierto: Son los sistemas que no tienen retroalimentación que su salida no tiene efecto con la señal de entrada

Sistemas de control en lazo cerrado: Son los sistemas de control realimentados que compara entrada con su salida con el fin de reducir el error del sistema y obtener una señal deseada.

2.3. Representación en variables de estado.

La teoría de control moderna está basada en el conocimiento del comportamiento interno de los sistemas, reflejado en las variables que describen la dinámica del sistema. Estas variables constituyen el concepto de estado actual del sistema. El conocimiento del estado actual del sistema junto a su señal de entrada en $t=t_0$, determina el comportamiento del sistema para un tiempo $t \geq t_0$.

2.3.1. Definiciones de espacios de estados o variables de estado

La teoría de control moderna se basa en la representación matemática de los sistemas dinámicos por medio del concepto de estado.

Estado: Es la caracterización interna del sistema en cada instante de tiempo. En él se resume la historia del sistema. Las salidas de los sistemas dependen de los estados y de las entradas.

Variables de estado: Son las variables que expresa el estado dinámico de un sistema.

Vector de estado: Es la agrupación de variables de estado en forma de vector que representa un punto en el espacio.

Ecuaciones de estado: son las ecuaciones matriciales que representan al comportamiento dinámico y las salidas del sistema.

Espacio de estado: Todo vector de estado estará representado por un punto en el espacio de estados.

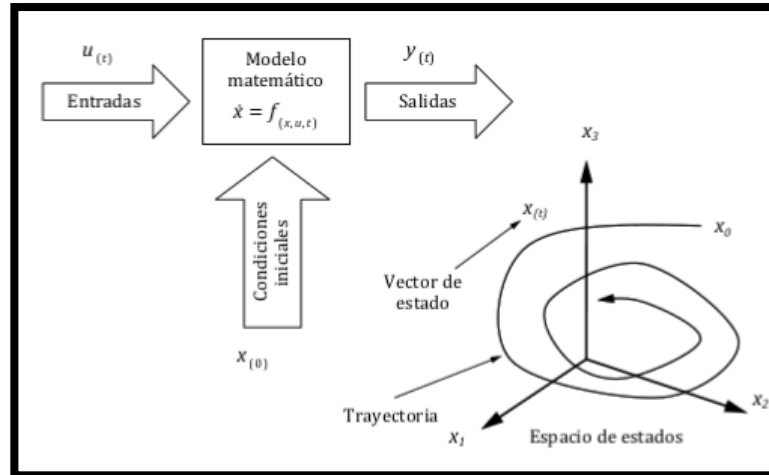


FIGURA 4: COMPORTAMIENTO DE UN SISTEMA EN EL ESPACIO DE ESTADOS. [9]

El modelado de los sistemas dinámicos normalmente se usa el análisis en el espacio de estados ya que trata directamente sobre las ecuaciones que describen el sistema ya que centra la atención en los tres tipos de variables; las variables de entrada, las variables de salida y las variables de estado. La representación de un sistema en el espacio de estados no es única, salvo que el espacio de estados es el espacio vectorial en el cual el vector de estado toma valores, teniendo por lo tanto la misma dimensión que el número de elementos de dicho vector [10].

Las variables de estado describen el estado dinámico de un sistema. Así como el estado recoge toda la información del sistema en un determinado instante [10].

En un sistema que hay r entradas $u_1(t)$, $u_2(t)$, ..., $u_r(t)$ y m salidas $y_1(t)$, $y_2(t)$, ..., $y_m(t)$. Se define las n salidas de los integradores como variables de estado $x_1(t)$, $x_2(t)$, ..., $x_n(t)$. El sistema se puede describir de la siguiente forma: [7]

$$\dot{x}_1(t) = f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t)$$

$$\dot{x}_2(t) = f_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t)$$

.

.

.

$$\dot{x}_n(t) = f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t)$$

Las salidas del sistema se obtienen de la siguiente forma:

$$y_1(t) = g_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t)$$

$$y_2(t) = g_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t)$$

.

.

.

$$y_n(t) = g_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t)$$

Si se define

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}, \quad f(x, u, t) = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ f_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ \vdots \\ f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \end{bmatrix}$$

$$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_m(t) \end{bmatrix}, \quad g(x, u, t) = \begin{bmatrix} g_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ g_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ \vdots \\ g_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \end{bmatrix}$$

$$u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_r(t) \end{bmatrix}$$

Las ecuaciones se convierten en:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \text{ ecuación de estado}$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}, \mathbf{u}, t) \text{ ecuación de salida}$$

si se linealizan las ecuaciones de estado y de salida obtenemos:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \text{ ecuación de estado}$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) \text{ ecuación de salida}$$

Donde:

$\mathbf{x}(t)$ es el vector de estado, de dimensión n

$\mathbf{u}(t)$ es el vector de entrada, de dimensión m

$\mathbf{y}(t)$ es el vector de estado, de dimensión p

$\mathbf{A}(t)$ es la matriz del sistema, de dimensión $n \times n$

$\mathbf{B}(t)$ es la matriz de entradas, de dimensión $n \times m$

$\mathbf{C}(t)$ es la matriz de salida, de dimensión $p \times n$

$\mathbf{D}(t)$ es la matriz de transmisión directa, de dimensión $p \times m$

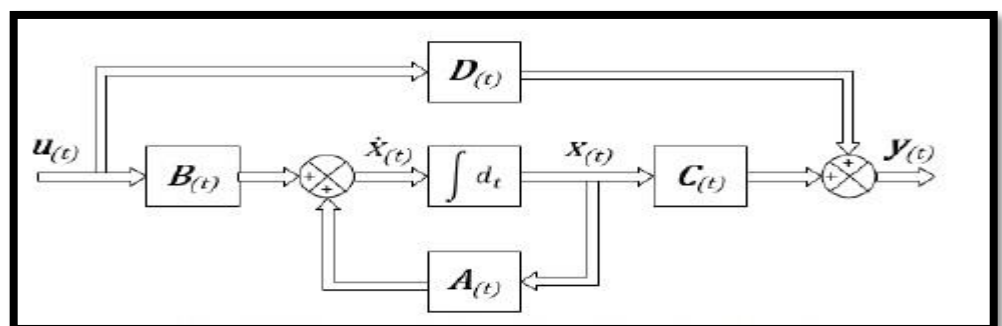


FIGURA 5: DIAGRAMA DE BLOQUES DEL SISTEMA DE CONTROL REPRESENTADO EN EL ESPACIO DE ESTADOS.
[7]

2.4. Representación en Control Predictivo.

El controlador predictivo basado en modelo o MPC se ha desarrollado considerablemente en las últimas décadas tanto en la industria como en la comunidad de investigación. Este éxito se debe a que el control predictivo basado en modelo es quizás la forma más general de formular el problema de control en el dominio del tiempo y es capaz de solucionar gran variedad de problemas de control multivariantes con restricciones. El control predictivo ha demostrado ser en la práctica una estrategia razonable de control y ha sido aplicado con éxitos numerosos procesos industriales. [11].

Tres aspectos principales del modelo de control predictivo hacen que la metodología de diseño sea atractiva tanto para ingenieros como para académicos. El primer aspecto es la formulación del diseño, que utiliza un marco de sistema multivariable donde los parámetros de rendimiento del sistema de control multivariable están relacionados con los aspectos de ingeniería del sistema. El segundo aspecto es la capacidad del método para manejar tanto las restricciones "blandas" como las duras en un marco de control multivariable. Esto es particularmente atractivo para la industria donde inevitablemente están presentes márgenes de ganancia ajustados y límites en la operación del proceso. El tercer aspecto es la capacidad para realizar procesos de optimización en línea. [12].

En la mayoría de las aplicaciones, MPC está diseñado para seguir señales de referencia constantes y rechazar las perturbaciones que tienen contenidos predominantemente de baja frecuencia. Sin embargo, estos tipos de controladores predictivos modelo no son adecuados para proporcionar el rendimiento de bucle cerrado requerido cuando las señales de referencia y las perturbaciones son complejas y tienen componentes periódicos. [12].

3. MARCO METODOLÓGICO

Para este proyecto de titulación se procedió analizar el funcionamiento del Aparatus Péndulo digital 33.936s, para reconocer las variables de entrada y salida del sistema. Una vez identificado el sistema (SIMO) se realiza el modelado de la planta para analizar si el sistema es controlable. Una vez comprobado la controlabilidad del sistema se comienza con los diseños de los diferentes algoritmos de control.

3.1.Modelo Matemático del Péndulo Invertido Simple

El modelado matemático de un sistema explica el comportamiento físico que por medio de ecuaciones captura la dinámica del proceso. En este capítulo se presenta la descripción física del péndulo invertido basadas en las leyes de Newton, para después realizar un análisis matemático y obtener las ecuaciones que describen el comportamiento del mismo [13]. Posteriormente, con las ecuaciones de la planta obtenidas se diseña el modelo en el software MATLAB/SIMULINK y se implementa los algoritmos de control.

3.1.1. Análisis Descriptivo de la Planta del Péndulo Digital modelo 33-936s marca Feedback

El primer paso para el diseño de un algoritmo de control capaz de mantener estable el péndulo es obtener el modelo matemático del sistema. El proyecto se basa en las características físicas del péndulo digital marca Feedback modelo 33-936s, con el fin obtener un modelo matemático y así poder reflejar el comportamiento y simulación de un sistema real.

El péndulo invertido consiste en un carro que se desplaza por un riel de 1 metro de longitud. El eje del carro está sujeto un péndulo con dos brazos el mismo que puede moverse libremente. El carro se mueve de manera horizontal hacia derecha y hacia la izquierda provocando una oscilación en el péndulo.

El movimiento del carro es dado por la tracción de una cadena acoplada al eje de un motor DC ubicado al final del riel. El voltaje aplicado al motor es quien permite mover el carro del péndulo en las dos direcciones. Las dos variables que son leídas por el péndulo (usando encoder's ópticos) son posición y ángulo del brazo del péndulo. [6]

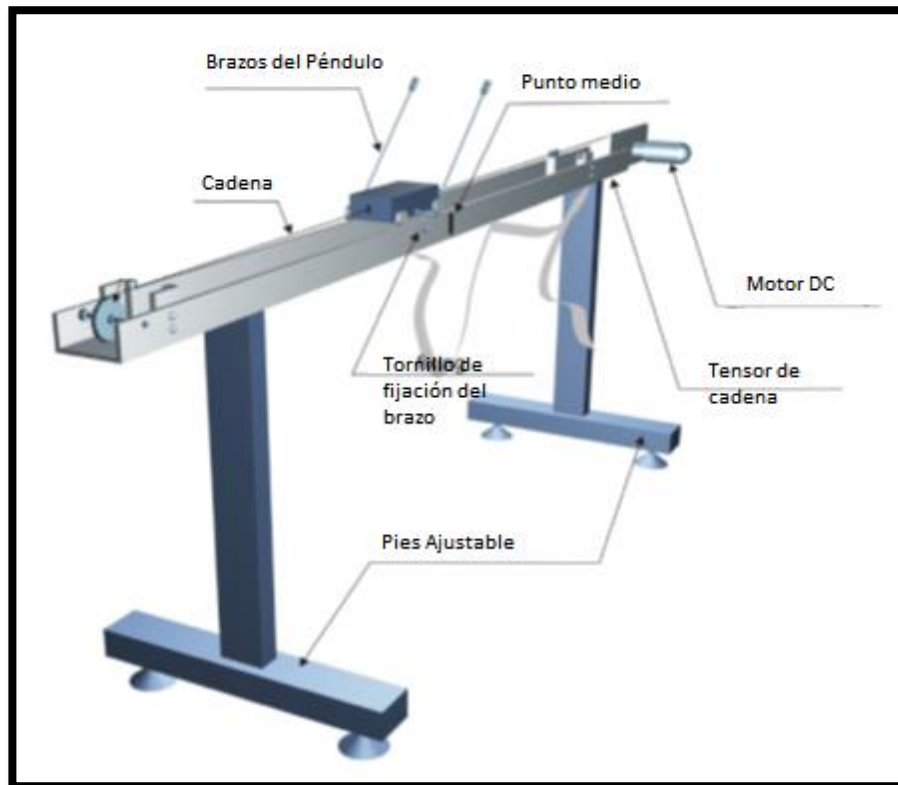


FIGURA 4: ESTRUCTURA MECÁNICA DEL PÉNDULO INVERTIDO. [6]

3.1.2. Modelamiento Matemático del Péndulo Digital modelo 33-936s marca Feedback

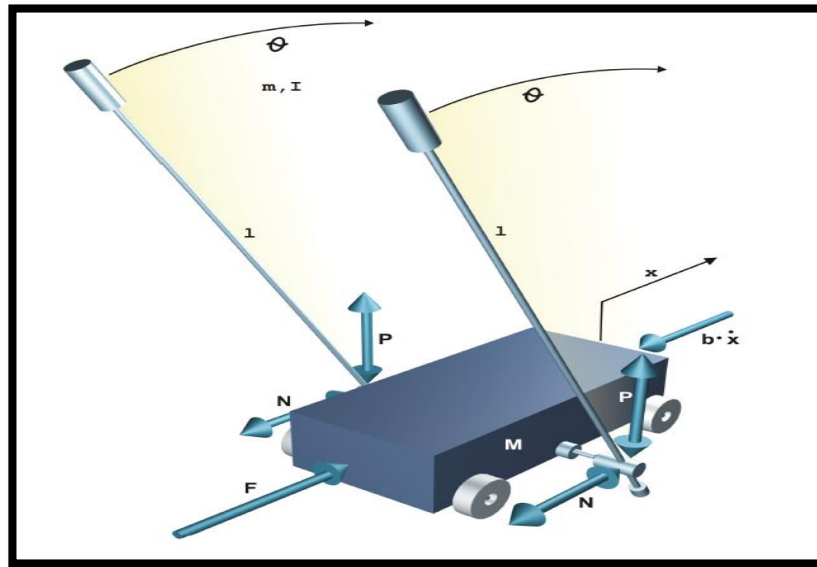


FIGURA 5: DIAGRAMA MECÁNICO DEL PÉNDULO INVERTIDO. [6]

El modelo mecánico del péndulo es un sistema no-lineal. Para realizar el análisis matemático se utiliza como base el análisis de Lagrangian [6].

Las limitaciones del sistema son:

- El carro tiene una longitud de riel para desplazarse de 1m, por lo que inicialmente se lo ubica en el centro del riel, para así tener 0,5 m para desplazarse en los dos lados.
- El voltaje aplicado al motor en DC es +2.5 V a -2.5V

TABLA 1 PARÁMETROS DEL PÉNDULO. [6]

Símbolo	Parámetro
G	Gravedad
L	Longitud de la varilla
M	Masa del carro
M	Masa del péndulo
I	Momento de inercia

B	Coeficiente de fricción
---	-------------------------

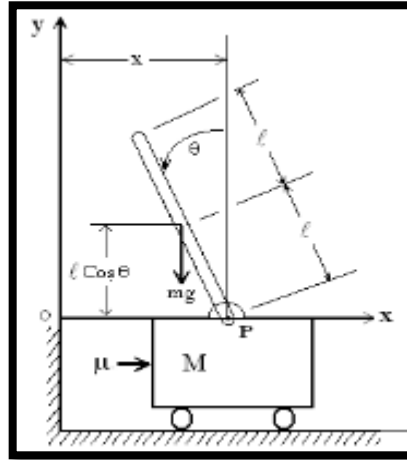


FIGURA 6 : DIAGRAMA CUERPO LIBRE DEL PÉNDULO INVERTIDO. [7]

La ecuación de Langrangian es dada por:

$$\frac{dy}{dx} \left(\frac{\partial L}{\partial \Sigma p_1} \right) - \frac{\partial L}{\partial \Sigma p_i} = \Sigma p_i$$

La función de Langrangian L es:

$$L = \text{energía cinética} - \text{energía potencial} = T - V$$

Σp_i es una fuerza no conservativa y está dada por:

$$\Sigma p_1 = F(t) - b\dot{x} ; \Sigma p_2 = D\theta$$

La energía cinética del sistema es descrita de la siguiente forma:

$$T = \frac{1}{2} M \dot{x}^2 + \frac{1}{2} I \dot{\theta}^2 + \frac{1}{2} m (\dot{x}^2 - 2\dot{x}\dot{\theta}L\cos\theta + L^2\dot{\theta}^2)$$

La energía potencial del sistema está dada por:

$$V = mgl\cos\theta$$

La Función de Langrangian se convierte:

$$L = \frac{1}{2} M \dot{x}^2 + \frac{1}{2} I \dot{\theta}^2 + \frac{1}{2} m (\dot{x}^2 - 2\dot{x}\dot{\theta}L\cos\theta + L^2\dot{\theta}^2) - mlg\cos\theta$$

Utilizando la ecuación de Langrangian utilizando i=1, se obtiene la siguiente ecuación no-lineal para la posición del carro x.

$$U = (M + m)\ddot{x} + b.\dot{x} - m.l.\ddot{\theta}.\cos\theta - m.l.\dot{\theta}^2.\sin\theta$$

Utilizando la ecuación de Langrangian utilizando i=2, se obtiene la siguiente ecuación no-lineal para el ángulo del péndulo.

$$0 = (I + ml^2)\ddot{\theta} - m.l.\ddot{x}.\cos\theta - m.g.l.\sin\theta + D.$$

El sistema del péndulo invertido tiene los siguientes parámetros:

TABLA 2: VALORES PARÁMETROS DEL PÉNDULO [6]

Símbolo	Parámetro	Valor
g	Gravedad	9.81 m/s ²
l	Longitud de la varilla	0.36 a 0.4 m
M	Masa del carro	2.4kg
m	Masa del péndulo	0.23kg
I	Momento de inercia	0.099 Kg.m ²
b	Coeficiente de fricción	0.05 Ns/m
d	Coeficiente de caída (despreciable)	0

3.1.3. Linealización del Modelo

Para analizar la dinámica del sistema se linealiza las ecuaciones matemáticas antes obtenidas. En el caso del péndulo invertido se tiene dos puntos de equilibrio cuando $\theta = 0$ (péndulo invertido) y cuando $\theta = \pi$ (control de grúa).

$$\sin\theta = \theta$$

$$\cos\theta = 1$$

$$\dot{\theta}^2 = 0$$

Reemplazando se obtiene el modelo lineal:

$$U = (M + m)\ddot{x} + b.\dot{x} + m.l.\ddot{\theta}$$

$$0 = (I + ml^2)\ddot{\theta} + m.l.\ddot{x} - m.g.l\theta + D\dot{\theta}$$

3.1.4. Representación mediante Variables de estado

En la representación en variables de estado para el péndulo invertido sabemos que tenemos un sistema SIMO (sistema una entrada con múltiple salida), donde la entrada u es la señal control y las salidas son posición del carro y ángulo del péndulo.

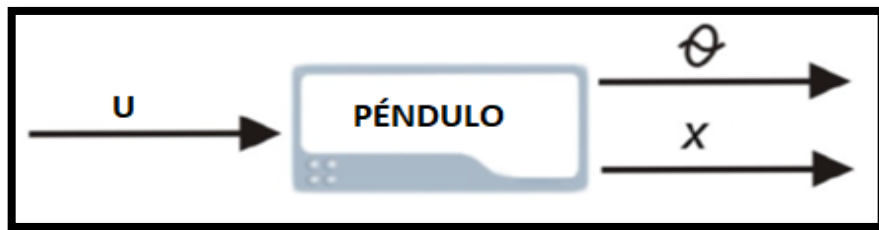


FIGURA 9: DIAGRAMA DE BLOQUE DEL SISTEMA PÉNDULO INVERTIDO. [6]

Las ecuaciones en variables de estado de un sistema están definidas mediante las ecuaciones siguientes:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \text{ ecuación de estado}$$

$$y(t) = C(t)x(t) + D(t)u(t) \text{ ecuación de salida}$$

Con el conocimiento de las propiedades físicas del péndulo invertido se escoge las dos variables de estado como sigue:

$$x_1 = \theta;$$

$$x_2 = \dot{\theta} = \dot{x}_1$$

$$x_3 = x$$

$$x_4 = \dot{x}_3 = \dot{x}$$

Usando las ecuaciones del Modelo Matemático Linealizado se obtiene el sistema de ecuaciones en variables de estado.

$$U = (M + m)\ddot{x} + b.\dot{x} - m.l.\ddot{\theta} \quad (1)$$

$$0 = (I + ml^2)\ddot{\theta} - m.l.\ddot{x} - m.g.l\theta + D\dot{\theta} \quad (2)$$

Reemplazando $\dot{x}_4$ a las ecuaciones 1 y 2

$$\begin{aligned} \dot{x}_2 = & \frac{mgl(M + m)}{I(M + m) + Mml^2}x_1 + \frac{-D(M + m)}{I(M + m) + Mml^2}x_2 + \frac{-mlb}{I(M + m) + Mml^2}x_4 \\ & + \frac{ml}{I(M + m) + Mml^2}F \end{aligned}$$

$$\begin{aligned} \dot{x}_4 = & \frac{gm^2l^2}{I(M + m) + Mml^2}x_1 + \frac{-Dml}{I(M + m) + Mml^2}x_2 + \frac{-b(I + ml^2)}{I(m + m) + Mml^2}x_3 \\ & + \frac{(I + ml^2)}{I(M + m) + Mml^2}F \end{aligned}$$

Por tanto, queda definida la entrada y las salidas del sistema de la siguiente forma:

Entrada:

$$U = F(t)$$

Salidas:

$$y1(t) = \theta(t)$$

$$y2(t) = X(t)$$

Construyendo las matrices en Variables de Estado:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

$$y(t) = C(t)x(t)$$

$$\dot{x} = \begin{bmatrix} \dot{\theta} \\ \dot{\theta} \\ \dot{x} \\ \dot{x} \end{bmatrix}; \quad U = F(t); \quad Y = \begin{bmatrix} \theta(t) \\ x(t) \end{bmatrix}$$

$$; A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{mgl(M+m)}{I(M+m)+Mml^2} & \frac{-D(M+m)}{I(M+m)+Mml^2} & 0 & \frac{-mlb}{I(M+m)+Mml^2} \\ 0 & 0 & 0 & 1 \\ \frac{gm^2l^2}{I(m+m)+Mml^2} & \frac{-mlD}{I(M+m)+Mml^2} & 0 & \frac{-b(I+Ml^2)}{I(m+m)+Mml^2} \end{bmatrix} B = \begin{bmatrix} 0 \\ ml \\ 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \end{bmatrix}; C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Reemplazando con los parámetros del péndulo:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 6.4363 & -0.0396 & 0 & -0.0125 \\ 0 & 0 & 0 & 1 \\ 0.2026 & -0.012 & 0 & -0.0194 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 0.2495 \\ 0 \\ 0.3881 \end{bmatrix}; C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

3.2. Análisis del modelo del Péndulo Invertido en el espacio de estados

Las ecuaciones en variables de estado del Péndulo Invertido se obtienen mediante el código de Matlab denominado VARIABLES_ESTADO.m, dicho código se muestra a continuación:

```
clear all
close all
clc

% Parámetros del sistema

M=2.4;
m=0.23;
g=9.81;
l=0.36;
I=0.099;
b=0.05;
d=0.005;

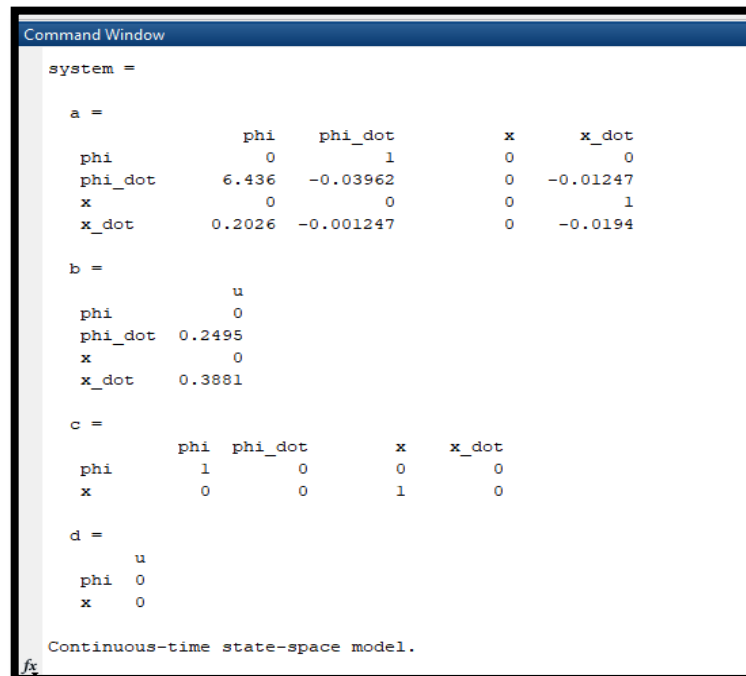
%ecuaciones en variables de estado del sistema
%theta - xdot
z= [I*(M+m)+(M*m*l*l)];
y= (I+(m*l*l));
A=[ 0 1 0 0 ; ((M+m)*m*g*l)/z -(M+m)*d/z 0 -(m*l*b)/z; 0 0 0 1;
(m*m*l*l*g)/z -(d*m*l)/z 0 -(b*y)/z]
B=[0; m*l/z; 0; (y)/(z)]
C=[1, 0, 0, 0; 0, 0, 1, 0]
D=[0;0]

%Representación de variables

%theta - xdot
s = tf('s');
states = {'phi' 'phi_dot' 'x' 'x_dot'};
inputs = {'u'};
outputs = {'phi'; 'x'};
%Representación en el espacio de estados
%Representación en el espacio de estados

system = ss(A,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs)
```

Resultado en software de Matlab:



```
Command Window

system =

a =

    phi    phi_dot    x    x_dot
    phi      0      1      0      0
    phi_dot  6.436 -0.03962  0 -0.01247
    x        0      0      0      1
    x_dot    0.2026 -0.001247  0 -0.0194

b =

    u
    phi
    phi_dot  0.2495
    x
    x_dot    0.3881

c =

    phi    phi_dot    x    x_dot
    phi      1      0      0      0
    x        0      0      1      0

d =

    u
    phi
    x

Continuous-time state-space model.
```

FIGURA 7: VARIABLES DE ESTADO DEL PÉNDULO INVERTIDO

3.2.1. Estabilidad del Péndulo

La estabilidad es la propiedad de un sistema de permanecer en estado de equilibrio cuando no está afectado por perturbaciones ni por fuerzas externas. Cuando todos los polos del sistema están ubicados en el semiplano izquierdo del plano S del dominio de Laplace el sistema es estable.

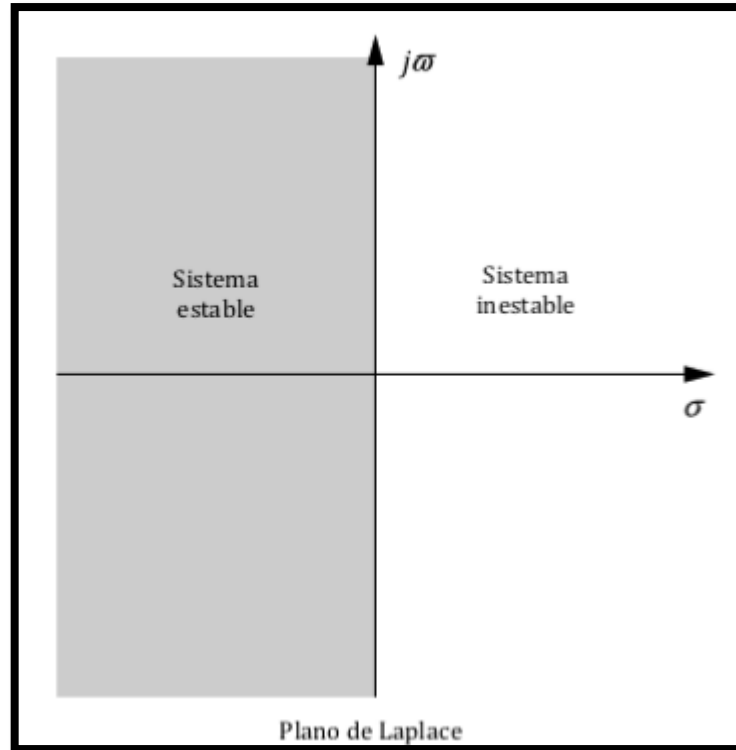


FIGURA 8: PLANO ESTABILIDAD EN EL DOMINIO DE LAPLACE [9]

Por tanto, para conocer la estabilidad del sistema se procede a determinar los polos y ceros del sistema y así conocer la estabilidad del mismo, para ello usamos el siguiente código en Matlab:

Código Digitado en Matlab:

```
% determinar estabilidad
```

```
Polos = eig(A)
```

```
coa = 0;
```

```
for co = 1:max(size(Polos))
```

```
    if Polos(co)>=0
```

```
        disp('El sistema es Inestable.')
```

```
        coa = 1;
```

```
        break
```

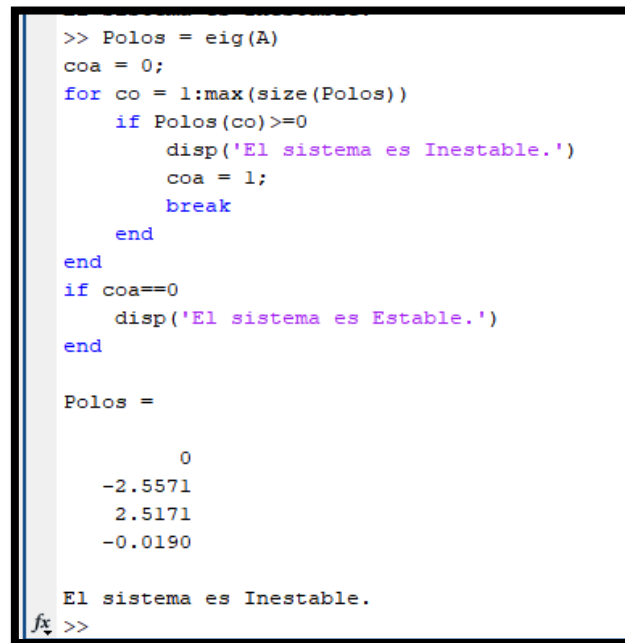
```
    end
```

```

end
if coa==0
    disp('El sistema es Estable.')
end

```

Respuesta de Matlab es:



```

>> Polos = eig(A)
coa = 0;
for co = 1:max(size(Polos))
    if Polos(co)>=0
        disp('El sistema es Inestable.')
        coa = 1;
        break
    end
end
if coa==0
    disp('El sistema es Estable.')
end

Polos =

         0
    -2.5571
     2.5171
    -0.0190

El sistema es Inestable.
fx >>

```

FIGURA 9: POLOS DEL SISTEMA PÉNDULO INVERTIDO

Se obtiene que el sistema es inestable dado que tiene un polo está ubicado en el lado derecho del semiplano S, el polo está ubicado en 2.5171.

3.2.2. Controlabilidad del Péndulo

El estudio de la controlabilidad de un sistema determina si las variables de estado del mismo son controlables o no son controlables, es decir, que los estados del sistema pueden ser alcanzados actuando sobre las entradas de éste, un sistema se dice controlable si todos los puntos de su espacio de estado son controlables.

Para conocer la controlabilidad del sistema se procede con el siguiente código de Matlab.

Código Digitado en Matlab:

```
% controlabilidad
```

```
Cont = [B A*B (A^2)*B (A^3)*B];
```

```
if rank(Cont)==4
```

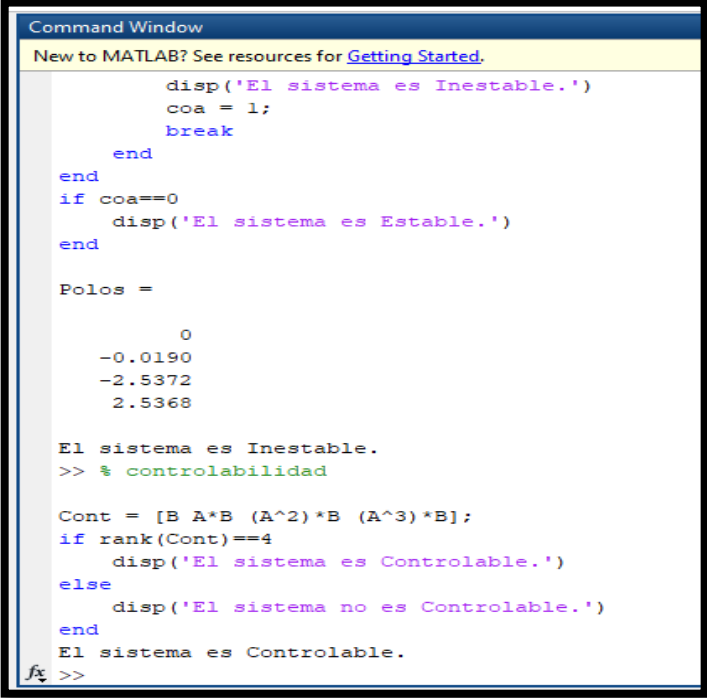
```
    disp('El sistema es Controlable.')
```

```
else
```

```
    disp('El sistema no es Controlable.')
```

```
end
```

Respuesta de Matlab es:



```
Command Window
New to MATLAB? See resources for Getting Started.

    disp('El sistema es Inestable.')
    coa = 1;
    break
end
end
if coa==0
    disp('El sistema es Estable.')
end

Polos =

     0
-0.0190
-2.5372
 2.5368

El sistema es Inestable.
>> % controlabilidad

Cont = [B A*B (A^2)*B (A^3)*B];
if rank(Cont)==4
    disp('El sistema es Controlable.')
else
    disp('El sistema no es Controlable.')
end
El sistema es Controlable.
fx >>
```

FIGURA 10: COMPROBACIÓN DE CONTROLABILIDAD DEL SISTEMA (PÉNDULO INVERTIDO) EN MATLAB

3.2.3. Observabilidad del Péndulo

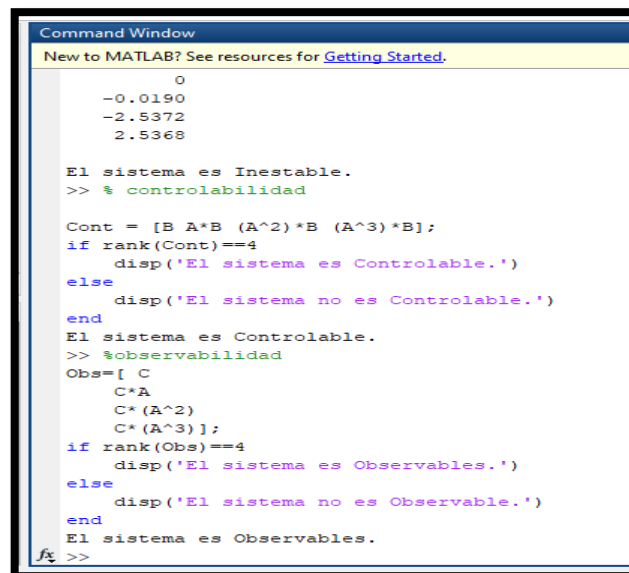
La idea de la observabilidad se relaciona con la posibilidad de obtener información del valor del estado de un sistema a partir de mediciones de la evolución de estado y salida por lo tanto se conoce la observabilidad como el estudio del estado-salida. Una vez conocido el estado en un instante inicial, se puede determinar el estado en cualquier otro instante posterior utilizando la solución de la ecuación de estado [10]. Se dice que un sistema es observable si todos los puntos del espacio de estado se pueden obtener información de las mediciones de las salidas y las entradas [7].

Código Digitado en Matlab:

```
% controlabilidad

Cont = [B A*B (A^2)*B (A^3)*B];
if rank(Cont)==4
    disp('El sistema es Controlable.')
else
    disp('El sistema no es Controlable.')
end
```

Respuesta de Matlab es:



```
Command Window
New to MATLAB? See resources for Getting Started.

0
-0.0190
-2.5372
2.5368

El sistema es Inestable.
>> % controlabilidad

Cont = [B A*B (A^2)*B (A^3)*B];
if rank(Cont)==4
    disp('El sistema es Controlable.')
else
    disp('El sistema no es Controlable.')
end
El sistema es Controlable.
>> %observabilidad
Obs=[ C
      C*A
      C*(A^2)
      C*(A^3)];
if rank(Obs)==4
    disp('El sistema es Observables.')
else
    disp('El sistema no es Observable.')
end
El sistema es Observables.
fx >>
```

FIGURA 11: COMPROBACIÓN DE OBSERVABILIDAD DEL SISTEMA (PÉNDULO INVERTIDO) EN MATLAB

3.3.Respuesta del sistema en lazo abierto

Un sistema en lazo abierto funciona sin retroalimentación y genera directamente la salida en respuesta a una señal de entrada.

A continuación, se presenta el código y las gráficas generadas de las respuestas tipo Paso, Impulso y Rampa, aplicadas al modelo del sistema péndulo invertido.

```
T = 0:0.01:1;  
impulse(system,T);  
title('Respuesta del Sistema en Lazo Abierto a entrada tipo Impulso')  
grid on
```

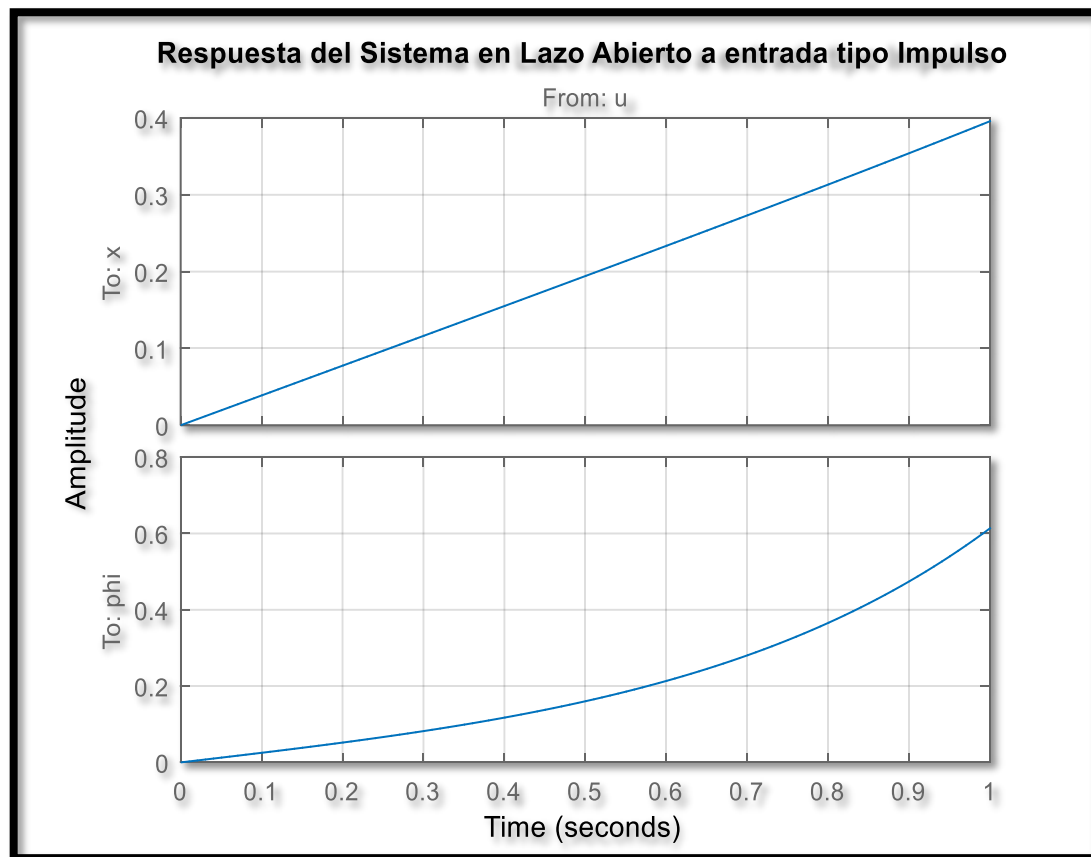


FIGURA 12: RESPUESTA DEL SISTEMA A ENTRADA TIPO IMPULSO

```
%Respuesta a tipo Paso
T = 0:0.001:6;
step(system,T);
title('Respuesta del Sistema en Lazo Abierto a entrada tipo Paso')
grid on
```

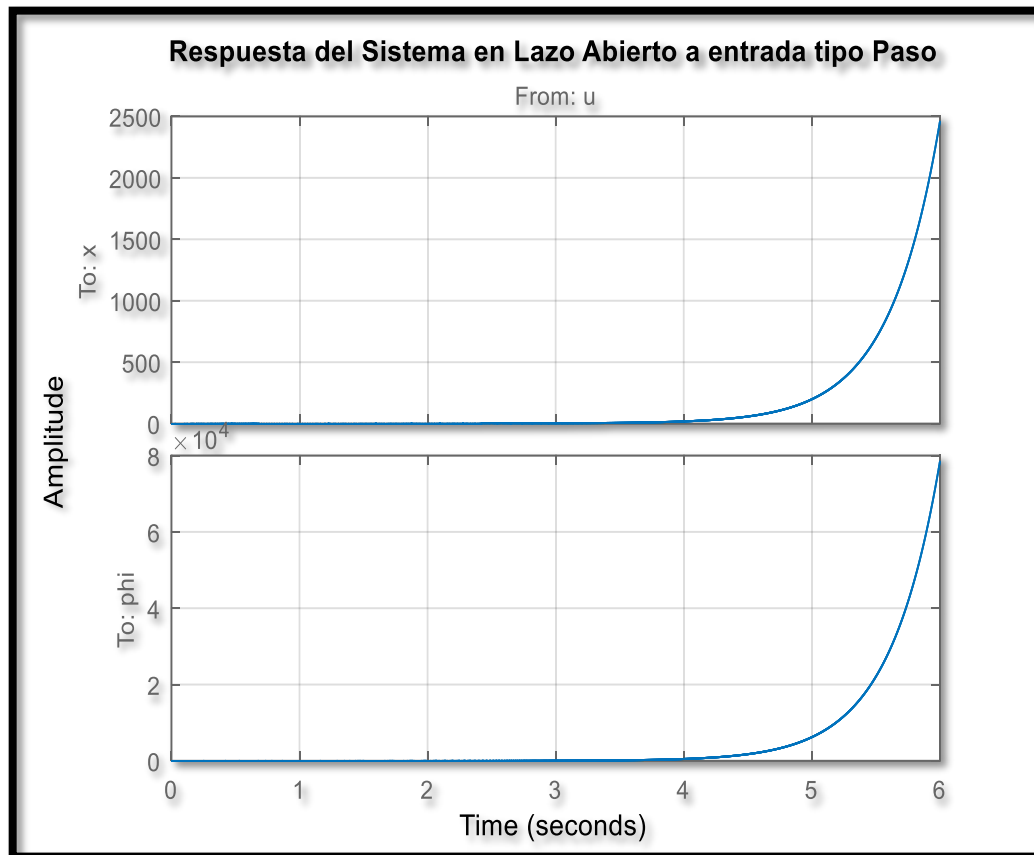


FIGURA 13 RESPUESTA DEL SISTEMA A ENTRADA TIPO PASO

También se puede trabajar en Simulink, para ello se ingresa el modelo linealizado del péndulo como se muestra en la figura 14.

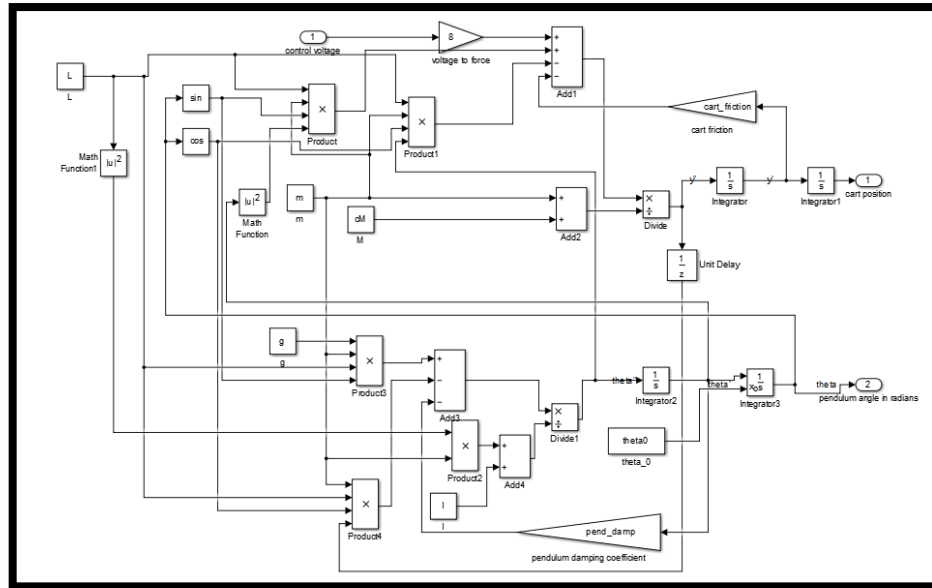


FIGURA 14: MODELO DEL PÉNDULO INVERTIDO. [6]

Análisis en lazo abierto en Simulink:

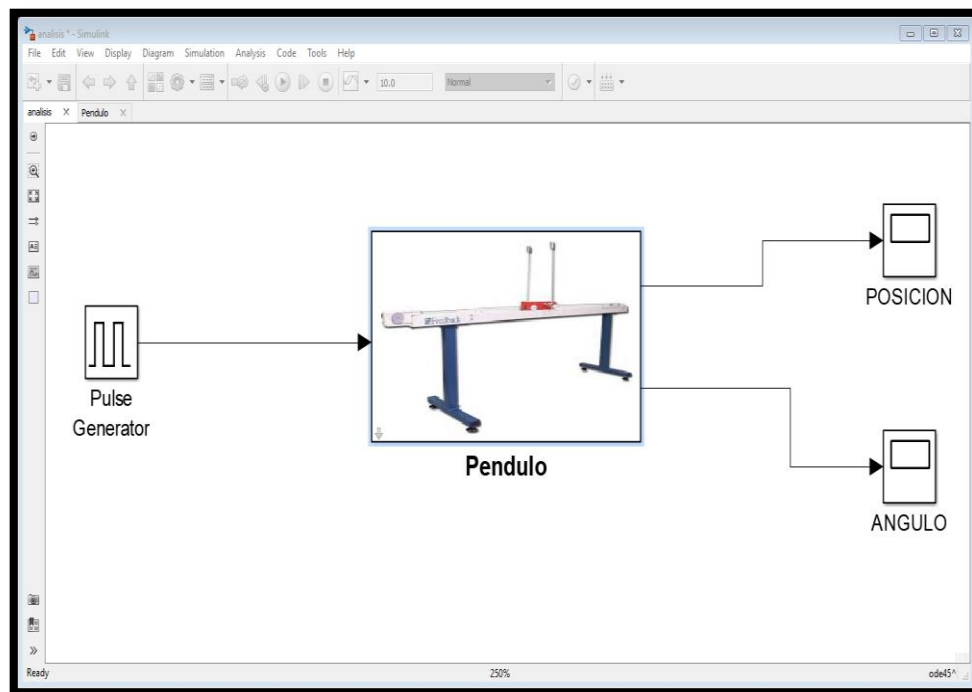


FIGURA 15: SISTEMA LAZO ABIERTO DEL PÉNDULO INVERTIDO EN SIMULINK

Para la visualización del análisis se han añadido bloques tipo “Scope” y para revisar las gráficas se accede a los bloques una vez ejecutada la simulación.

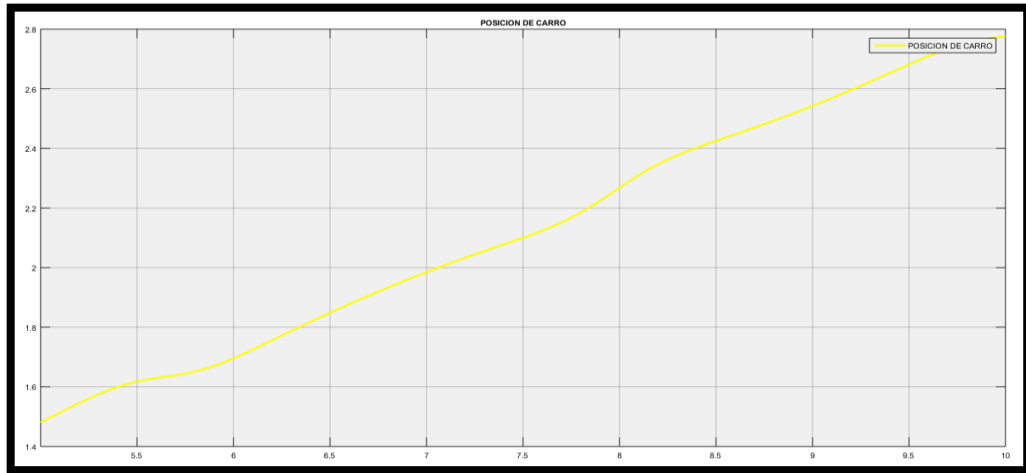


FIGURA 16 GRÁFICA DE POSICIÓN DEL CARRO EN LAZO ABIERTO DEL SISTEMA PÉNDULO INVERTIDO

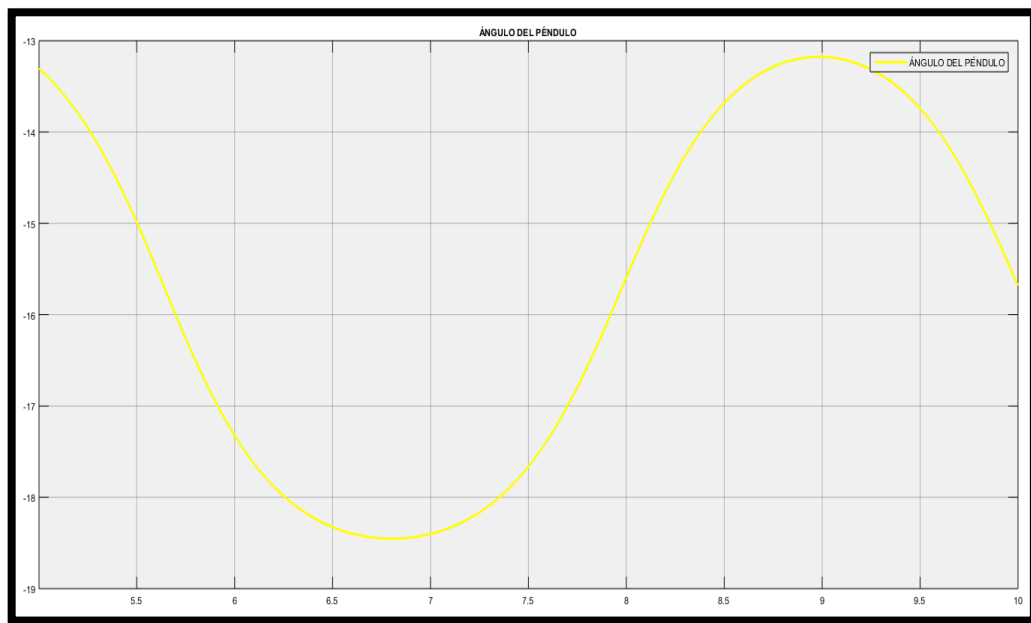


FIGURA 17: GRÁFICA DE ÁNGULO DEL PÉNDULO EN LAZO ABIERTO DEL SISTEMA PÉNDULO INVERTIDO

3.4.Respuesta del sistema en lazo cerrado

Un sistema en lazo cerrado consiste en retroalimentar la señal de salida para ser comparada con la señal de entrada o de referencia y determinar el error que está presente en el mismo [8].

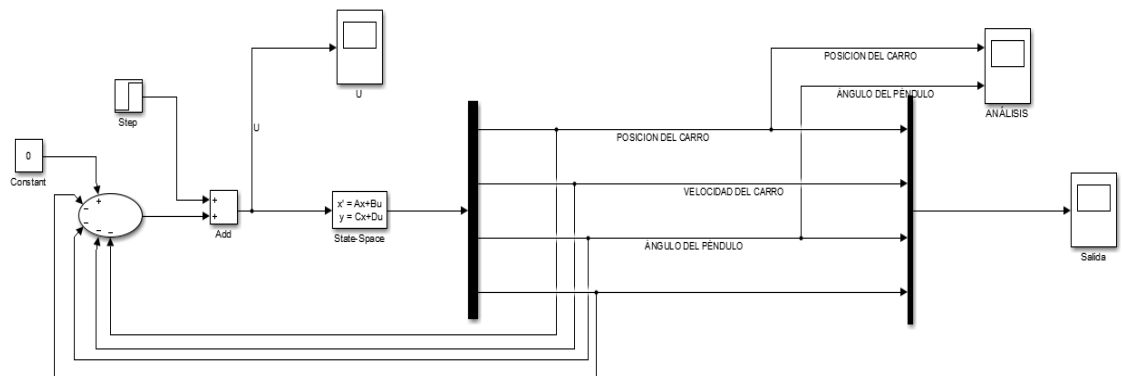


FIGURA 18: SISTEMA REALIMENTADO EN VARIABLES ESTADO

Para la visualización del análisis se han añadido bloques tipo “Scope” y para revisar las gráficas se accede a los bloques una vez ejecutada la simulación.

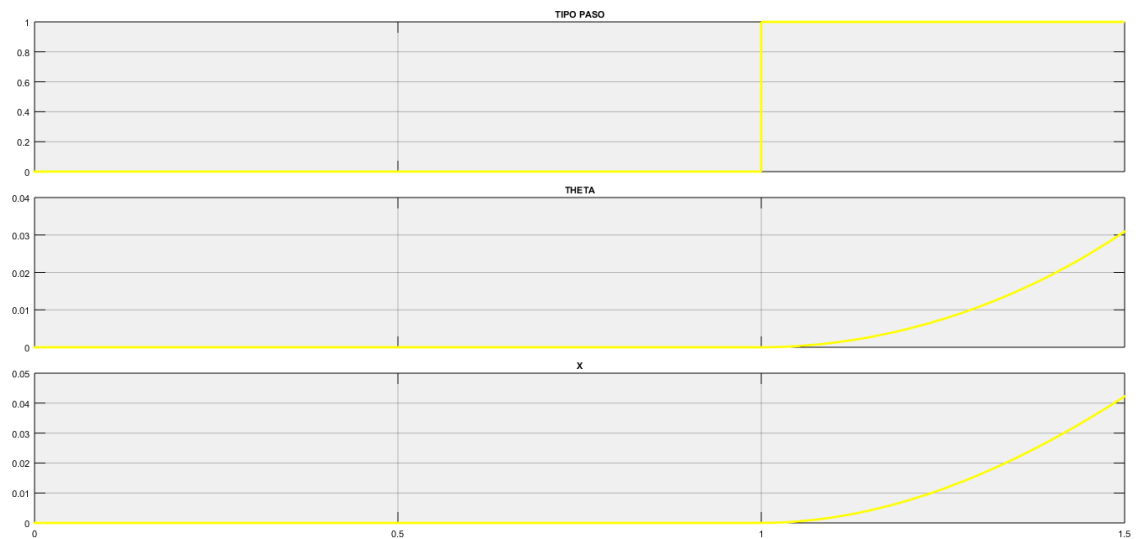


FIGURA 19: RESPUESTA DEL SISTEMA A ENTRADA TIPO PASO

3.5. Diseño del controlador para la estabilización del Péndulo

Una vez realizado el estudio del modelo del Péndulo invertido se determina que la dinámica del sistema no es capaz de estabilizar el péndulo por sí sola, por tanto, se requiere diseñar controladores para la estabilización del péndulo [9]. Existen diferentes tipos de algoritmo de control en el estudio de la Teoría de Control, pero en este caso de estudio se concentra en 3 tipos de algoritmos, esto es, mediante el método de variables de estado, control PID y control PREDICTIVO basado en modelo (MPC).

El diseño de los controladores se realiza en el software Matlab/Simulink con sus diferentes herramientas.

3.5.1. Controlador por Variables de estado

La representación del sistema del péndulo invertido en el espacio de estados realizados en el capítulo 4.1.4 son:

$$A = \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix} \begin{bmatrix} x & \dot{x} & \theta & \dot{\theta} \\ 0 & 1 & 0 & 0 \\ 6.436 & -0.03962 & 0 & -0.01247 \\ 0 & 0 & 0 & 1 \\ 0.2026 & -0.01247 & 0 & -0.0194 \end{bmatrix}; B = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \begin{bmatrix} u \\ 0 \\ 0.2495 \\ 0 \\ 0.3881 \end{bmatrix}; C = \begin{bmatrix} x \\ \theta \end{bmatrix} \begin{bmatrix} x & \dot{x} & \theta & \dot{\theta} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Donde θ , $\dot{\theta}$, x , $\dot{x}$ corresponden a:

θ = ángulo del péndulo

$\dot{\theta}$ = velocidad angular del péndulo

x = posición del péndulo

$\dot{x}$ = velocidad del carro

Recapitulando, los polos del sistema son $s = \{0; -2.5571; 2.5171; -0.0190\}$ donde se puede evidenciar la inestabilidad del sistema ya que uno de sus polos se

encuentra en el lado derecho del semiplano del dominio de Laplace, según se muestra en la figura 24.

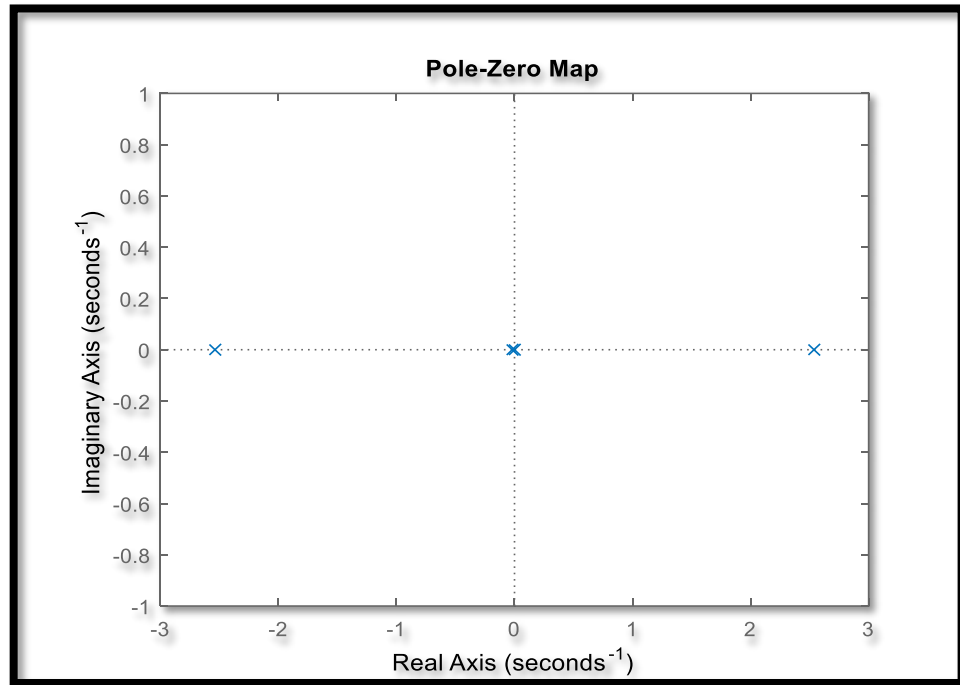


FIGURA 20: UBICACIÓN DE POLOS EN EL DOMINIO DE LAPLACE

Para que el sistema presente una respuesta temporal estable debe tener una realimentación de estados completo con el fin de obtener la situación deseada de los polos del sistema en lazo cerrado [8]. La única condición necesaria para realizar la asignación arbitraria de polos es que el sistema sea controlable [7]. Lo cual no obstaculiza el diseño de la realimentación de estados ya que el sistema es controlable y observable.

Este método tiene como principal objetivo determinar la matriz de ganancia $\mathbf{k}$ por medio la fórmula de Ackerman. [8].

3.5.1.1 Polos deseados

La técnica de ubicación de polos permite situar los polos del sistema en lazo cerrado en una posición deseada por el diseñador, de modo que su comportamiento se adecue a lo que éste desee. La única condición para ubicar los polos en cualquier posición del semiplano S, es que el sistema sea controlable.

De acuerdo al cumplimiento de las especificaciones de desempeño, esto es, sobrenivel porcentual de 5% y el tiempo de asentamiento de 5s, se obtiene los polos deseados en lazo cerrado.

$$s = \{-0.6230 + 0.2178i; -0.630; -0.2178i; -10 - 10\}$$

3.5.1.2 Diseño de controladores en variables de estado

Considerando al sistema la forma general:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \text{ ecuación de estado}$$

$$y(t) = C(t)x(t) + D(t)u(t) \text{ ecuación de salida}$$

para la cual existe una ley de control:

$$u = -Kx$$

Teniendo en cuenta que los valores de K del controlador son realimentados al sistema, obtenemos:

$$\dot{x} = Ax + B(-Kx)$$

$$\dot{x} = Ax - BKx$$

$$\dot{x} = (A - BK)x$$

Si la matriz que afecta a x estable, el sistema lleva a todos sus estados a cero.

$$|\lambda I - (A - BK)| = 0$$

Si y solo si, todos los valores propios λ tienen parte real negativa. Todos los sistemas controlados mediante esta ley de control llevan a todos los estados a

cero. La ganancia K determina cuál es la mejor señal de control u tomando como referencia el vector de estado actual x.

Los criterios de diseño del controlador son:

- Tiempo de asentamiento de x y θ menor que 5 segundos
- Máximo sobre impulso de 5/100

Teniendo los Polos deseados del sistema:

$$Polos = \{-0.6230 + 0.2178i; -0.6230; -0.2178i; -10 - 10\}$$

Mediante la fórmula de Ackerman usando Matlab se tiene:

$$K = \text{acker}(A, B, \text{Polos})$$

$$K = [559.28; 169.9197; -17.79812; -54.6333]$$

En la figura 22, se muestra el diseño en simulink del sistema de control por realimentacion de estados.

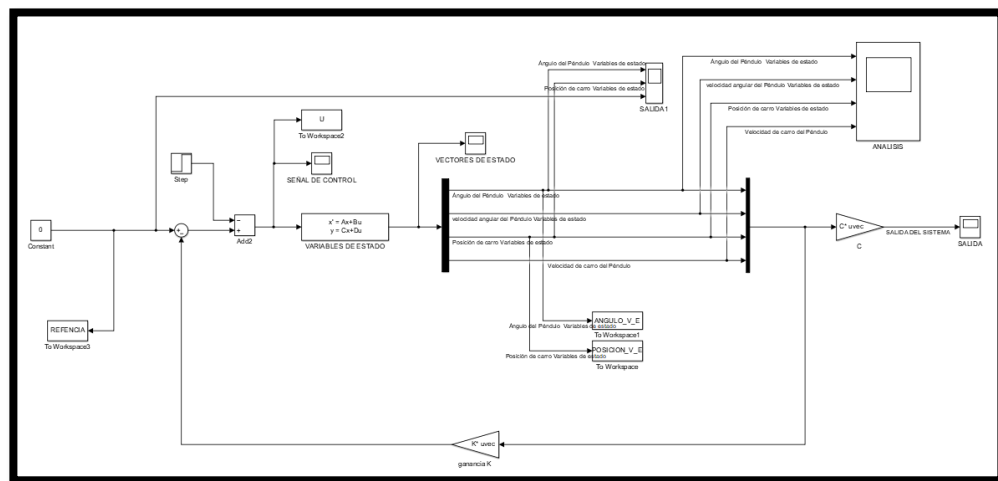


FIGURA 21: DISEÑO DEL ALGORITMO DE CONTROL EN VARIABLES DE ESTADO EN SIMULINK

En la figura 23, se muestra las señales obtenidas tanto de posición y ángulo para el sistema diseñado usando reubicación de polos deseados.

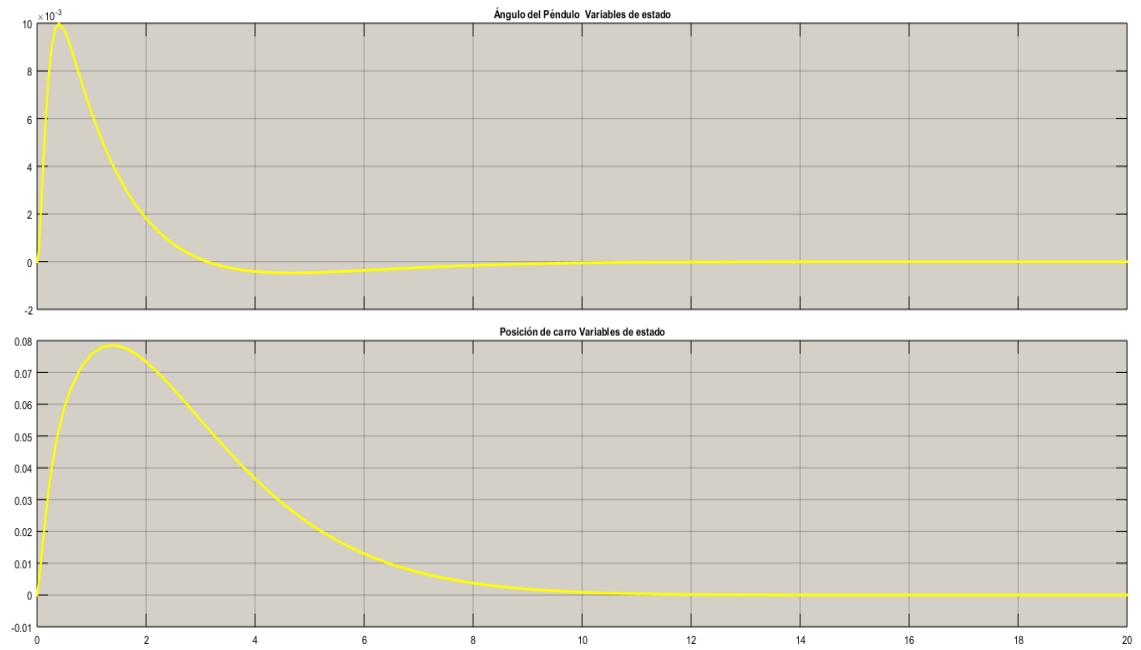


FIGURA 22: RESPUESTA DEL ALGORITMO DE CONTROL EN VARIABLES DE ESTADO SIMULADO EN SIMULINK

3.5.2. Controlador PID

El controlador PID es uno de los controladores más usados en la industria de control. el controlador PID son utilizados en plantas de sistemas SISO. Por lo tanto, para el sistema del péndulo Invertido SIMO se modifican los algoritmos considerando la combinación de dos controladores PID, uno para la estabilización del péndulo y otra para la posición del carro.

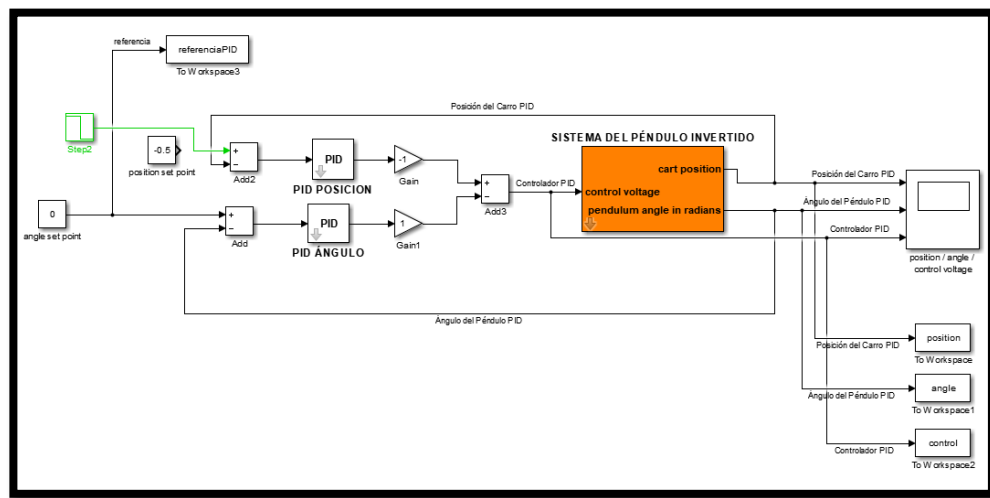


FIGURA 23: CONTROLADOR PID CONSIDERADO PARA LA PLANTA PÉNDULO INVERTIDO. [6]

Los tres términos del controlador PID son Proporcional, Integrador y Derivativo y su función es:

$$K_p + \frac{K_i}{s} + K_d s = \frac{K_p s + K_i + K_d s^2}{s}$$

- **Kp = Ganancia Proporcional**
- **Ki = Ganancia Integrativa**
- **KD = Ganancia Derivativa**

Para el diseño del controlador PID se toma como base los parámetros usados en la planta real, dichos datos fueron proporcionados por el fabricante del equipo, sin embargo, se realizaron ajustes para mejorar la respuesta del sistema en el presente estudio, por lo que se obtiene los siguientes parámetros que se muestran en la tabla 3.

TABLA 3 GANANCIAS PID [6]

Controlador	P	I	D
Posición	0.8	0.001	0.64
Péndulo	45	5	9

Al ejecutar el sistema con PID se obtiene los resultados que se muestran en la figura 25.

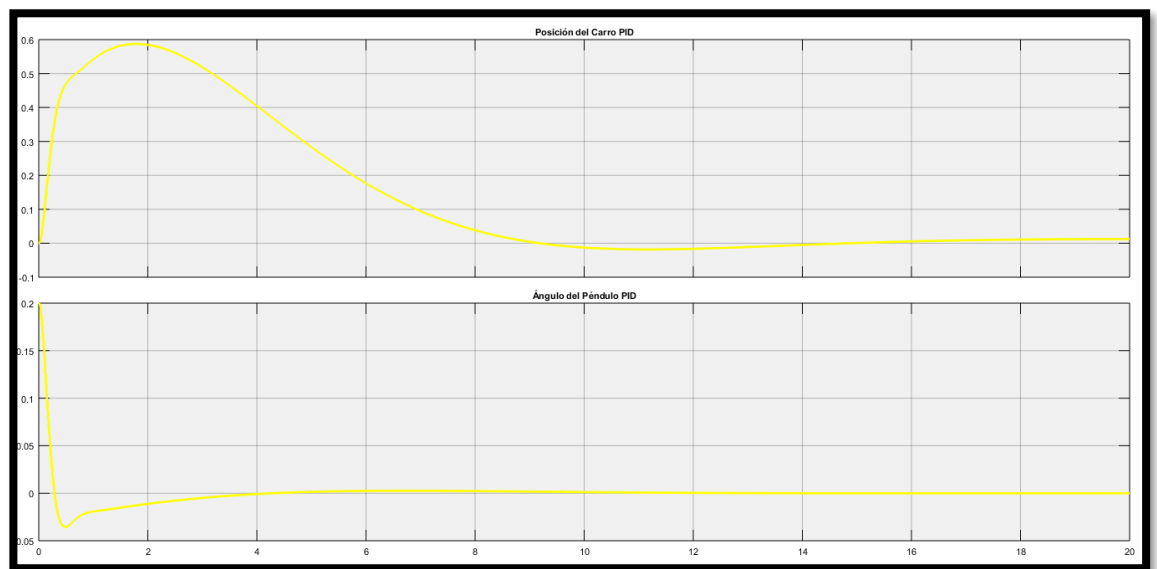


FIGURA 24: SALIDAS CONTROLADOR PID PARA LA PLANTA DEL PÉNDULO INVERTIDO

3.5.3. Diseño del controlador Predictivo MPC

El objetivo del controlador MPC consiste en el hallazgo de una trayectoria futura donde la estrategia de control del MPC consiste en fijar un horizonte de predicción N para el cual se van a predecir en cada instante K_i las salidas futuras. Esto se logra utilizando un modelo de predicción que describe el comportamiento de las variables del proceso a controlar el cual reside en el controlador. [12]

En la figura 26 se muestra un esquema que evidencia la operación de un control predictivo.

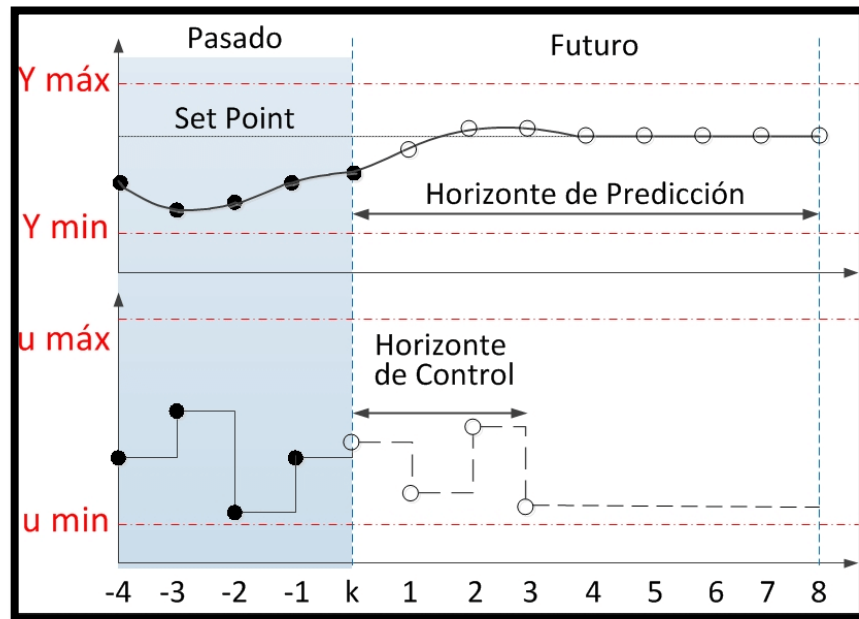


FIGURA 25: IDEA DE CONTROL PREDICTIVO. [11]

El diseño del controlador Predictivo basado en modelo “MPC” puede ser realizado en el espacio de estados a partir del modelo matemático de la planta. El Toolbox MPC Designer es una herramienta de diseño y análisis del controlador.

En la figura 27, se muestra la interface gráfica de Matlab para el diseño de un controlador MPC. Esta interface gráfica se obtiene al usar el comando MPC Designer.

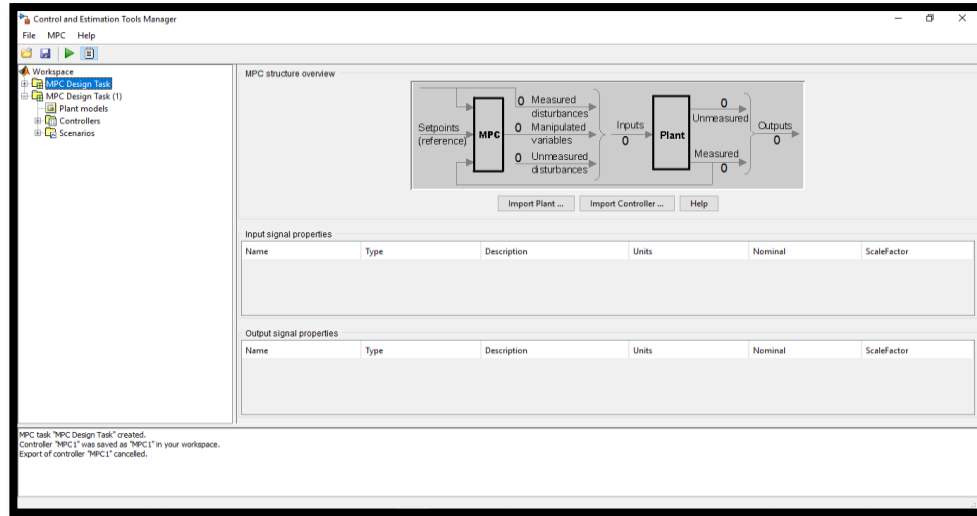


FIGURA 26: MPC DESIGNER TOOLBOX

Una vez abierta la interface gráfica, se procede a importar la planta desde el espacio de estados en la ventana MPC Structure. Según se muestra en la figura 28.

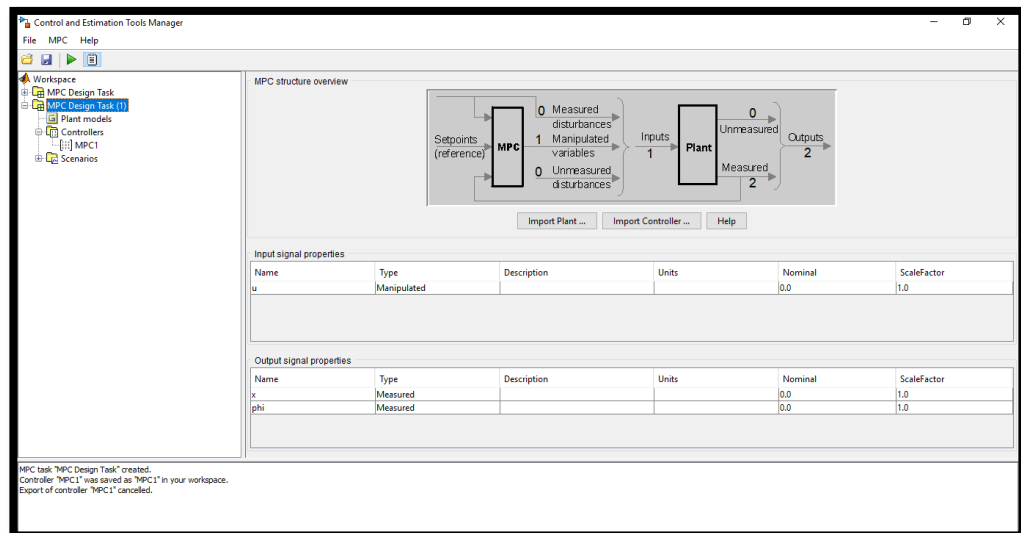


FIGURA 27: VENTANA MPC STRUCTURE.

Una vez importada la planta se puede diseñar el controlador, usando los escenarios de prueba para diferentes parámetros de predicción.

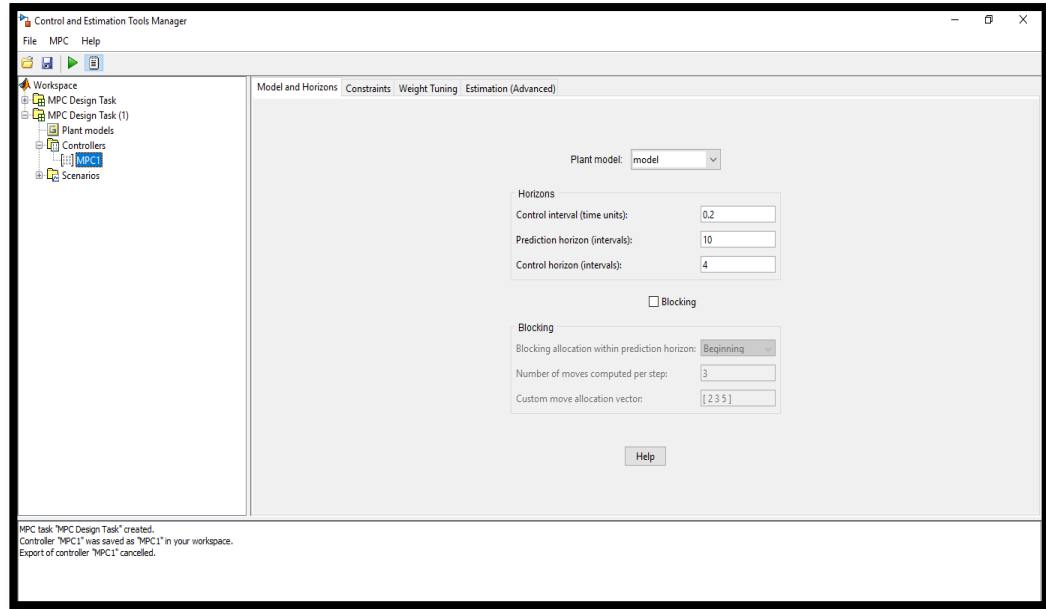


FIGURA 28: DISEÑO DEL CONTROLADOR MPC.

Una vez terminado el diseño de parámetros de predicción se procede a realizar pruebas de los diferentes escenarios y se observa el funcionamiento de la planta.

Para el presente trabajo se elige un tiempo de muestreo de 0.2s, un horizonte de predicción de 10 y un horizonte de control de 4.

En la Figura 30 y 31, se muestra la respuesta del sistema usando control MPC para uno de los escenarios que presenta mejor respuesta.

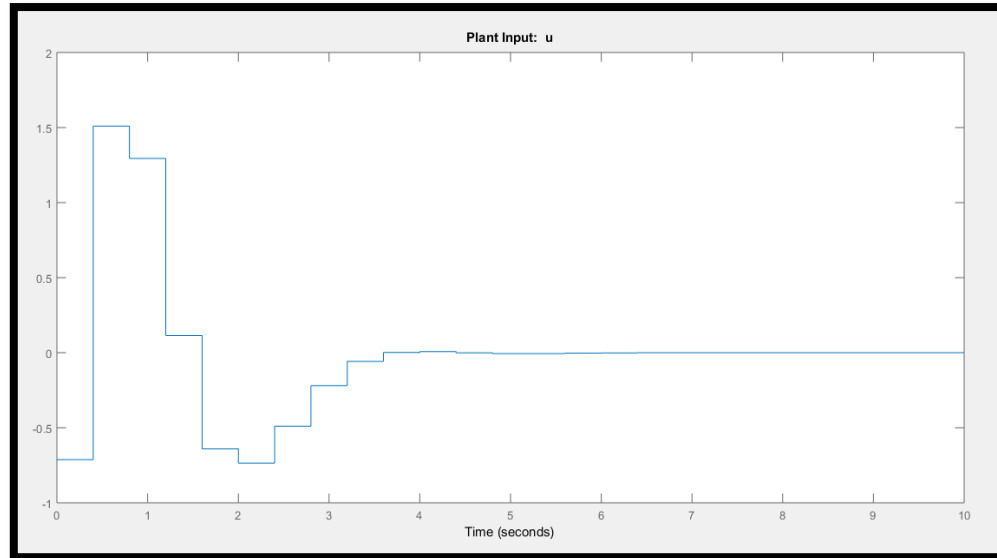


FIGURA 29: RESPUESTA DEL CONTROLADOR MPC, SEÑAL DE ENTRADA U

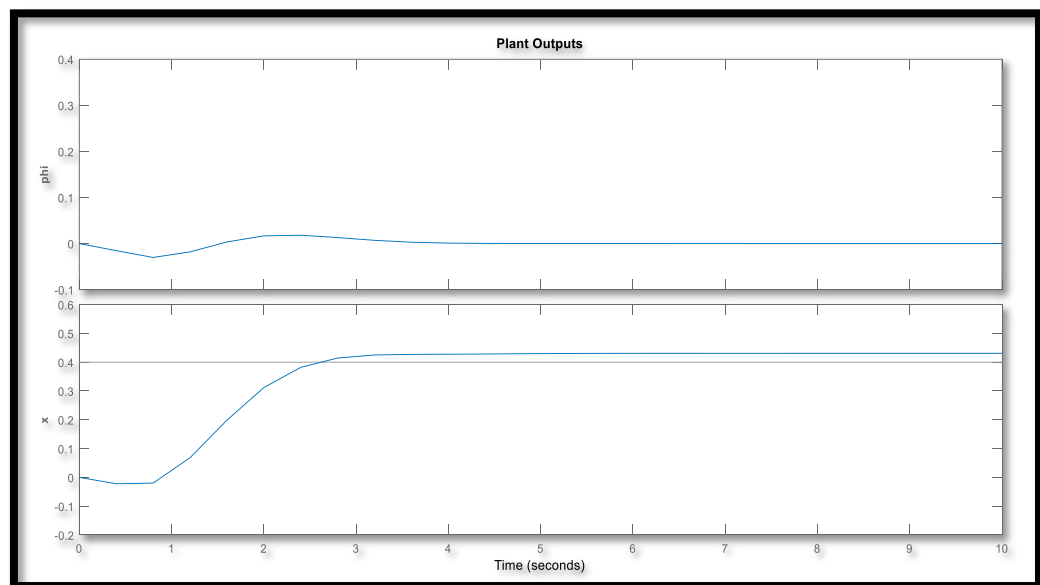


FIGURA 30: RESPUESTA DEL CONTROLADOR MPC, SEÑALES DE SALIDA

Una vez diseñado el control MPC, el diagrama en simulink del diseño del control MPC se muestra en la figura 32.

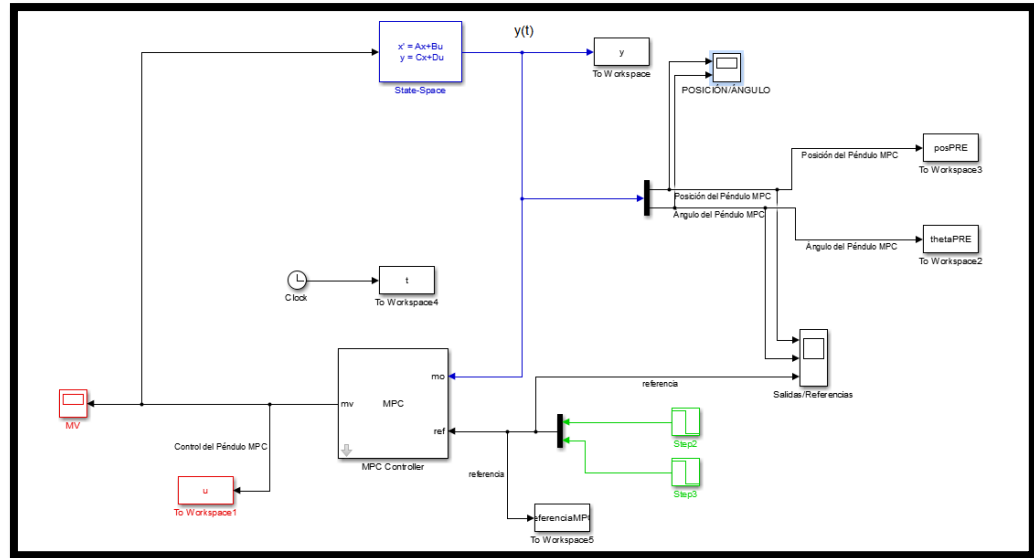


FIGURA 31: CONTROLADOR PREDICTIVO DEL SISTEMA DEL PÉNDULO INVERTIDO

4. RESULTADOS

4.1 Análisis de resultados de la simulación de controladores en Matlab

En este capítulo se presenta la operación y resultados obtenidos de cada controlador (PID, Variables de estado, MPC) y se realiza un análisis comparativo de los algoritmos de control con el fin de indicar cuál de ellos tiene un mejor desempeño según la respuesta del sistema y el tiempo de estabilización para cada alternativa de control.

4.2 Análisis comparativo de los algoritmos de control

4.2.1 Análisis comparativo de los algoritmos de control usando Variables de estado, PID y Predictivo

Los algoritmos de control PID, variables de estado y MPC están simuladas en tiempo discreto tiempo de muestreo 0.05 segundos. A continuación, se muestran los resultados obtenidos de las simulaciones en MATLAB para cada alternativa de control propuesta. Para conocer el desempeño de los controladores se realiza un análisis del tiempo de asentamiento, sobrepaso máximo.

4.2.2 Análisis comparativo de los algoritmos de control usando Variables de Estado, PID y Predictivo

De acuerdo al análisis del capítulo 4 se puede observar que los índices de desempeño del controlador de MPC tienen mejor desempeño presenta un sobrepaso máximo menor a los otros dos controladores y un tiempo de asentamiento menor a los controladores PID y MPC.

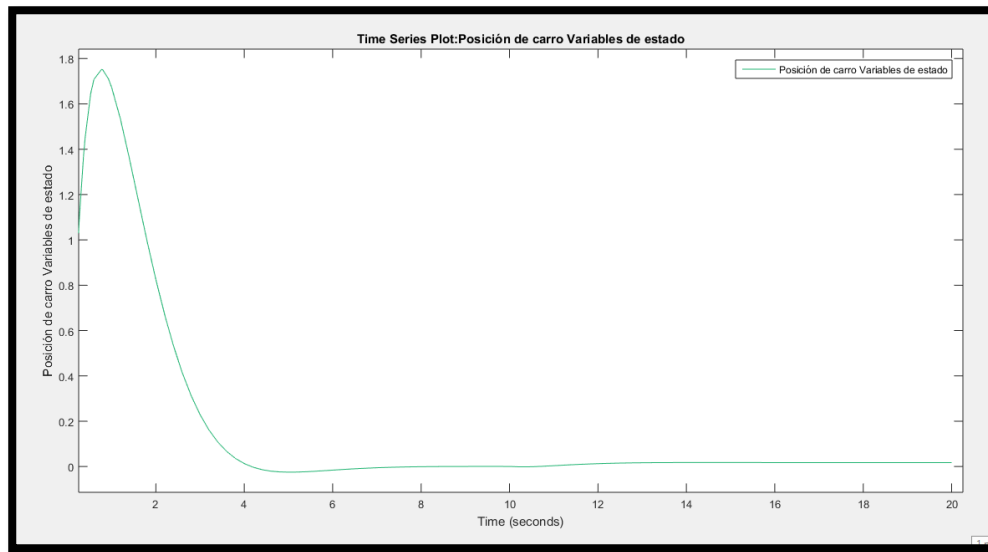


FIGURA 32: SIMULACIÓN DE MATLAB CORRESPONDIENTE A LA POSICIÓN DEL CARRO ANTE EL CONTROLADOR DE VARIABLES DE ESTADO

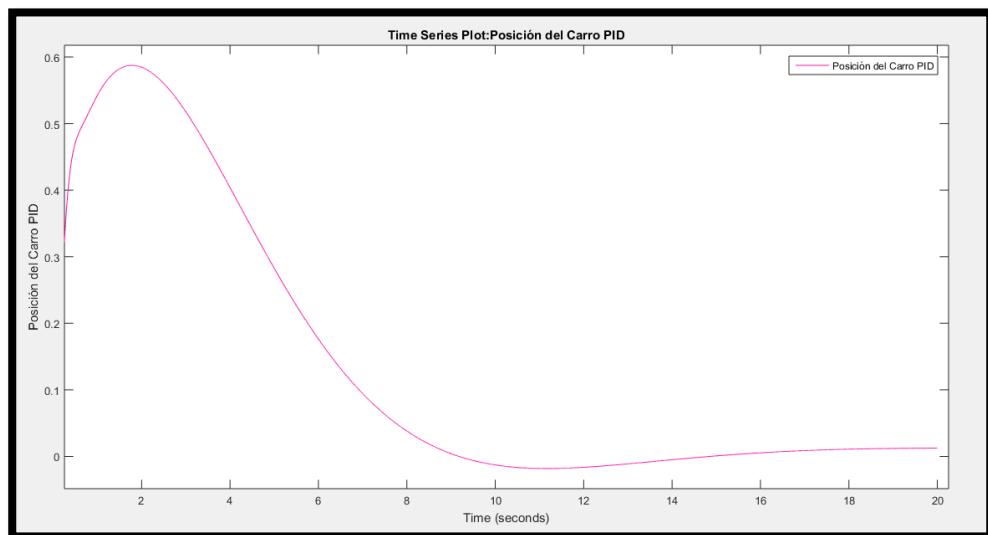


FIGURA 33: SIMULACIÓN DE MATLAB CORRESPONDIENTE A LA POSICIÓN DEL CARRO ANTE EL CONTROLADOR PID

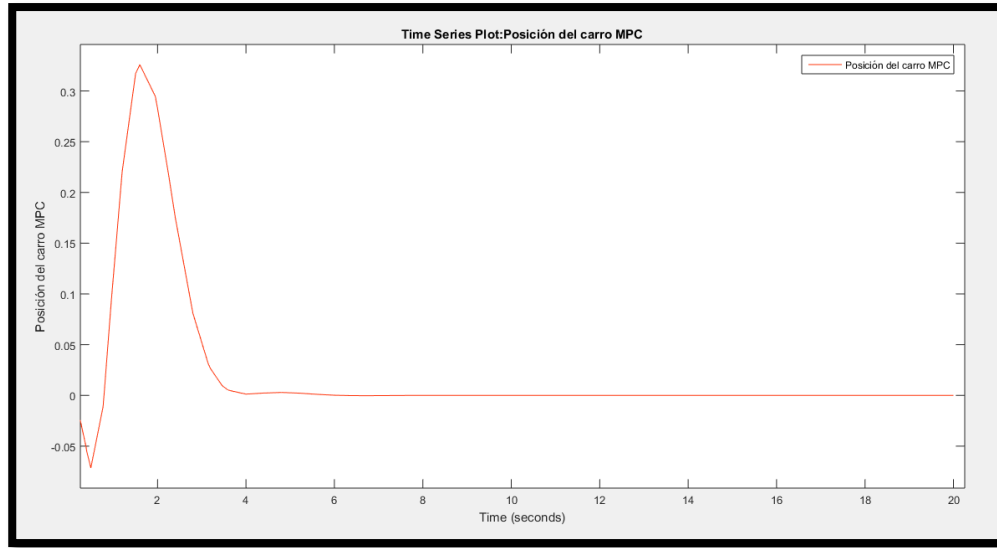


FIGURA 34: SIMULACIÓN DE MATLAB CORRESPONDIENTE A LA POSICIÓN DEL CARRO ANTE EL CONTROLADOR MPC.

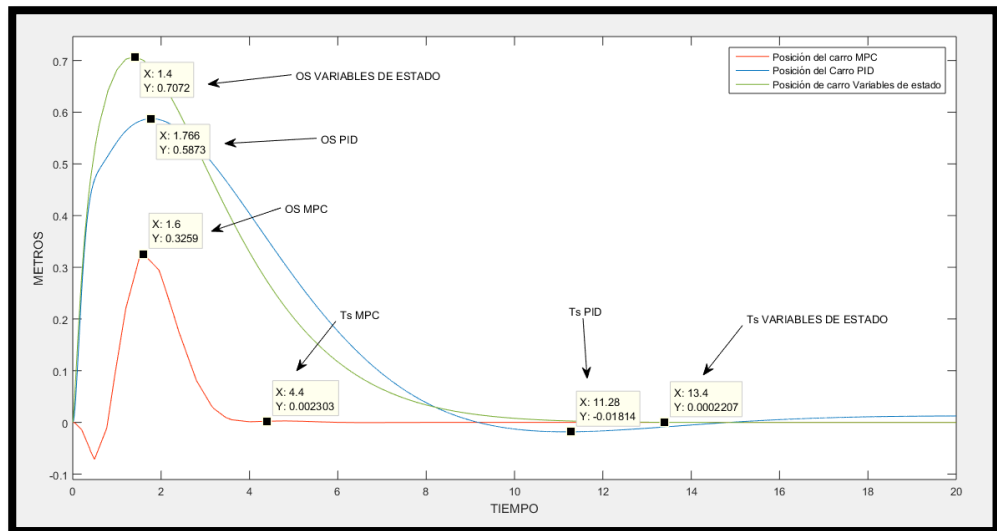


FIGURA 35: COMPARACIÓN DE ESTABILIZACIÓN DE POSICIÓN DEL CARRO ENTRE LOS CONTROLADORES DISEÑADOS.

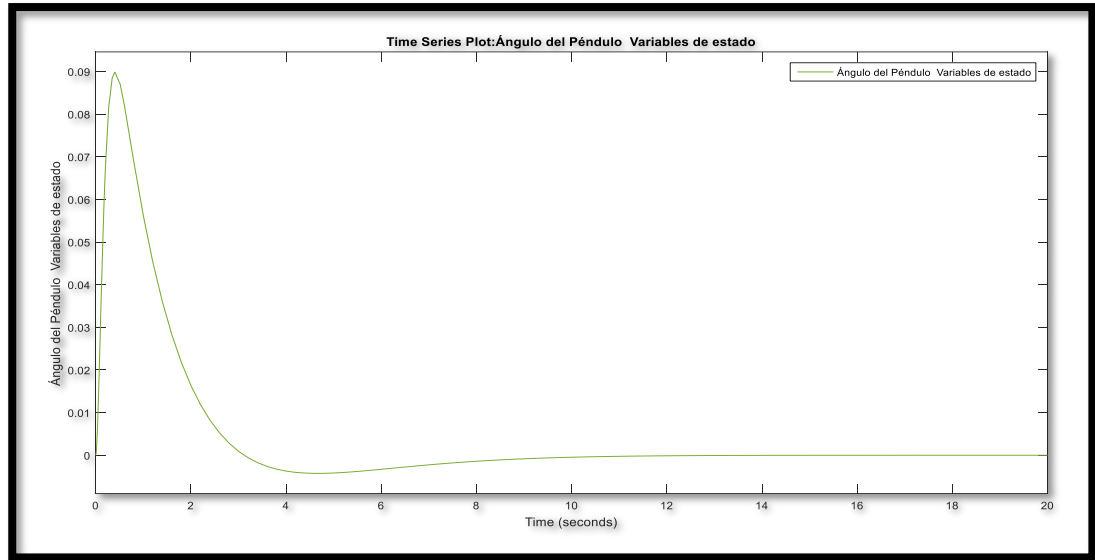


FIGURA 36: SIMULACIÓN DE MATLAB CORRESPONDIENTE AL ÁNGULO DEL PÉNDULO ANTE EL CONTROLADOR DE VARIABLES DE ESTADO

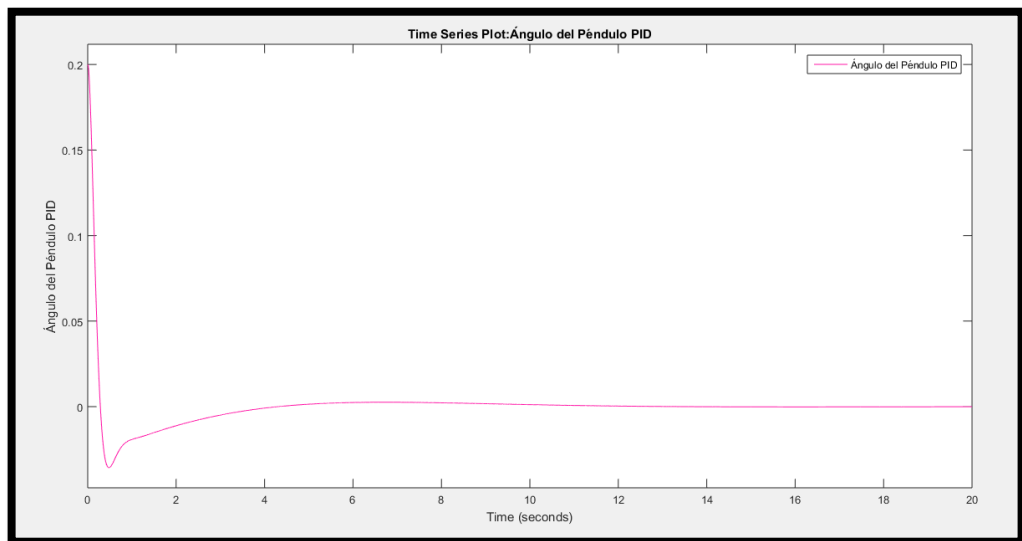


FIGURA 37: SIMULACIÓN DE MATLAB CORRESPONDIENTE AL ÁNGULO DEL PÉNDULO ANTE EL CONTROLADOR PID.

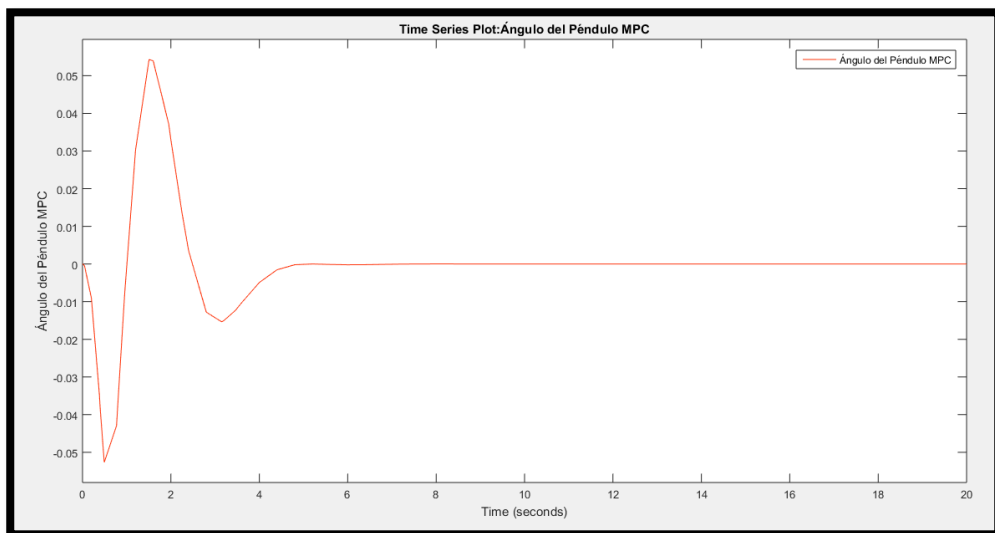


FIGURA 38: SIMULACIÓN DE MATLAB CORRESPONDIENTE AL ÁNGULO DEL PÉNDULO ANTE EL CONTROLADOR MPC.

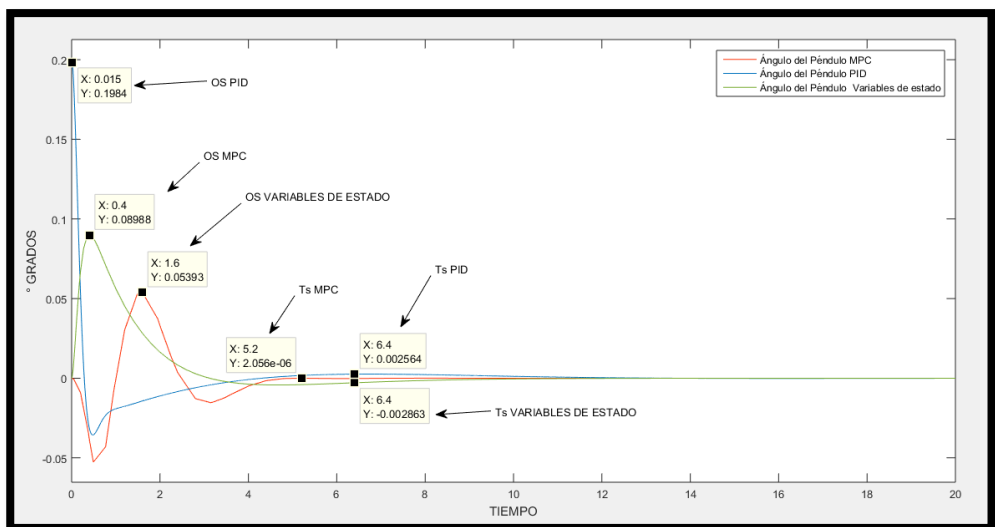


FIGURA 39: COMPARACIÓN DE ESTABILIZACIÓN DEL ANGULO DEL PENDULO ENTRE LOS CONTROLADORES DISEÑADOS.

En base a los resultados obtenidos en las FIGURAS 39 y 43 se procede con el análisis de resultados lo cual se muestra en las tablas 4 y 5.

TABLA 4 ÍNDICES DE DESEMPEÑO DE LOS CONTROLADORES PID, VARIABLES DE ESTADO Y MPC CORRESPONDIENTE A LA POSICIÓN DEL CARRO.

	Variables de estado		PID		MPC	
	Tiempo	Posición	Tiempo	Posición	Tiempo	Posición
%OS	1.4	0.7072	1.766	0.5873	1.6	0.3259
Tss	13.4	0	11.28	0	4.4	0

TABLA 5: ÍNDICES DE DESEMPEÑO DE LOS CONTROLADORES PID, VARIABLES DE ESTADO Y MPC CORRESPONDIENTE AL ÁNGULO DEL PÉNDULO

	Variables de estado		PID		MPC	
	Tiempo	Angulo	Tiempo	Angulo	Tiempo	Angulo
%OS	1.6	0.05393	0	0.1984	0.4	0.089
Tss	5	0	6.4	0	5.2	0

5. CONCLUSIONES

El péndulo invertido es una planta donde se pueden implementar y probar varios tipos de controladores y comparar ventajas del uno con el otro, teniendo eso en cuenta se presenta este proyecto con fines de establecer una base para futuros trabajos con sistemas no lineales. De acuerdo al análisis de los tres controladores se evidencia el comportamiento de cada uno de ellos y se concluye que los tres algoritmos son capaces de estabilizar al sistema. Sus características de respuesta tanto de sobrepaso máximo y tiempo de asentamiento se encuentran dentro de los parámetros aceptables para la operación de un controlador.

El desempeño de los tres algoritmos de control se compara en base al análisis de tiempo asentamiento, sobrepaso máximo. Y se concluye que el algoritmo con mejor rendimiento es el controlador MPC, su sobrepaso máximo es menor que el de PID y variables de estado y alcanza el estado estacionario en menor tiempo.

Con el presente trabajo se deja 10 prácticas de laboratorio con fines pedagógicos de poder enlazar y aplicar conceptos teóricos dictados en Teoría de control.

RECOMENDACIONES

Una de las ventajas de este proyecto es que su modelo es obtenido a partir del Péndulo Invertido 33-936s marca Feedback, por lo que, los algoritmos de control diseñados pueden ser usados en la planta real. Por lo tanto, se recomienda como trabajo futuro:

- Estudiar la factibilidad de aplicar otros algoritmos de control y comparar con lo obtenido en el presente trabajo.
- Usar el software Matlab para el diseño de controladores, por la facilidad que brinda al momento de implementar los mismos ya sea de manera simulada o en la planta real.
- Para obtener resultados óptimos en el diseño de un controlador, se debe tener un modelo matemático correcto que exprese el sistema,
- Cuando se realice la implementación en una planta real considerar ruidos externos y fricción,

REFERENCIAS

- [1] S. Khalil, "Inverted Pendulum: Analysis, design and implementation" in 2008 IEEE VVisionaries, Pakistan, pp. 9-12.
- [2] R. Dorf and R. Bishop, *Sistemas de control moderno*, Decima. Madrid, 2005.
- [3] Y. Liu, Z. Chen, D. Xue, "Real-time controlling of Inverted Pendulum by Fuzzy Logic " in 2009 IEEE International Conference on Automation and Logistics, pp. 1180-1183.
- [4] R. Balan, V. Maties, O. Hancu, S. Stan, "A Predictive Control Approach for the Inverse Pendulum on a Cart Problem" in 2005 IEEE International Conference on Mechatronics & Automation, pp. 2026-2031.
- [5] J Khan, K. Munawar, R. Azeem, M. Salman, "Inverted Pendulum with moving Reference for bernchmarking Control System Perfomance" in 2009 American Control Conference, pp. 3746-3768.
- [6] Feedback Instrumented Limited. "Digital Pendulum Control Experiments 33-936s", 2013
- [7] B, Kuo, *Sistemas de control Automático*, Séptima. Bogota, 2005.
- [8] “. d. a. e. I. Wireless sensor network, «[http://arcos.inf.uc3m.es/~sescolar/index_files/presentacion/red de sensores .pdf](http://arcos.inf.uc3m.es/~sescolar/index_files/presentacion/red_de_sensores .pdf),» 5 7 2014. [En línea]. Available: http://arcos.inf.uc3m.es/~sescolar/index_files/presentacion/red de sensores .pdf.
- [9] J. M. a. P. M. M. Koutny, «M. Koutny, J. Misurec, and P. Mlynek, “The new approaches of a multicasting in the wireless sensor networks,” in 33rd International Conference on TELECOMMUNICATIONS AND SIGNAL,» vol. 33, nº NN, p. NN, 2013.
- [10] “. R. f. P. T. Wes Iversen, «Wes Iversen, “WirelessHart Ready for Prime Time”,» NN, 6 8 2014. [En línea]. Available: Wes Iversen, “WirelessHart Ready for Prime Time”.
- [11] “. c. p. e. a. d. e. d. v. p. e. m. d. c. d. . G. -. 2012, «http://dspace.epn.edu.ec/bitstream/15000/8884/4/T11025_CAPITULO_4.pdf,» 6 7 2015. [En línea]. Available: http://dspace.epn.edu.ec/bitstream/15000/8884/4/T11025_CAPITULO_4.pdf.

- [12] L. Wang, E. Rogers, C. Buhr, "Predictive Repetitive Control: Design, Implementation and Applications Using MATLAB" in 2009 American Control Conference, pp. 1.
- [13] R. García and J. Ortiz, "Modelamiento y Simulación de un sistema con doble péndulo Invertido," 2015. [Online]. Available: <http://www.dspace.espol.edu.ec/xmlui/bitstream/handle/123456789/30823/D-84570.pdf?sequence=-1&isAllowed=y>. [Accessed: 22-Jul-2017].

ANEXOS

ANEXO 1 CÓDIGO DE LA SIMULACIÓN

```
clear all
close all
clc

% Parámetros del sistema

M=2.4;
m=0.23;
g=9.81;
l=0.36;
I=0.099;
b=0.05;
d=0.005;

%ecuaciones en variables de estado del sistema
% xdot - theta
z= [I*(M+m)+(M*m*l*l)];
y= (I+(m*l*l));
A=[ 0 1 0 0 ; 0 -(b*y)/z (m*m*l*l*g)/z 0; 0 0 0 1; 0 -((m*l*b)/z)
  ((M+m)*m*g*l)/z 0]
B=[0; (y)/(z); 0; m*l/z]
C=[1, 0, 0, 0; 0, 0, 1, 0]
D=[0;0]

% xdot - theta

s = tf('s');
states = {'x' 'x_dot' 'phi' 'phi_dot'};
inputs = {'u'};
outputs = {'x'; 'phi'};
%Representación en el espacio de estados

system = ss(A,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs)
```

```
%ecuaciones en variables de estado del sistema
```

```
%theta - xdot
```

```
z= [I*(M+m)+(M*m*l*l)];
```

```
y= (I+(m*l*l));
```

```
A=[ 0 1 0 0 ; ((M+m)*m*g*l)/z 0 0 -((m*l*b)/z); 0 0 0 1; (m*m*l*l*g)/z  
0 0 -(b*y)/z]
```

```
B=[0; m*l/z; 0; (y)/(z)]
```

```
C=[1, 0, 0, 0; 0, 0, 1, 0]
```

```
D=[0;0]
```

```
%Representación de variables de estado
```

```
% theta - xdot
```

```
%s = tf('s');
```

```
%states = {'phi' 'phi_dot' 'x' 'x_dot'};
```

```
%inputs = {'u'};
```

```
%outputs = {'phi'; 'x'};
```

```
%Respuesta a escalón
```

```
%T = 0:0.01:1;
```

```
%U = 0.2*ones(size(T));
```

```
%[Y,X] = lsim(A,B,C,D,U,T);
```

```
%plot(T,Y)
```

```
%axis([0 2 0 100])
```

```
% Control predictivo
```

```
Ts = 0.2; % sampling time
```

```
model = c2d(system,Ts);
```

```
% Obtención de la función de Transferencia
```

```

system_tf = tf(system)

%Respuesta a Impulso
T = 0:0.01:1;
impulse(system,T);
title('Respuesta del Sistema en Lazo Abierto a entrada tipo Impulso')
grid on

%Respuesta a tipo Paso
T = 0:0.001:6;
step(system,T);
title('Respuesta del Sistema en Lazo Abierto a entrada tipo Paso')
grid on

% Polinomio caracteristico
% Find Tranformation Matrix W
% Characteristic equation:  $s^4 + a_4s^3 + a_3s^2 + a_2s + a_1$ 
% x-theta
% inverted pendulum characteristic equation:  $s^4 + 0.0194s^3 - 6.4363s^2 - 0.01224s$ 
a = poly(A)
% Companion matrices (Ac, Bc)
Ac = [ 0 1 0 0;
       0 0 1 0;
       0 0 0 1;
       -a(5) -a(4) -a(3) -a(2)]

%
Bc = [0; 0; 0; 1]

% Controllability
T = ctrb(A,B)
% Controllability of companion matrices
Tc = ctrb(Ac,Bc)
%
% Transformation matrices
W = T*inv(Tc)

```

```

% Obtención de polos

%P=eig(A)
pzmap(system)

syms s
I=eye(4,4);
det(s*I-A);

% Polos deseados

%Polos_deseados =[0.9501+0.0464i, 0.9501-0.0464i, 0.7788, 0.7788];

%Diseño de los polos en lazo cerrado
%Regulación de estados
ts= 5; % Tiempo de establecimiento
mp= 5/100; % Máximo sobre impulso
zeta = abs(log(mp))/sqrt(pi^2+log(mp)^2)
Wn = 4/(zeta*ts);
Wd=Wn*sqrt(1-zeta^2)
phi= zeta*Wn
P=roots(conv([1 2*zeta*Wn Wn^2],[1 alpha*Wn]));
%K = acker(A,B,P)

% determinar estabilidad

Polos = eig(A)
coa = 0;
for co = 1:max(size(Polos))
    if Polos(co)>=0
        disp('El sistema es Inestable.')
        coa = 1;
        break
    end
end

```

```

        end
    end
    if coa==0
        disp('El sistema es Estable.')
    end

    % controlabilidad

    Cont = [B A*B (A^2)*B (A^3)*B];
    if rank(Cont)==4
        disp('El sistema es Controlable.')
    else
        disp('El sistema no es Controlable.')
    end

    %observabilidad
    Obs=[ C
          C*A
          C*(A^2)
          C*(A^3)];
    if rank(Obs)==4
        disp('El sistema es Observables.')
    else
        disp('El sistema no es Observable.')
    end

    % entrada (definimos la entrada como un impulso unitario,
    % las condiciones iniciales y los tiempos de simulación desde cero hasta
    % dos segundos
    % tomados con una diferencial de tiempo 0.001s lo suficiente pequeña para
    % q el sistema
    % se comporte como si tratase de un sistema de tiempo continuo)

    u=1 % Impulso Unitario
    % Definición de tiempos de simulación
    ti=0;
    tf=80;

```

```

dt=0.005;
% Condiciones Iniciales
x = [2 -0.5 0 0]'; %Asumimos que el péndulo está inclinado 2°

%Simulación
k = 1;
for t = ti:dt:tf
    X1(k,1)=x(1,1); % Ángulo(Theta)
    X2(k,1)=x(2,1); % Velocidad angular (Theta Punto)
    X3(k,1)=x(3,1); % Posición del carro (Equis)
    X4(k,1)=x(4,1); % velocidad del carro (Equis)
    TT(k,1)=t;

    % Ecuaciones de Estado
    xp=A*x + B*u;
    y=C*x + D*u;
    Y1(k,1)=y(1,1); % X1
    Y2(k,1)=y(2,1); %X3

    % Integrando mediante Euler
    x=x + dt*xp;
    k=k + 1;
    u=0;
end

%Graficamos las Salidas
figure(1)
subplot(221)
plot(TT,X1,'b')
title('Ángulo del péndulo')
xlabel('Tiempo(segundos) ')
ylabel('°')
subplot(222)
plot(TT,X2,'b')
title('Velocidad Angular del Péndulo')
xlabel('Tiempo(Segundos) ')
ylabel('°/s')

```

```

subplot (223)
plot(TT,X3,'b')
title('Estado X de la Posición del carro')
xlabel('Tiempo (Segundos)')
ylabel('Metros')
subplot(224)
plot(TT,X4,'b')
title('Estado X de la velocidad del Carro')
xlabel('Tiempo (Segundos)')
ylabel('Metros/Segundo')

figure(2)
subplot(221)
plot(TT,Y1)
title('Salida Y1 de la Ángulo del Péndulo')
xlabel('Tiempo (Segundos)')
ylabel('°')
subplot(222)
plot(TT,Y2)
title('Salida Y2 de la Posición del Carro.')
xlabel('Tiempo (Segundos)')
ylabel('Metros')
subplot(212)
plot(TT,Y1,'r-',TT,Y2,'b-')
title('Salidas:Ángulo del Péndulo y Posición del carro.')
xlabel('Tiempo (Segundos)'); legend('Ángulo','Posición')

%Diseño del controlador.
%Polinomio Característico.

p1=[1 -Polos(1)];
p2=[1 -Polos(2)];
p3=[1 -Polos(3)];
p4=[1 -Polos(4)];
apc=conv(p1,conv(p2,conv(p3,p4)));

```

```

% Matriz de Transformación T.
%T = Cont*W
W = [ apc(4) apc(3) apc(2) 1
      apc(3) apc(2) 1 0
      apc(2) 1 0 0
      1 0 0 0];
T = Cont*W;

%Introducimos los polos deseados
plot(Po, 'o')
%Polos_d=[0; -0.019; -0.8+0.724i; -0.8-0.724i]
Polos_d=[-7; -5; -0.8+0.724i; -0.8-0.724i]
%Polos_d=[0; -0.019; -0.8+0.724i; -0.8-0.724i]
%Polos_d = [-1
             -1
             -0.801+1.61i
             -0.801-1.61i]
Polos_d = [-2
             -1
             -0.8390+0.8i
             -0.8390-0.8i]
pd1 = [1 -Polos_d(1) ]
pd2 = [1 -Polos_d(2) ]
pd3 = [1 -Polos_d(3) ]
pd4 = [1 -Polos_d(4) ]
adpc = conv(pd1, conv(pd2, conv(pd3, pd4)))
%K = [ adpc(5)-apc(5) adpc(4)-apc(4) adpc(3)-apc(3) adpc(2)-
apc(2)]*inv(T)
K= acker(A,B,Polos_d)
N_Polos = eig(A-B*K)

% entrada (Simulando el lazo de control al aplicar la Ley de control
% mediante k

u=1 % Impulso Unitario
% Definición de tiempos de simulación

```

```

%Simulación
k = 1;
for t = ti:dt:tf
    X1(k,1)=x(1,1); % Ángulo(Theta)
    X2(k,1)=x(2,1); % Velocidad angular (Theta Punto)
    X3(k,1)=x(3,1); % Posición del carro (Equis)
    X4(k,1)=x(4,1); % velocidad del carro (Equis)
    TT(k,1)=t;

    % Ecuaciones de Estado
    xp=A*x + B*u;
    y=C*x + D*u;
    u = -K*x; % Ley de Control
    Y1(k,1)=y(1,1); % X1
    Y2(k,1)=y(2,1); %X3

    % Integrando mediante Euler
    x=x + dt*xp;
    k=k + 1;
end

%Graficamos las Salidas con el controlador
figure(1)
subplot(221)
plot(TT,X1,'b')
title('Posición del carro')
xlabel('Tiempo(segundos)')
ylabel('Metros')
subplot(222)
plot(TT,X2,'b')
title('Velocidad del carro')
xlabel('Tiempo(Segundos)')
ylabel('Metros/Segundo')
subplot(223)
plot(TT,X3,'b')
title('Ángulo del carro')

```

```

xlabel('Tiempo (Segundos)')
ylabel('°')
subplot(224)
plot(TT,X4,'b')
title('velocidad angular del Carro')
xlabel('Tiempo (Segundos)')
ylabel('°/s')

figure(2)
subplot(221)
plot(TT,Y1)
title('Salida Y1 de la Posición del carro')
xlabel('Tiempo (Segundos)')
ylabel('Metros')
subplot(222)
plot(TT,Y2)
title('Salida Y2 de la Ángulo del Péndulo.')
xlabel('Tiempo (Segundos)')
ylabel('°')
subplot(212)
plot(TT,Y1,'r-',TT,Y2,'b-')
title('Salidas:Posición del carro y Ángulo del Péndulo y.')
xlabel('Tiempo (Segundos)'); legend('Posición','Ángulo')

```

PRÁCTICAS DE LABORATORIO

ANEXO 2 PRÁCTICA#1:

“Linealizar Modelado Matemático del sistema dinámico del Péndulo Invertido en variables de estado”

1.1.1. Objetivos.

Objetivo General:

- Realizar el modelado matemático del sistema Péndulo Invertido y linealizar el sistema.

Objetivos específicos:

- Obtener la representación del modelo en términos de función de transferencia y variables de estado.
- Usar los códigos de Matlab correspondientes a variables de estado.
- Analizar la estabilidad, controlabilidad, observabilidad del sistema.
- Comprender las ecuaciones de estado su funcionamiento e implementación para poder realizar un correcto análisis del modelo.

1.1.2. Recursos utilizados.

- Sistema dinámico del Péndulo Digital modelo 33-936s marca Feedback
- Matlab/Simulink.

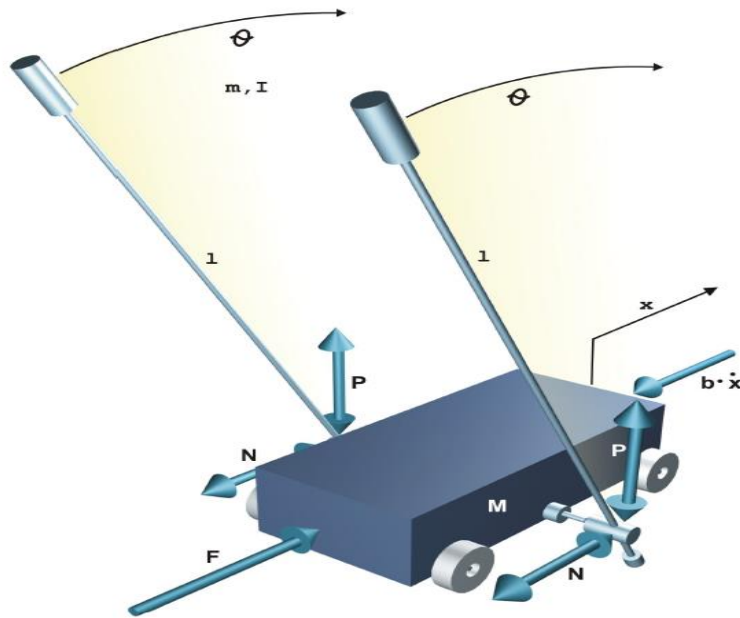
1.1.3. Tiempo estimado.

El tiempo estimado para la realización de esta práctica es de 2 horas.

1.1.4. Desarrollo.

CÁLCULOS TEÓRICOS

1. El primer paso importante es entender el comportamiento dinámico de un sistema para conocer el funcionamiento de un proceso-planta. Es necesario tener las ecuaciones que describen el proceso correctas teniendo en cuenta las leyes físicas.



2. La dinámica del Péndulo Invertido es un sistema electro-mecánico y se describe en términos de ecuaciones diferenciales.

$$U = (M + m)\ddot{x} + b \cdot \dot{x} - m \cdot l \cdot \ddot{\theta} \cdot \cos\theta + m \cdot l \cdot \dot{\theta}^2 \cdot \sin\theta$$

$$0 = (I + ml^2)\ddot{\theta} - m \cdot l \cdot \ddot{x} \cdot \cos\theta - m \cdot g \cdot l \cdot \sin\theta + D \cdot \dot{\theta}$$

El sistema del péndulo invertido tiene los siguientes parámetros:

Símbolo	Parámetro	Valor
g	Gravedad	9.81 m/s ²
l	Longitud de la varilla	0.36 a 0.4 m
M	Masa del carro	2.4kg
m	Masa del péndulo	0.23kg
I	Momento de inercia	0.099 Kg.m ²

b	Coeficiente de fricción	0.05 Ns/m
d	Coeficiente de caída (despreciable)	0

3. Linealización del Modelo. Para analizar la dinámica del sistema se linealiza las ecuaciones matemáticas antes obtenidas. En el caso del péndulo invertido se tiene dos puntos de equilibrio cuando $\theta = 0$ (péndulo invertido) y cuando $\theta = \pi$ (control de grúa).

$$\sin\theta = \theta$$

$$\cos\theta = 1$$

$$\dot{\theta}^2 = 0$$

Reemplazando se obtiene el modelo lineal:

$$U = (M + m)\ddot{x} + b.\dot{x} + m.l.\ddot{\theta}$$

$$0 = (I + ml^2)\ddot{\theta} + m.l.\ddot{x} - m.g.l\theta + D\dot{\theta}$$

Las ecuaciones en variables de estado de un sistema están definidas mediante las ecuaciones siguientes:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \text{ ecuación de estado}$$

$$y(t) = C(t)x(t) + D(t)u(t) \text{ ecuación de salida}$$

Con el conocimiento de las propiedades físicas del péndulo invertido se escoge las dos variables de estado como sigue:

$$x_1 = \theta;$$

$$x_2 = \dot{\theta} = \dot{x}_1$$

$$x_3 = x$$

$$x_4 = \dot{x} = \dot{x}_3$$

Usando las ecuaciones del Modelo Matemático Linealizado se obtiene el sistema de ecuaciones en variables de estado.

$$U = (M + m)\ddot{x} + b.\dot{x} - m.l.\ddot{\theta} \quad (1)$$

$$0 = (I + ml^2)\ddot{\theta} - m.l.\ddot{x} - m.g.l\theta + D\dot{\theta} \quad (2)$$

Reemplazando $\dot{x}_4$ a las ecuaciones 1 y 2

$$\begin{aligned} \dot{x}_2 = & \frac{mgl(M + m)}{I(M + m) + Mml^2}x_1 + \frac{-D(M + m)}{I(M + m) + Mml^2}x_2 + \frac{-mlb}{I(M + m) + Mml^2}x_4 \\ & + \frac{ml}{I(M + m) + Mml^2}F \end{aligned}$$

$$\begin{aligned} \dot{x}_4 = & \frac{gm^2l^2}{I(M + m) + Mml^2}x_1 + \frac{-Dml}{I(M + m) + Mml^2}x_2 + \frac{-b(I + ml^2)}{I(m + m) + Mml^2}x_3 \\ & + \frac{(I + ml^2)}{I(M + m) + Mml^2}F \end{aligned}$$

Por tanto, queda definida la entrada y las salidas del sistema de la siguiente forma:

Entrada:

$$U = F(t)$$

Salidas:

$$y_1(t) = \theta(t)$$

$$y_2(t) = X(t)$$

Construyendo las matrices en Variables de Estado:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

$$y(t) = C(t)x(t)$$

$$\dot{x} = \begin{bmatrix} \dot{\theta} \\ \dot{\theta} \\ \dot{x} \\ \ddot{x} \end{bmatrix}; \quad U = F(t); \quad Y = \begin{bmatrix} \theta(t) \\ x(t) \end{bmatrix}$$

$$; A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{mgl(M+m)}{I(M+m)+Mml^2} & \frac{-D(M+m)}{I(M+m)+Mml^2} & 0 & \frac{-mlb}{I(M+m)+Mml^2} \\ 0 & 0 & 0 & 1 \\ \frac{gm^2l^2}{I(m+m)+Mml^2} & \frac{-mlD}{I(M+m)+Mml^2} & 0 & \frac{-b(I+Ml^2)}{I(m+m)+Mml^2} \end{bmatrix} B = \begin{bmatrix} 0 \\ ml \\ 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \end{bmatrix}; C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

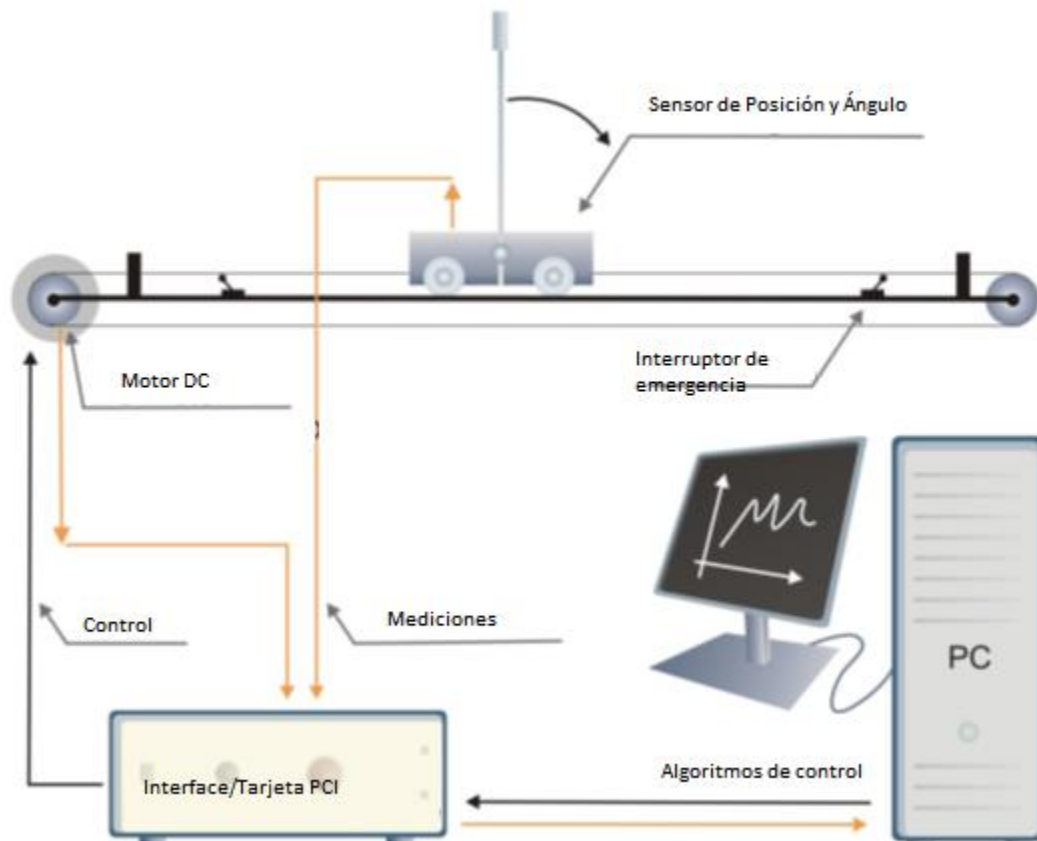
Reemplazando con los parámetros del péndulo:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 6.4363 & -0.0396 & 0 & -0.0125 \\ 0 & 0 & 0 & 1 \\ 0.2026 & -0.012 & 0 & -0.0194 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 0.2495 \\ 0 \\ 0.3881 \end{bmatrix}; C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

SIMULACIÓN EN SIMULINK

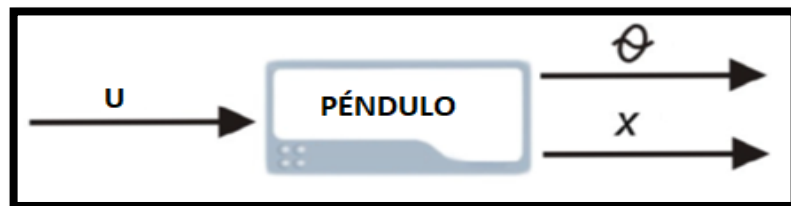
1. Identificar la variable que se va a controlar (¿Que deseo controlar?) según la figura 39 que describe el sistema de control del Péndulo
2. Realizar un diagrama de bloques donde se pueda observar todos los componentes del sistema de control del Péndulo Invertido y justifique su diagrama a desarrollar.
3. Utilizando la representación en variables de estado hacer un análisis de estabilidad, controlabilidad, observabilidad del sistema del Péndulo Invertido en Matlab
4. Realizar la representación de función de transferencia

El sistema de control del Péndulo Invertido se detalla a continuación:

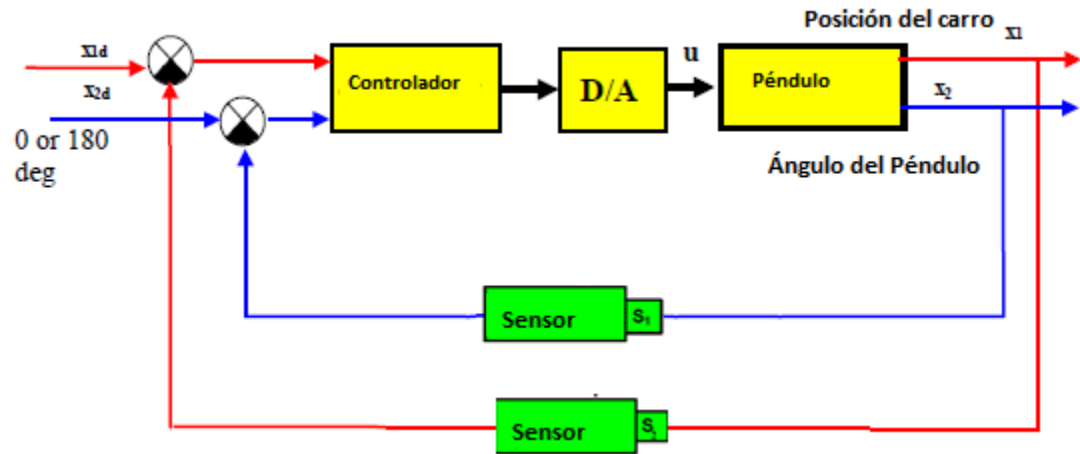


EXPERIMENTACIÓN

1. Observando el sistema de control tenemos 3 variables (señal de control, posición de carro y ángulo del péndulo). Tenemos que la variable manipulada es la señal de control y las variables medidas son posición y ángulo. Lo que se va a controlar con el fin de estabilizar la planta son las variables medidas mediante el manipulamiento de la señal de control.



2. El diagrama de bloque del sistema del Péndulo Invertido se lo muestra en la gráfica a continuación.



3. Simulación en representación de las variables estado en Matlab se muestra a continuación:

1. % Parámetros del sistema

2.

3. M=2.4;

4. m=0.23;

5. g=9.81;

6. l=0.36;

7. I=0.099;

8. b=0.05;

9. d=0.005;

10.

11. %ecuaciones en variables de estado del sistema

12. %theta - xdot

13. z= [I*(M+m)+(M*m*l)];

14. y= (I+(m*l));

15. A=[0 1 0 0 ; ((M+m)*m*g*l)/z -(M+m)*d/z 0 -(m*l*b)/z; 0 0 0 1; (m*m*l*I*g)/z -(d*m*l)/z
0 -(b*y)/z]

16. B=[0; m*l/z; 0; (y)/(z)]

17. C=[1, 0, 0, 0; 0, 0, 1, 0]

18. D=[0;0]

19.

```

20. %Representación de variables
21.
22. %theta - xdot
23. s = tf('s');
24. states = {'phi' 'phi_dot' 'x' 'x_dot'};
25. inputs = {'u'};
26. outputs = {'phi'; 'x'};
27.
28. system = ss(A,B,C,D,'statename',states, 'inputname',inputs,'outputname',outputs)

```

```

1.  % determinar estabilidad
2.
3.  Polos = eig(A)
4.  coa = 0;
5.  for co = 1:max(size(Polos))
6.      if Polos(co)>=0
7.          disp('El sistema es Inestable.')
8.          coa = 1;
9.          break
10.     end
11. end
12. if coa==0
13.     disp('El sistema es Estable.')
14. end
15.
16. % controlabilidad
17.
18. Cont = [B A*B (A^2*B (A^3)*B];
19. if rank(Cont)==4
20.     disp('El sistema es Controlable.')
21. else
22.     disp('El sistema no es Controlable.')
23. end
24.
25. %observabilidad
26. Obs=[ C
27.     C*A

```

28. $C*(A^2)$
29. $C*(A^3)];$
30. if rank(Obs)==4
31. disp('El sistema es Observables.')
32. else
33. disp('El sistema no es Observable.')
34. End

La representación en función de transferencia es la siguiente:

1. % Obtención de de la funcion de Transferencia
2. system_tf = tf(system)

Respuesta de Matlab:

```

From input "u" to output...
      0.2495 s - 5.193e-18
phi:  -----
      s^3 + 0.05902 s^2 - 6.436 s - 0.1224

      0.3881 s^2 + 0.01506 s - 2.447
x:  -----
      s^4 + 0.05902 s^3 - 6.436 s^2 - 0.1224 s

Continuous-time transfer function.

>> system_tf = tf(system)

system_tf =

From input "u" to output...
      0.2495 s - 5.193e-18
phi:  -----
      s^3 + 0.05902 s^2 - 6.436 s - 0.1224

      0.3881 s^2 + 0.01506 s - 2.447
x:  -----
      s^4 + 0.05902 s^3 - 6.436 s^2 - 0.1224 s

Continuous-time transfer function.

```

1.1.5. Conclusiones de la práctica.

- Se obtiene el modelo matemático del sistema linealizado.
- Identifica las variables de la planta.
- Explica el funcionamiento del sistema mediante un diagrama de bloques.
- Teniendo en cuenta el funcionamiento de la planta se analiza estabilidad, controlabilidad y observabilidad sistema
- Se obtiene mediante Matlab la representación de función de transferencia

ANEXO 3 PRÁCTICA#2:

“Simulación y Análisis del sistema en lazo abierto en el espacio de estados”

1.3.1. Objetivos.

Objetivo General:

- Realizar un análisis en lazo abierto del sistema del Péndulo Invertido utilizando las herramientas de MATLAB/SIMULINK.

Objetivos específicos:

- Diseñar en Simulink el sistema del Péndulo Invertido en lazo abierto.
- Analizar repuestas al sistema
- Representar mediante gráficas los resultados.

1.3.2. Recursos utilizados.

- Matlab/Simulink
- Modelado Matemático del Péndulo Invertido

1.3.3. Tiempo estimado.

El tiempo estimado para la realización de esta práctica es de 2 horas.

1.3.4. Desarrollo.

SIMULACIÓN EN SIMULINK

1. Diseñar en Simulink el sistema del Péndulo Invertido en el espacio de estado
2. Analizar respuestas al sistema (Tipo paso, Rampa)
3. Representar mediante gráficos los resultados

EXPERIMENTACIÓN

1. Para la simulación de las variables de la Planta del Péndulo Invertido en lazo abierto en el espacio de estados se evaluarán en el archivo Pract2_Lazo_Abierto.

Tipo Paso

La prueba tipo paso va a hacer aplicada en 1 s.

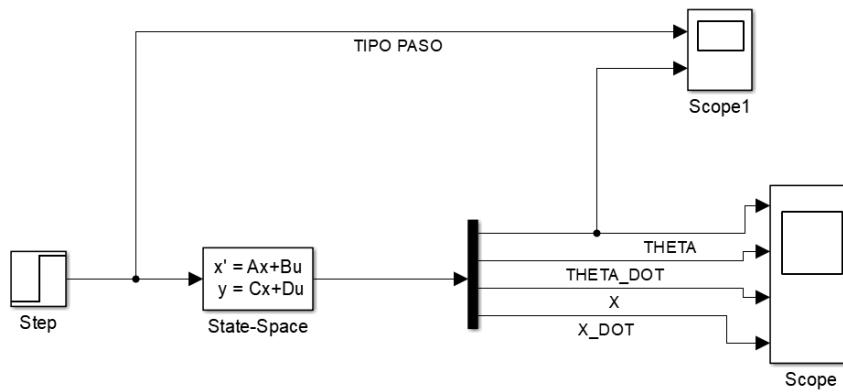
Parámetros de Simulación

Hora inicio:0

Finalización:1.5

Tipo de Algoritmo: Variable-Step (ODE45)

Máximo tamaño de la señal Tipo Paso: 0.05



Tipo Rampa

La prueba tipo paso va a hacer aplicada en 1 s.

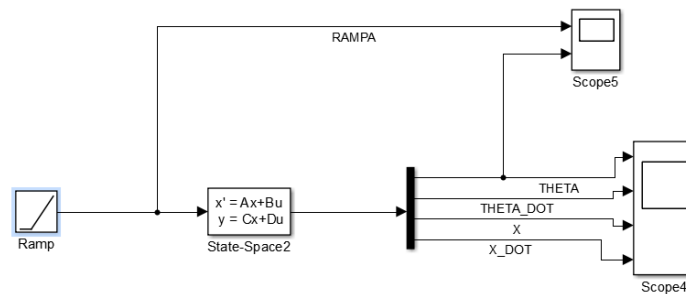
Parámetros de Simulación

Hora inicio:0

Finalización:1.5

Tipo de Algoritmo: Variable-Step (ODE45)

Máximo tamaño de la señal Tipo Paso: 0.5



1.3.5. Conclusiones de la práctica.

- El diseño del sistema en lazo abierto muestra similares respuestas a las entradas, tipo paso y rampa.
- La simulación del Péndulo Invertido es realizada en variables de estado, con el bloque “State Space” de simulink
- Las respuestas del sistema del Péndulo Invertido están simuladas mediante el bloque “scope” de Simulink

ANEXO 4: PRÁCTICA#3:

“Simulación y Análisis del sistema en lazo cerrado en el espacio de estados”

1.4.1. Objetivos.

Objetivo General:

- Familiarizar al estudiante con las herramientas de análisis utilizando MATLAB/SIMULINK del sistema del Péndulo Invertido en lazo cerrado

Objetivos específicos:

- Habilidad de diseñar en Simulink el sistema del Péndulo Invertido en lazo cerrado.
- Analizar repuestas al sistema
- Representar mediante gráficas los resultados.

1.4.2. Recursos utilizados.

- Matlab/Simulink
- Modelado Matemático del Péndulo Invertido

1.4.3. Tiempo estimado.

El tiempo estimado para la realización de esta práctica es de 2 horas.

1.4.4. Desarrollo.

SIMULACIÓN EN SIMULINK

1. Diseñar en Simulink el sistema del Péndulo Invertido en el espacio de estado
2. Analizar respuestas al sistema (Tipo paso y Rampa)
3. Representar mediante gráficos los resultados

EXPERIMENTACIÓN

1. Para la simulación de las variables de la Planta del Péndulo Invertido en lazo abierto en el espacio de estados se evaluarán en el archivo Pract2_Lazo_Cerrado.

Tipo Paso

La prueba tipo paso va a hacer aplicada en 1 s.

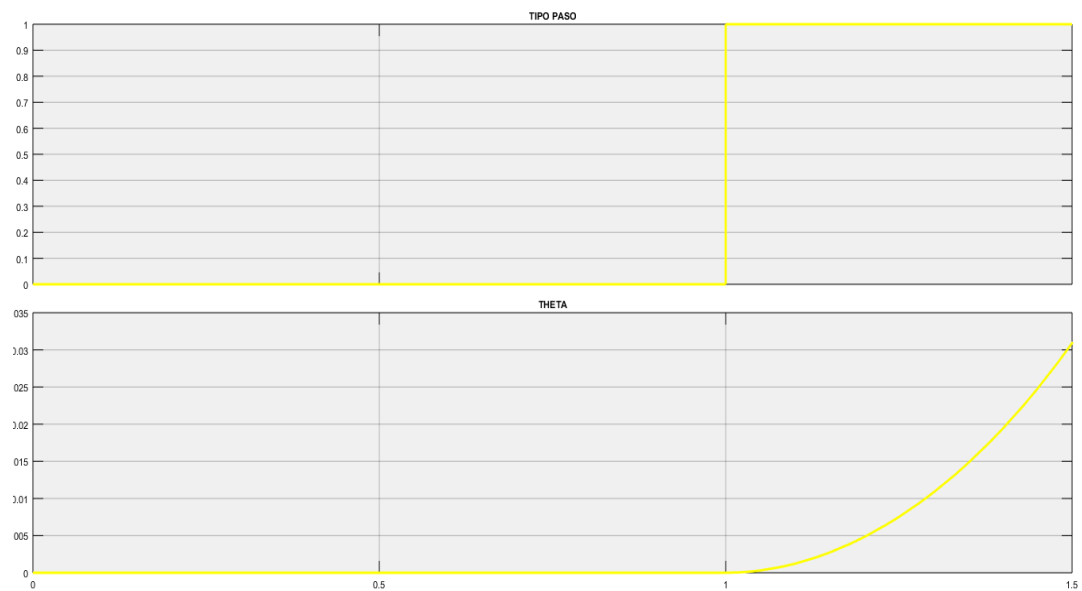
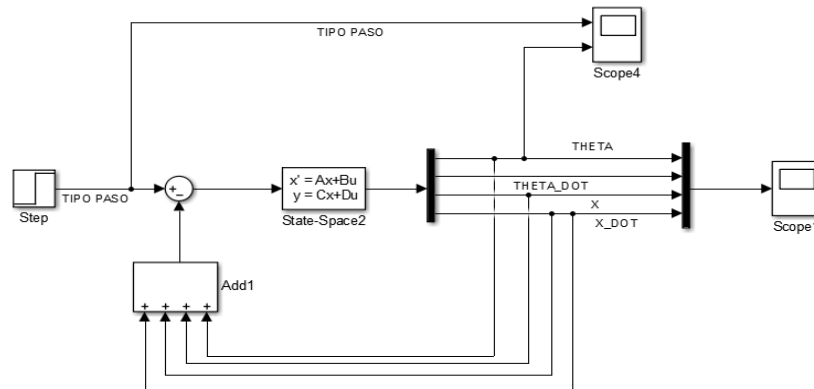
Parámetros de Simulación

Hora inicio:0

Finalización:1.5

Tipo de Algoritmo: Variable-Step (ODE45)

Máximo tamaño de la señal Tipo Paso: 0.05



Tipo Rampa

La prueba tipo paso va a hacer aplicada en 1 s.

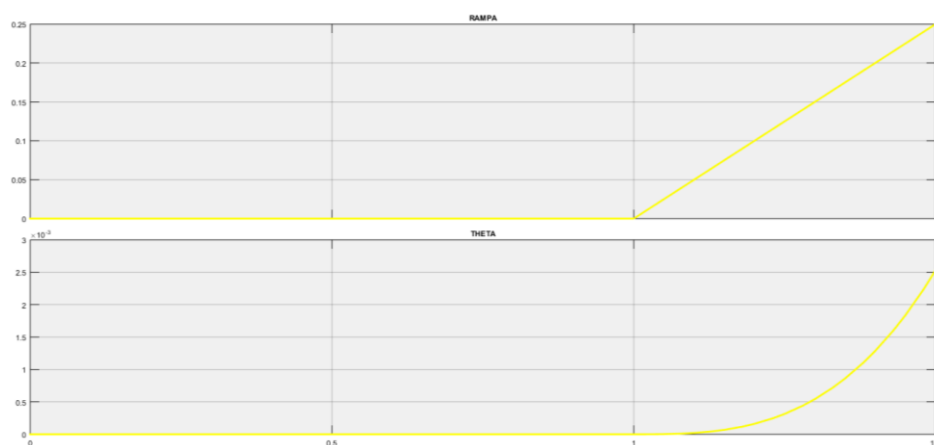
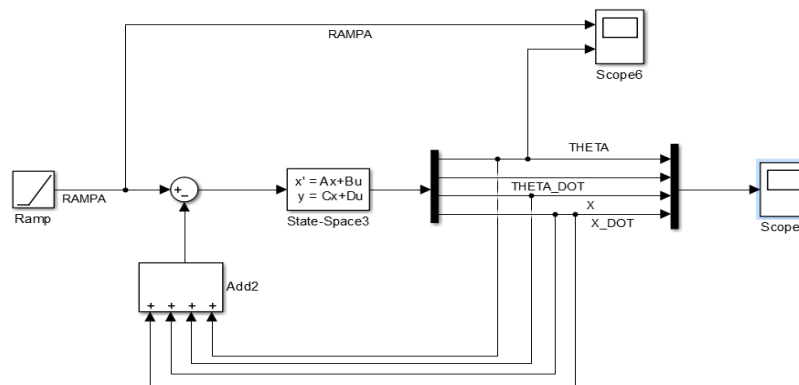
Parámetros de Simulación

Hora inicio:0

Finalización:1.5

Tipo de Algoritmo: Variable-Step (ODE45)

Máximo tamaño de la señal Tipo Paso: 0.05



1.4.5. Conclusiones de la práctica.

- El diseño del sistema en lazo cerrado muestra similares respuestas a las entradas tipo paso y rampa.
- La simulación del Péndulo Invertido es realizada en variables de estado, con el bloque “State Space” de simulink
- Las respuestas del sistema del Péndulo Invertido están simuladas mediante el bloque “scope” de Simulink

ANEXO 5: PRÁCTICA#4:

“Diseño de un algoritmo de control de variables de estado realimentado”

1.5.1. Objetivos.

Objetivo General:

- Diseñar un algoritmo de control realimentado en el espacio de estados

Objetivos específicos:

- Diseñar en Simulink el sistema realimentado del Péndulo Invertido.
- Obtener mediante la fórmula de Ackerman las ganancias del sistema
- Realizar un análisis de la salida de la planta y observar si el controlador estabiliza el sistema

1.5.2. Recursos utilizados.

- Software de Matlab/Simulink
- Modelo Linealizado del Péndulo Invertido
- Modelo en el espacio de estados

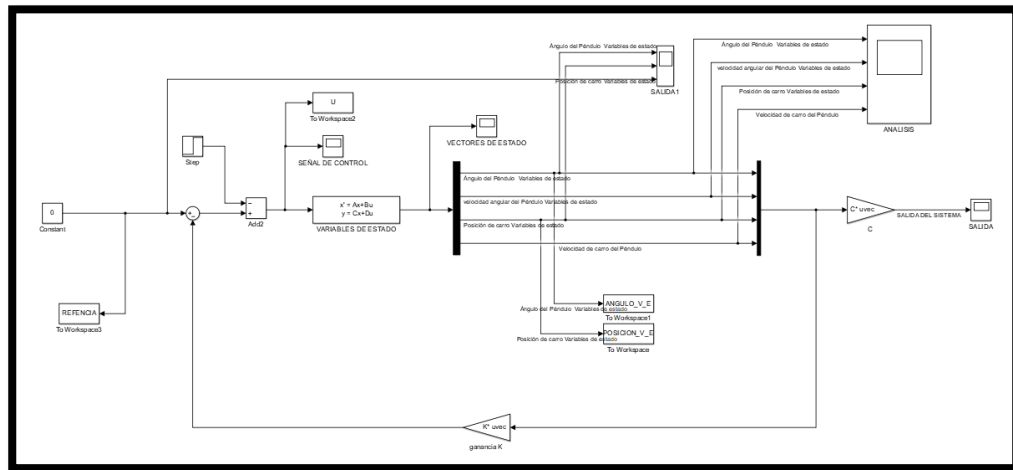
1.5.3. Tiempo estimado.

El tiempo estimado para la realización de esta práctica es de 2 horas.

1.5.4. Desarrollo.

SIMULACIÓN EN SIMULINK

1. Diseñar el sistema realimentado en variables de estado en Simulink teniendo los vectores de estado indicado en la Práctica 1. El diseño se evaluará en Pract4_Variables_de_estado.



EXPERIMENTACIÓN

1. Obtener los polos del sistema en Matlab y graficarlos. Los polos del sistema se los puede obtener de varias formas en el espacio de estado, función de transferencia y matemáticamente en forma matricial $\det(sI-A) = 0$. Los códigos de Matlab se evaluarán en los archivos Cod_Pract4.

Respuesta de Matlab para obtener los Polos del sistema:

```
Command Window
>> phi = 0
phi_dot = 0.2495
x = 0
x_dot = 0.3881

c =
    phi    phi_dot    x    x_dot
    phi    1         0    0     0
    x      0         0    1     0

d =
    u
    phi 0
    x   0

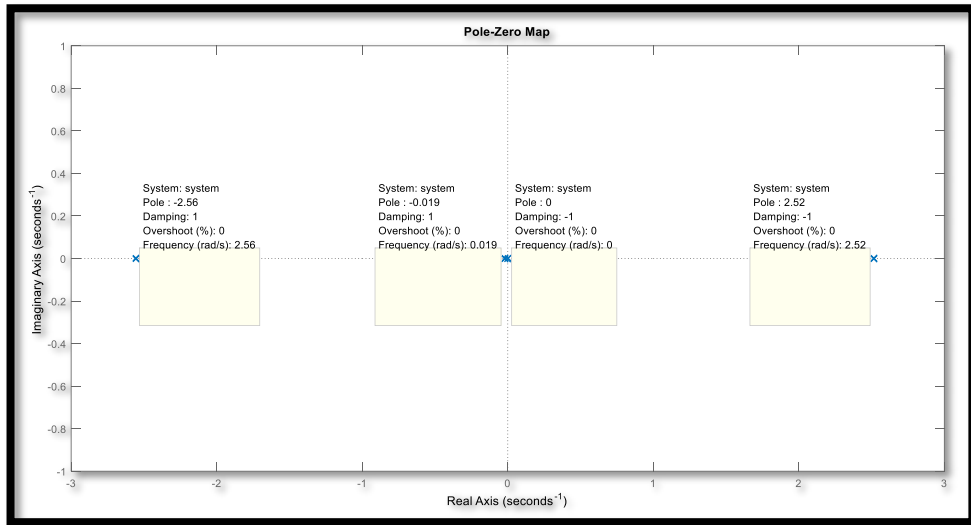
Continuous-time state-space model.

>> P=eig(A)
pzmap(system)

P =
    0
   -2.5571
    2.5171
   -0.0190

fx >>
```

Mapecto de polos del sistema en Matlab:



2. Obtener las ganancias K la ley de control de realimentación de estado. Mediante la fórmula de Ackerman ($K = \text{acker}(A, B, \text{Polos})$) se obtiene las ganancias K.

```
K =
1.0e-14 *
0.4594 -0.0068 0 -0.0034

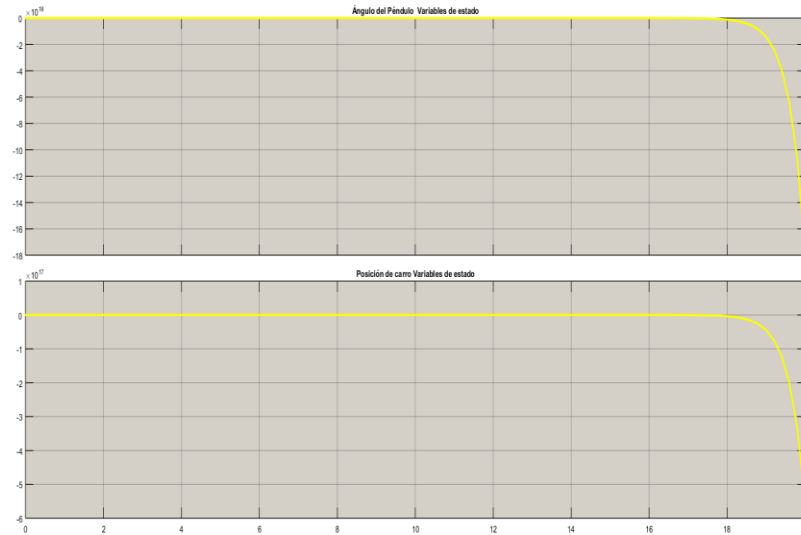
>> k
Undefined function or variable 'k'.

k
Did you mean:
Undefined function or variable 'k'.

Did you mean:
>> K

K =
1.0e-14 *
0.4594 -0.0068 0 -0.0034
```

3. Análisis de la respuesta del sistema al controlador en variables de estado



1.5.5. Conclusiones de la práctica.

- Las ganancias del sistema K se obtiene mediante la ley de control aplicando la fórmula de “Ackerman $K = \text{acker}(A, B, \text{Polos})$ ” en Matlab.
- Se diseña y analiza el controlador realimentado de variables de estado en el software especializado Matlab/Simulink.
- Analizando la salida del sistema se observa inestabilidad, no se logra los objetivos deseados habría que Regular el sistema mediante el método de ubicación de polos deseados.

ANEXO 6 PRÁCTICA#5:

“Diseño de un algoritmo de control variables de estado mediante Ubicación de Polos Deseados”

1.6.1. Objetivos.

Objetivo General:

- Diseñar un algoritmo de control en variables de estado mediante el método Ubicación Polos Deseados

Objetivos específicos:

- Estabilizar el sistema del Péndulo Invertido
- Realizar la Regulación del sistema mediante las características de criterio del diseño y hallar los polos dominantes
- Análisis con diferentes Polos deseados y observar cual tiene mejor respuesta
- Obtener las ganancias K del sistema

1.6.2. Recursos utilizados.

- Matlab/Simulink
- Modelo Linealizado del Péndulo Invertido
- Modelo en el espacio de estados

1.6.3. Tiempo estimado.

El tiempo estimado para la realización de esta práctica es de 2 horas.

1.6.4. Desarrollo.

SIMULACIÓN EN SIMULINK

1. Identificar el criterio de desempeño del sistema
 - Tiempo de asentamiento (T_s) 5 segundos
 - % Sobreimpulso 5%
2. Realizar Regulación de estados teniendo el criterio de desempeño del sistema y especificaciones del dominio del tiempo se obtiene los polos de $s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$. Se utiliza las siguientes fórmulas para hallar los polos dominantes.

$$OS\% = e^{-\frac{\pi}{\zeta}\sqrt{1-\zeta^2}} = 0.944$$

$$T_s = \frac{\pi}{\zeta\omega_n} = 0.66$$

$$\omega_d = \omega_n\sqrt{1-\zeta^2} = 1.05$$

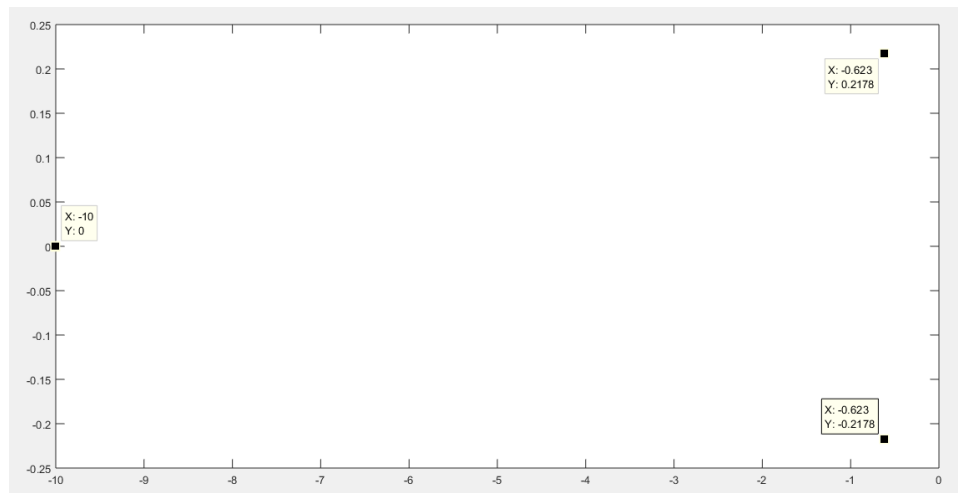
$$\sigma = \zeta\omega_n = 0.95$$

$$\beta = \cos^{-1}\zeta = 0.809$$

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$$

Si los polos dominantes son $-\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} = 0.6230 \pm j0.2178$

Mapeo de los Polos re-ubicados:

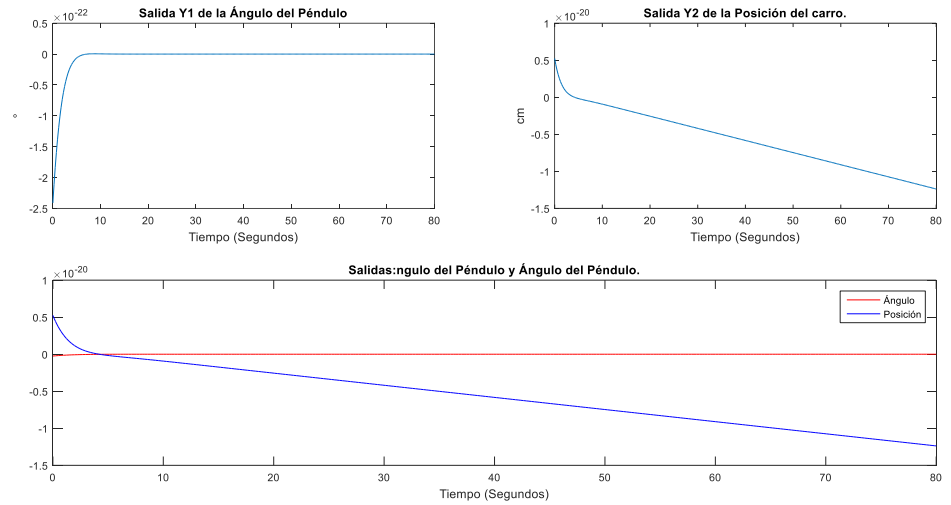


3. Utilizando el software de Matlab se realizan prueba con diferentes ganancias y los Polos deseados para observar la mejor respuesta del sistema. Los códigos de Matlab se evaluarán en los archivos Cod_Pract5.

- Prueba 1

$K1 = [27.5463; 4.8359; 0; -0.05]$

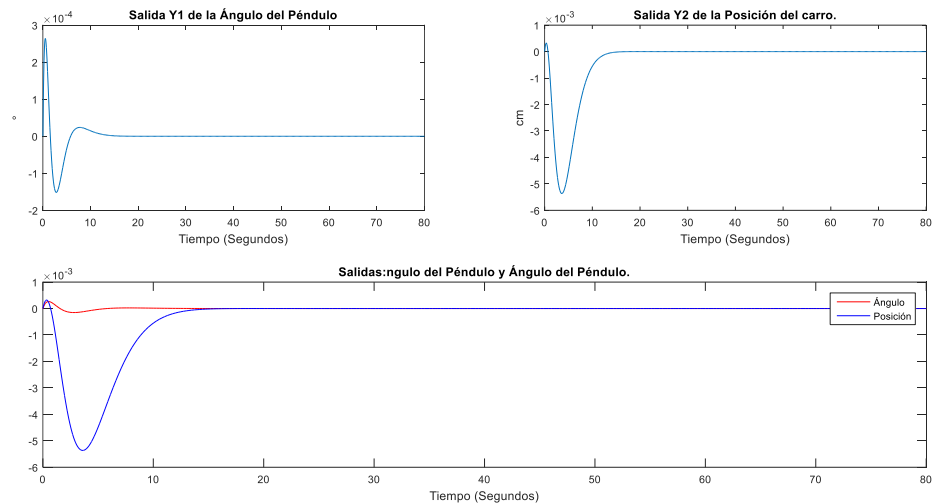
$\text{Polos_deseados}_1 = [-0.6230 + 0.2178i; -0.6230 - 0.2178i; 0; 0]$



- Prueba 2

$K2 = [185.5068 \ 67.5340 \ -4.4495 \ -14.5857]$

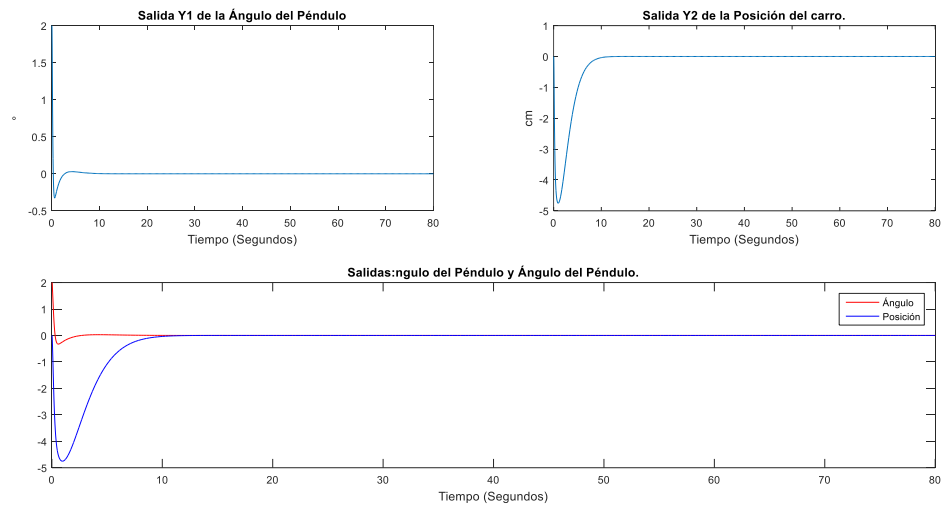
$\text{Polos_deseados}_2 = [-0.6230 + 0.2178i; -0.6230 - 0.2178i; -5; -5]$



- Prueba 3

$$K3 = [559.28 \ 169.9197 \ -17.7981 \ -54.6333]$$

$$\text{Polos_deseados}_3 = [-0.6230+0.2178i; -0.6230-0.2178i; -10; -10]$$



1.6.5. Conclusiones de la práctica.

- Se logra estabilizar el sistema mediante el método Ubicación de Polos Deseados
- Se obtiene las ganancias del sistema con la fórmula de Ackerman
- Por medio de las regulaciones de estado se obtiene los polos dominantes
- Se realizan pruebas con diferentes ubicaciones de polos y se observa que, por cada ganancia calculada por ubicaciones de polos, mientras más lejos se ubican los polos reales en el semiplano s negativo se obtiene mejor estabilización.

ANEXO 7 PRÁCTICA#6:

“Análisis de la respuesta transitoria y estacionaria del sistema del Péndulo Invertido”

1.7.1. Objetivos.

Objetivo General:

- Lograr que el estudiante analice el sistema del Péndulo Invertido a partir de la respuesta transitoria y estacionaria.

Objetivos específicos:

- Análisis del error de estado estacionaria al sistema del Péndulo Invertido ante la entrada de prueba tipo paso
- Interpretar la respuesta transitoria de un sistema ante la entrada de prueba tipo paso
- Implementar y utilizar las herramientas de Matlab/Simulink para analizar la respuesta transitoria y estacionaria del sistema Péndulo Invertido

1.7.2. Recursos utilizados.

- Matlab/Simulink
- Modelo Matemático del sistema Péndulo Invertido

1.7.3. Tiempo estimado.

El tiempo estimado para la realización de esta práctica es de 2 horas.

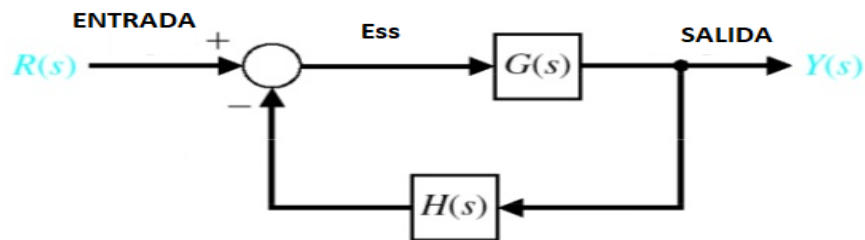
1.7.4. Desarrollo.

SIMULACIÓN EN SIMULINK

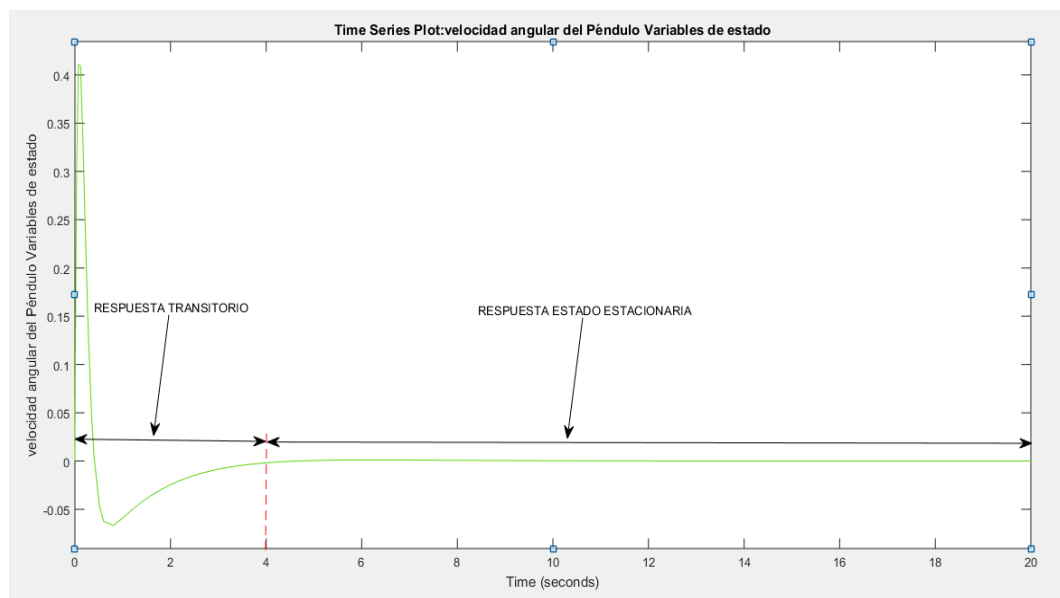
1. Diseño de un sistema de cuarto orden (Péndulo Invertido)
2. Para obtener una ganancia K con un porcentaje de sobreelongación de 5% y tiempo de asentamiento de 5 segundos. Obtener todas las especificaciones de tiempo de subida, tiempo pico y tiempo de asentamiento
3. Obtener los errores de estado estacionario del sistema ante la entrada pruebas Tipo paso.
4. Concluir acerca del comportamiento del sistema a partir de respuesta ante cada una de las entradas anteriores y comprobar si los criterios solicitados en el diseño se dan.

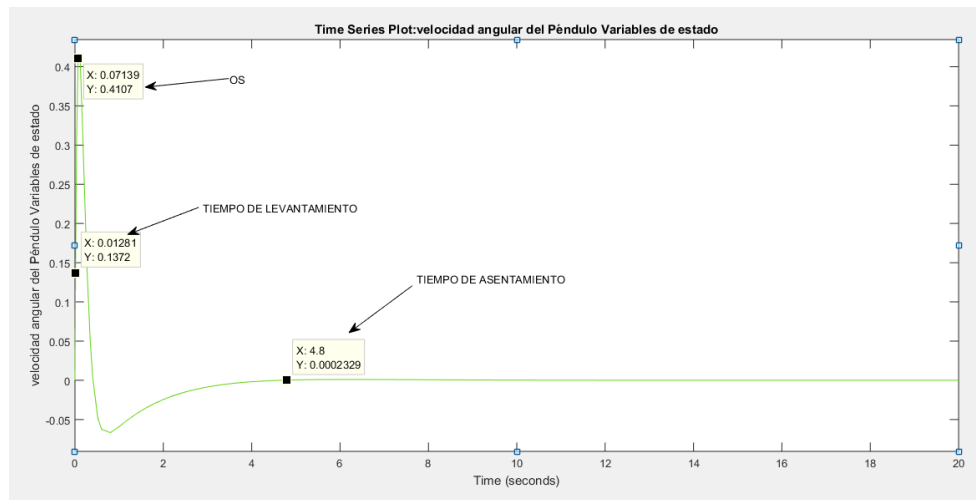
EXPERIMENTACIÓN

1. Diseño del sistema Péndulo Invertido en diagrama de Bloques.



2. Obtenemos $K = [559.28 \ 169.9197 \ -17.7981 \ -54.6333]$ con los criterios de diseño porcentaje de sobreelongación de 5% y tiempo de asentamiento de 5 segundos





3. El sistema se estabiliza según el criterio dado en 5 segundos y además se obtiene las otras especificaciones restantes de la respuesta transitoria.

1.7.5. Conclusiones de la práctica.

- Se verifica que el péndulo se estabiliza según el tiempo de asentamiento otorgado y además se obtiene todas las especificaciones en estado transitorio del sistema.
- Se obtiene los errores en estado estacionario del sistema ante una tipo paso

ANEXO 8: PRÁCTICA#7:

“Diseño de observador de estados para el Péndulo Invertido”

1.8.1. Objetivos.

Objetivo General:

- Diseñar un observador de estado completo para el sistema del Péndulo Invertido

Objetivos específicos:

- Obtener las ganancias L mediante ubicación de polos deseados
- Obtener las nuevas matrices del sistema
- Implementar y Simular en Matlab los observadores de estado
- Comparar las variables de Posición y Ángulo de los controladores por variables de estado mediante Polos deseados con los observadores de estado completo

1.8.2. Recursos utilizados.

- Matlab/Simulink
- Modelo Matemático del sistema Péndulo Invertido
- Controlador por variables de estado mediante Polos ubicación de polos deseados

1.8.3. Tiempo estimado.

El tiempo estimado para la realización de esta práctica es de 2 horas.

1.8.4. Desarrollo.

SIMULACIÓN EN SIMULINK

1. Mediante la técnica de Polos deseados ubicar los polos para obtener las ganancias del observador de estado
2. Utilizando la ganancia K de la Práctica 5 obtener la ganancia L
3. Obtener los nuevos vectores de estado
4. Comparar el observador de estado completo con el controlador de la Practica 5.

EXPERIMENTACIÓN

4. Teniendo en cuenta los Polos deseados de la prueba 5 se obtiene los Polos deseados para diseñar los observadores. Los códigos de Matlab se evaluarán en los archivos Cod_Pract7.

1.

Polos_deseados = [-0.6230+0.2178i; -0.6230-0.2178i; -10; -10]

Polos deseados del estimador = [-10; -11; -12; -13]

2. Utilizando los polos y la ganancia k obtenemos la ganancia del estimador L

$K = [559.28 \ 169.9197 \ -17.7981 \ -54.6333]$

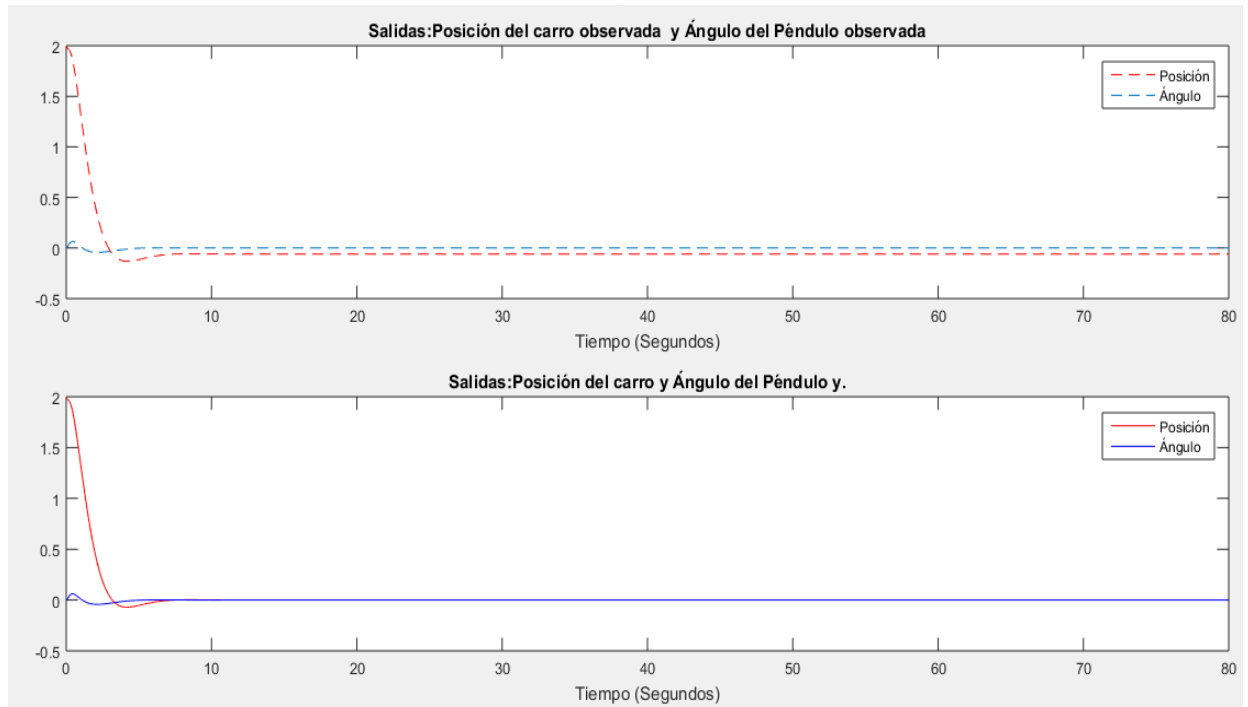
Utilizando la fórmula de Ackerman $L = \text{place}(A', C', P)'$

$L = [22.9112 \ -1.0494; 136.5156 \ -12.1998; -0.9620 \ 23.0297; -10.8336 \ 131.5781]$

3. Se obtienen los nuevos vectores de estado mediante los siguientes códigos de Matlab:

```
55      %Designación de los polos del estimador
56 -   L=place(A',C',P) '
57 -   Ace = [A-B*K B*K;zeros(size(A)) (A-L*C) ]
58 -   Bce = [B; zeros(size(B)) ]
59 -   Cce = [C zeros(size(C)) ]
60 -   Dce = [0;0]
61 -   est_cl = ss(Ace,Bce,Cce,Dce);
62 -   T = 0:0.005:80;                                % Tiempo de Simulación = 10 seg
63 -   U = ones(size(T));                                % u = 1, Señal Escalon
64 -   X0 = [2 -0.5 0 0 0 0 0 0];
65 -   %X0 = [0 0 0 0 0 0 0 0.5];                        %Condiciones iniciales
66 -   [Y,T,X]=lsim(est_cl,U,T,X0);                    %Simulación
```

4. Se simula el controlador en variables de estado con el observador en estado completo y se compara sus salidas.



1.8.5. Conclusiones de la practica.

- Se obtiene los nuevos Polos deseados para estimadores de estado.
- Se calcula la ganancia L del observador de estado
- Se encuentran los nuevos vectores de estado A_{ce} , B_{ce} , C_{ce} y D_{ce}
- Se diseña y se compara donde se aprecia que los observadores convergen mayor velocidad debido al tamaño de sus vectores de estado.

ANEXO 9 PRACTICA #8:
“Diseño del controlador PID”

1.9.1. Objetivos.

Objetivo General:

- Diseñar un controlador PID.

Objetivos específicos:

- Estabilizar el Péndulo Invertido con un controlador PID
- Mediante su respuesta tipo paso calcular K_i , K_p , K_d mediante el primer método de sintonización de Ziegler Nichols
- Analizar mediante pruebas ensayo-error las respuestas del sistema al alterar las ganancias del PID

1.9.2. Recursos utilizados.

- Software Matlab/Simulink
- Analisis transitorio del sistema

1.9.3. Tiempo estimado.

El tiempo estimado para la realización de esta prueba es de 2 horas.

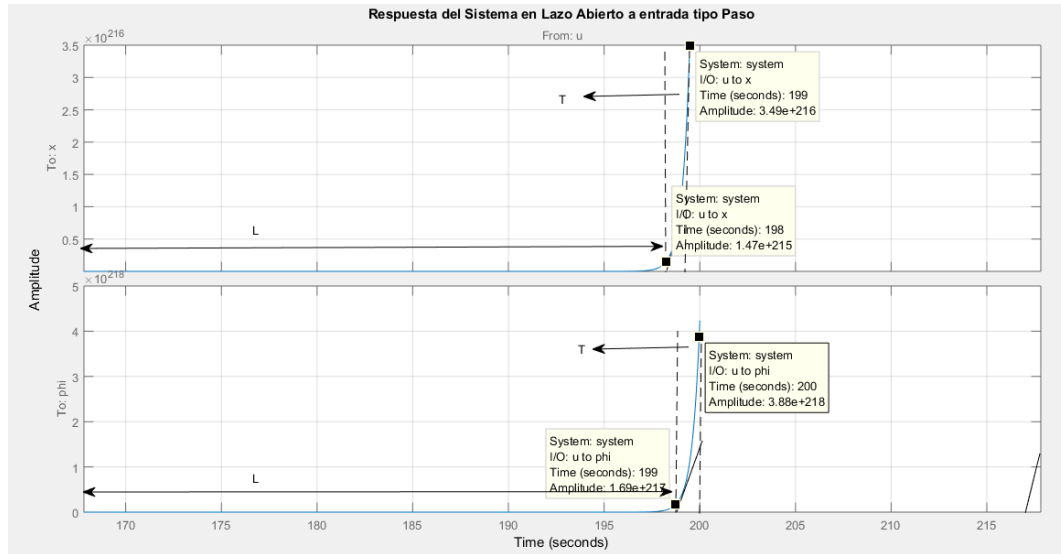
1.9.4. Desarrollo.

SIMULACIÓN EN SIMULINK

1. Analizar con una señal tipo paso el sistema en lazo abierto
2. Obtener K_p , K_i , K_d del sistema
3. Diseño del sistema en Simulink
4. Analizar la salida del sistema

EXPERIMENTACIÓN

1. Análisis mediante el primer método de sintonización de Ziegler-Nichols ante una entrada de prueba Tipo Paso.



2. Calcular K_p, K_i, K_d

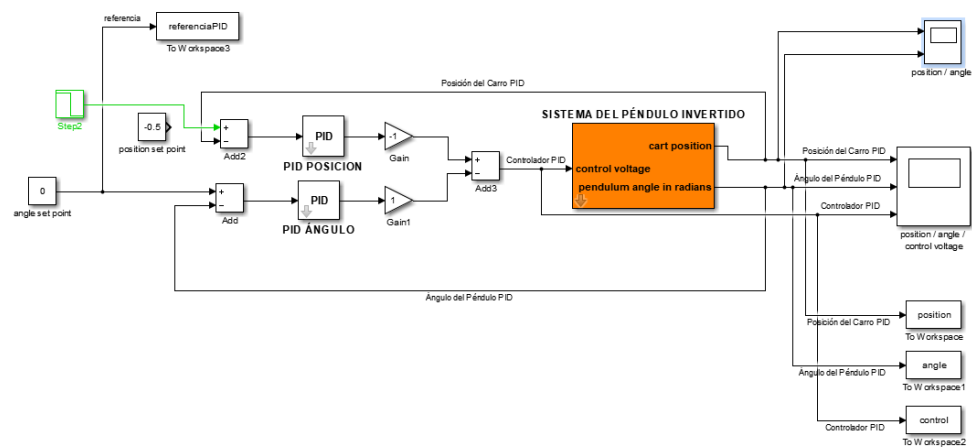
Controlador	K_p	T_i	T_d
PID	$1.2 \frac{T}{L}$	2L	0.5L

$K_{ix} = \frac{K_{px}}{T_{ix}} = 0.01$ $K_{px} = 2.75$ $K_{dx} = K_{px} * T_{dx} = 0.64$ (Proporcional, Integral derivativo correspondiente a posición)

$K_{io} = 40$ $K_{p0} = 5$ $K_{do} = 9$ (Proporcional Integral derivativo correspondiente a theta)

5. Mediante el Diseño del sistema en Simulink del controlador PID del archivo PRACT8_PID:. Mediante la técnica ensayo-error analizar los cambios si se ajusta las ganancias Proporcional, Integrador, Derivativo de Posicion y Angulo. Justificar su respuesta con un breve análisis.

Controlador	P	I	D
Posición	0.8	0.001	2
Péndulo	45	5	9



.....

.....

.....

.....

.....

.....

1.9.5. Conclusiones de la practica.

- Mediante el primer método de sintonía de Ziegler-Nichols se encuentra Kp, Ki, Kd para Posición del carro y ángulo del péndulo.
- Se estudia cómo trabaja el PID.
- Mediante Pruebas ensayo-error alterando las ganancias del PID se observa los cambios del sistema.

ANEXO 10 PRACTICA #9:

“Diseño del controlador MPC basado en modelo”

1.10.1. Objetivos.

Objetivo General:

- Diseñar un controlador MPC basado en modelo.

Objetivos específicos:

- Mediante la herramienta “MPC designer Task” diseñar el controlador MPC.
- Estabilizar el sistema del Péndulo Invertido.
- Diseñar en Simulink el sistema usando el bloque “MPC Controller”.
- Análisis de la salida del sistema.

1.10.2. Recursos utilizados.

- Software Matlab/Simulink
- Modelo matemático en el espacio de estados

1.10.3. Tiempo estimado.

El tiempo estimado para la realización de esta prueba es de 2 horas.

1.10.4. Desarrollo.

SIMULACIÓN EN SIMULINK

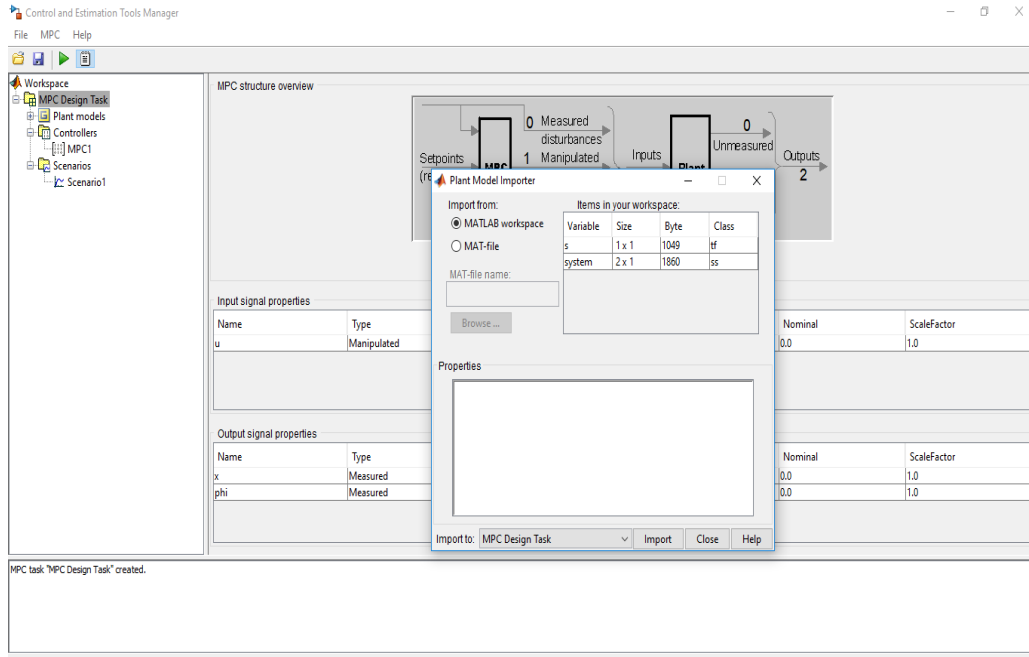
1. Importar el sistema en el espacio de estados en la herramienta “MPC Designer Task”.
2. Definir cuáles son las variable medibles y manipuladas
3. Definir los horizontes de predicción, horizonte de control, Constraints, Weight Tunning, Estimation.
4. Cargar el escenario y llevarlo al workspace.
5. Diseñar el sistema en Simulink usando el bloque MPC controller y simular las salidas del sistema.

EXPERIMENTACIÓN

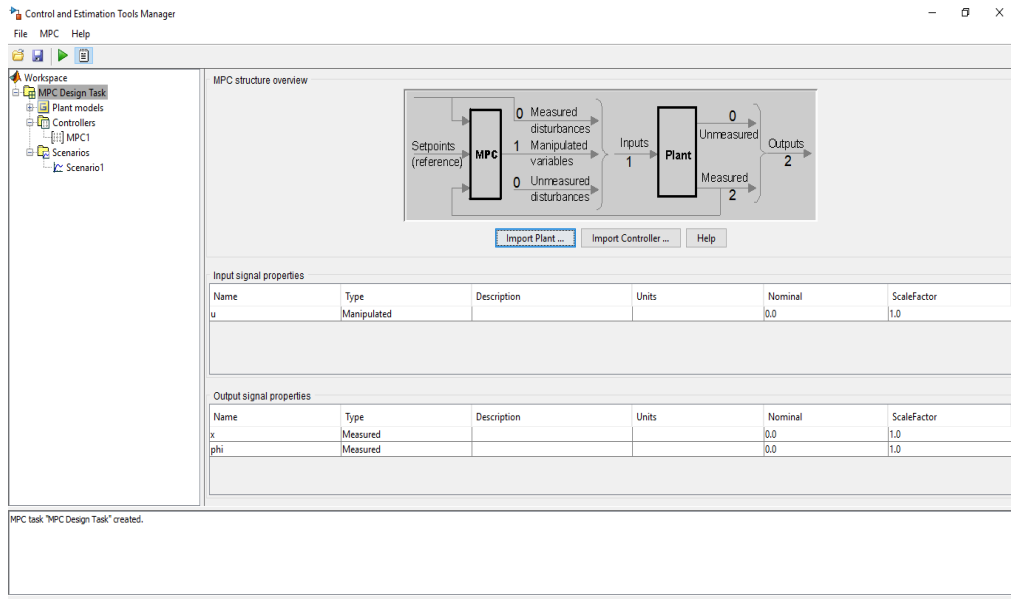
1. El sistema en el espacio de estados del sistema es:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 6.4363 & -0.0396 & 0 & -0.0125 \\ 0 & 0 & 0 & 1 \\ 0.2026 & -0.012 & 0 & -0.0194 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 0.2495 \\ 0 \\ 0.3881 \end{bmatrix}; \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Se importa la planta del Péndulo Invertido en el Toolbox MPC Design Task.



2. Se define posición y ángulo como variables medibles y la señal de control como manipulada.



- Se define los horizontes de predicción 10, horizonte de control 3, constraints, Weight Tuning y Estimation.

Model and Horizons

Plant model: model

Horizons

Control interval (time units): 0.4

Prediction horizon (intervals): 10

Control horizon (intervals): 3

☐ Blocking

Blocking

Blocking allocation within prediction horizon: Beginning

Number of moves computed per step: 3

Custom move allocation vector: [2 3 3]

Help

MPC task "MPC Design Task" created.

Control and Estimation Tools Manager

File MPC Help

Workspace

- MPC Design Task
 - Plant models
 - Controllers
 - MPC1
 - Scenarios
 - Scenario1

Model and Horizons Constraints Weight Tuning Estimation (Advanced)

Constraints on manipulated variables

Name	Units	Minimum	Maximum	Max Down Rate	Max Up Rate
u		-10	10		

Constraints on output variables

Name	Units	Minimum	Maximum
x			
phi			

Constraint Softening Help

MPC task "MPC Design Task" created.

Control and Estimation Tools Manager

File MPC Help

Workspace

- MPC Design Task
 - Plant models
 - Controllers
 - MPC1
 - Scenarios
 - Scenario1

Model and Horizons Constraints Weight Tuning Estimation (Advanced)

Overall

More robust Faster response

Value: 0.8

Input weights

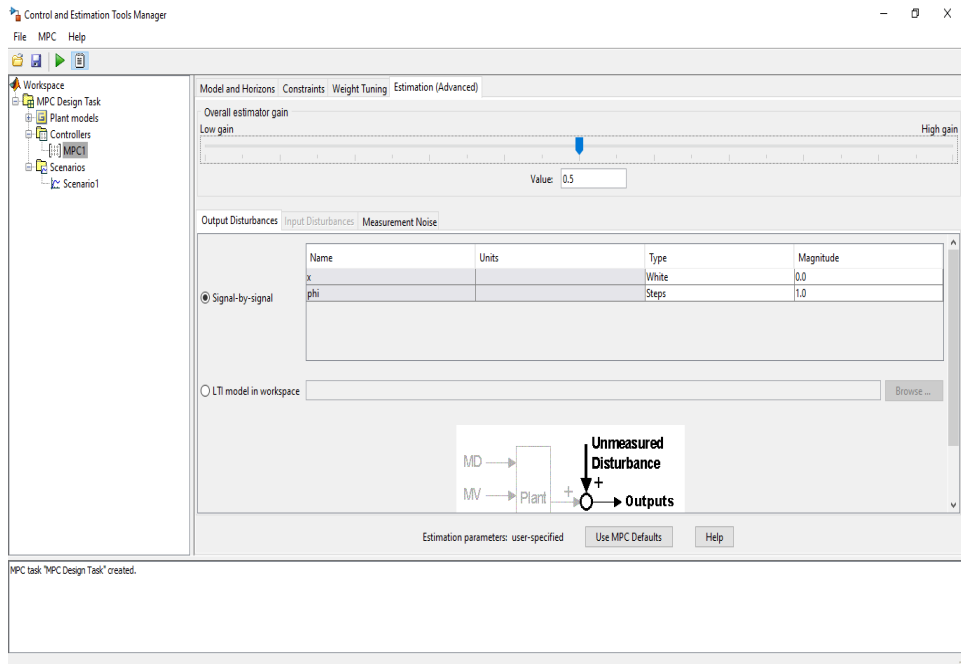
Name	Description	Units	Weight	Rate Weight
u			0	0.1

Output weights

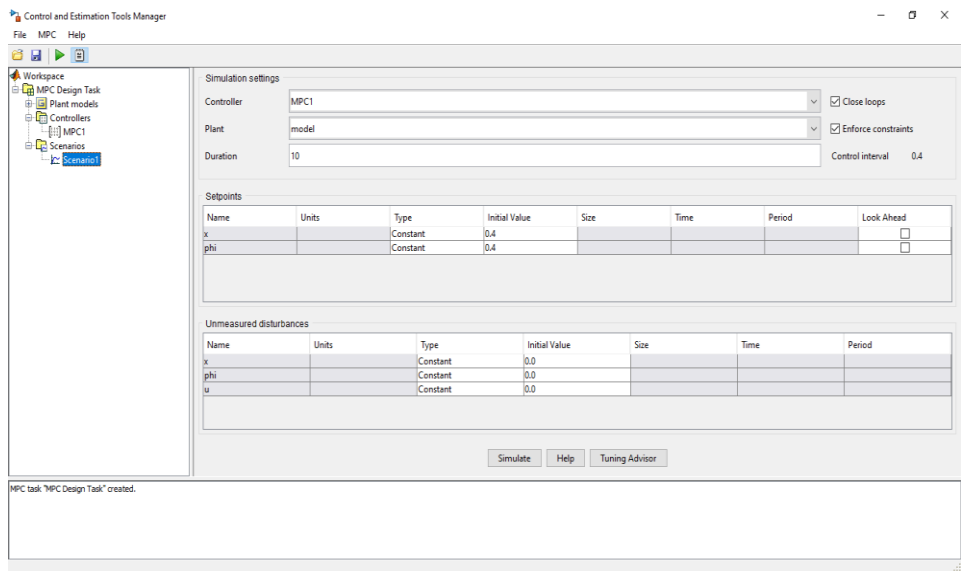
Name	Description	Units	Weight
x			1.0
phi			1.0

Help

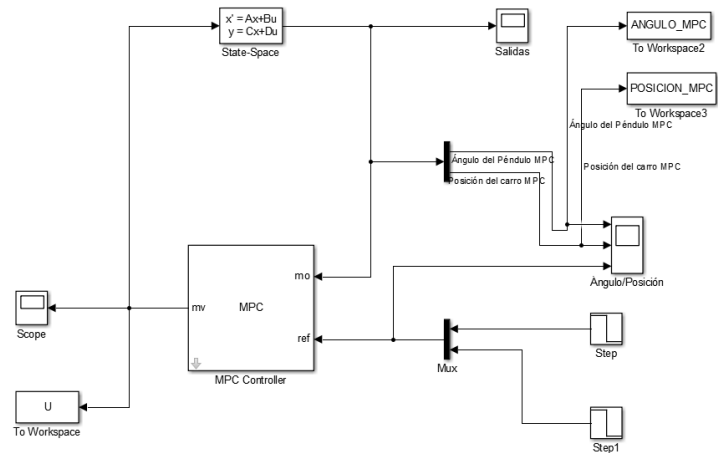
MPC task "MPC Design Task" created.



4. Se simula el escenario y sus resultados son enviados al workspace.



5. Diseño del sistema en simulink usando el bloque MPC controller.



6. Respuesta del sistema con el controlador MPC



1.10.5. Conclusiones de la practica.

- El diseño del controlador MPC se lo realiza en el espacio de estado usando la herramienta “MPC design Task” y se lo envía al workspace.
- Con el bloque “MPC Controller” de simulink se carga el diseño del controlador MPC enviado al workspace.
- Se logra evidenciar estabilidad en el diseño del controlador MPC al sistema del Péndulo Invertido Mediante Simulink y MPC designer.

ANEXO 11:PRÁCTICA #10:

“Análisis comparativo los algoritmos de control PID, Variables de estado, MPC”

1.11.1. Objetivos.

Objetivo General:

- Análisis comparativo del desempeño de los algoritmos de control (variables de estado, PID y MPC)

Objetivos específicos:

- Simular los algoritmos de control y enviar sus variables medibles al workspace
- Realizar un análisis comparativo usando la herramienta “Simulation data Inspector” de Matlab
- Indicar el sobrepaso máximo, tiempo de asentamiento de cada algoritmo de control
- Observar cual controlador tiene mejor desempeño y dar una breve explicación.

1.11.2. Recursos utilizados.

- Software Matlab/Simulink

1.11.3. Tiempo estimado.

El tiempo estimado para la realización de esta práctica es de 2 horas.

1.11.4. Desarrollo.

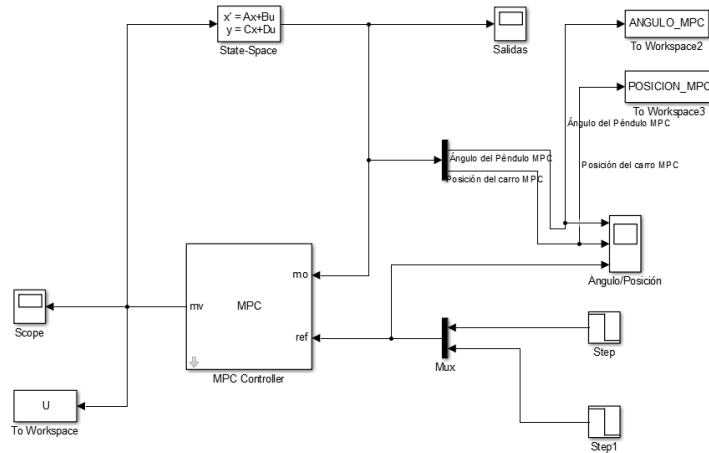
SIMULACIÓN SIMULINK

1. Simular los diferentes algoritmos de control en Simulink y enviarlos en el workspace.
2. Análisis de posición del carro y ángulo del péndulo en la herramienta “Simulation data Inspector” e indicar el tiempo de asentamiento, sobrepaso máximo.
3. Observar cual algoritmo tiene mejor desempeño y dar una breve explicación.

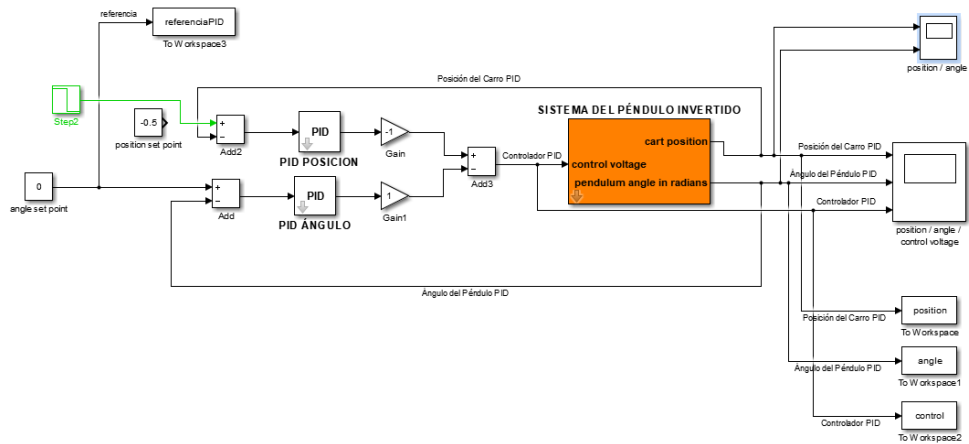
EXPERIMENTACIÓN

1. Diseño en simulink de los algoritmos de control (Variables de estado, PID, MPC).

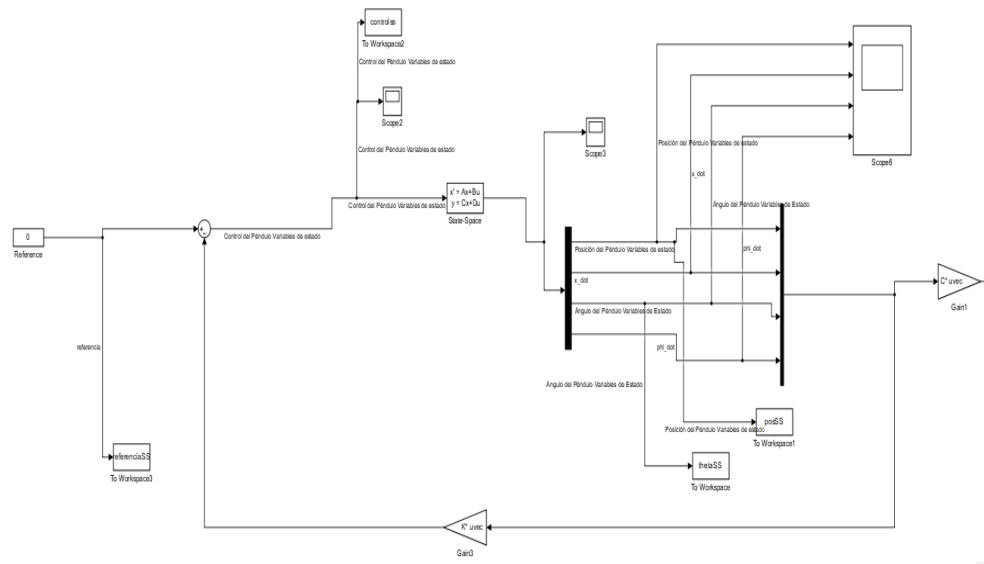
MPC



PID

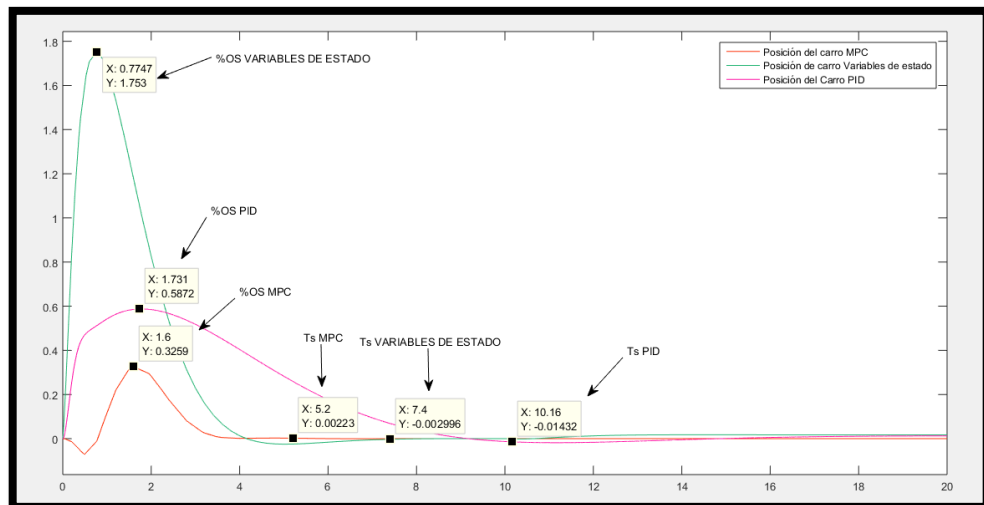


Variables de estado

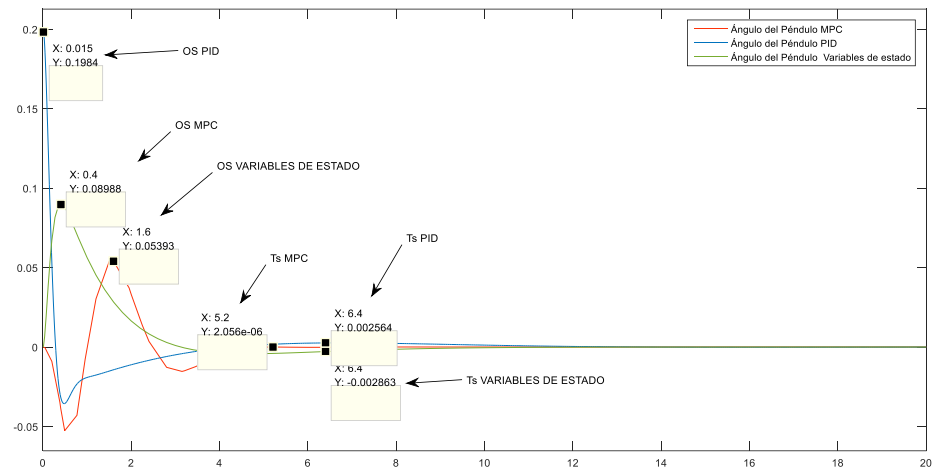


2. Las salidas de posición del carro y ángulo del péndulo son analizadas con la herramienta Simulation Data Inspector para poder comparar el desempeño de cada uno. A continuación, se muestra el análisis comparativo de Posición y Ángulo.

Posición



Ángulo



3. Analizando las gráficas anteriores pueden evidenciar que los tres controladores estabilizan el sistema del Péndulo Invertido con diferente tiempo de asentamiento. El controlador en MPC presenta mejor desempeño en cuanto tiempo de asentamiento, sobrepaso máximo tiene un MENOR valor de pico a los otros dos controladores.

1.11.5. Conclusiones de la Práctica.

- En los 3 controladores se evidencia la estabilización del péndulo
- El controlador en variables de estado es el que mejor desempeño presenta
- El controlador VARIABLES DE ESTADO, aunque tiene un tiempo mayor para estabilizarse no quiere decir que sea el de menor desempeño ya que no fuerza un cambio brusco en su valor de pico para buscar la estabilización del Péndulo Invertido