

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Escuela de Posgrado



DISEÑO DE UN SISTEMA DE CONTROL ÓPTIMO MULTIVARIABLE APLICADO A UN SISTEMA DE PÉNDULO DOBLE

Tesis para optar el grado académico de Magíster en
Ingeniería de Control y Automatización
que presenta:

Carlos Miguel Raymundo Luyo

Asesor:
Carlos Gustavo Perez Zuñiga

Lima, 2023

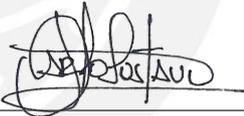
INFORME DE SIMILITUD

Yo, Carlos Gustavo Pérez Zuñiga, docente de la Escuela de Posgrado de la Pontificia Universidad Católica del Perú, asesor de la tesis titulada “Diseño de un sistema de control óptimo multivariable aplicado a un sistema de Péndulo Doble”, del autor Carlos Miguel Raymundo Luyo,

dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 10%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 03/05/2023.
- He revisado con detalle dicho reporte y la Tesis, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lima, 03 de mayo de 2023

Apellidos y nombres del asesor:	Carlos Gustavo Pérez Zuñiga
DNI: 41864666	Firma: 
ORCID: 0000-0001-5946-1395	

AGRADECIMIENTOS

Agradezco al Dr. Gustavo Pérez por sus recomendaciones y asistencia para la elaboración de esta tesis y a los profesores de la maestría de Control y Automatización que me dieron el conocimiento para lograr esta tesis.

Además a los profesores y cursos de la TU Darmstadt por el uso de la literatura empleada en esta tesis.

DEDICATORIA

*Quisiera dedicar esta tesis a mi familia comenzando con mi madre **Margarita Luyo** por su apoyo durante toda mi vida y a mi hermano **Fernando Raymundo** por ser un ejemplo para mí. Y sobre todo a mi padre **Rodolfo Raymundo** que trabajó y luchó para mí y mi familia hasta el día de su muerte durante esta pandemia del COVID-19, a él le dedico este trabajo en su honor.*

RESUMEN

La presente tesis se enfoca en el diseño, análisis y validación de los controles óptimos para el Sistema de Péndulo Doble. Para ello, se desarrolla un modelo matemático del sistema desde un enfoque energético y se analiza su dinámica en diferentes puntos de operación. Los controladores óptimos utilizados en este estudio tuvieron como objetivo principal llevar los péndulos a su punto de equilibrio y mantenerlos allí. Entre los controladores lineales implementados, se incluyen el LQR, el Observador con controlador de estado, el MPC y en cuanto al controlador no lineal el NMPC. Este último se basa en el solver ACADO y utiliza una interfaz en MATLAB para resolver el problema óptimo de control en su forma no lineal, considerando la dinámica real del sistema, restricciones en las variables de estado y entrada de control, así como en las incertidumbres.

Los resultados obtenidos fueron contrastados y se determina cuál de los controladores se adecua mejor al *Sistema de Péndulo Doble*.

La tesis esta estructurada de la siguiente manera:

Capítulo 1 provee una vista general de los trabajos e investigaciones sobre el *Sistema de péndulo doble*.

Capítulo 2 presenta una visión general de la base teórica así como el modelamiento del *Sistema de péndulo doble*. Además incluye la linealización, implementación y validación del modelo.

Capítulo 3 muestra las diferentes estrategias de control propuestas para la estabilización del *Sistema de péndulo doble*. Los controladores será posteriormente comparados en un mismo de operación de modo a fin de evaluar su performance. Por otro, se adiciona un apartado extra sobre el estudio de la robustez inherente del NMPC frente a perturbaciones.

Capítulo 4 presenta los comentarios e impresiones sobre la realización de la presente tesis.

Capítulo 5 termina la tesis con las conclusiones y observaciones.

ÍNDICE DE CONTENIDO

Informe de Similitud	II
Agradecimientos	III
Dedicatoria	IV
Resumen	V
Índice de Contenido	VI
Índice de figuras	IX
Índice de tablas	XII
INTRODUCCIÓN	1
Capítulo I: ESTADO DEL ARTE	4
1.1. Introducción	4
1.2. Estado del Arte de los péndulos dobles	4
1.3. Problemática de control del <i>Sistema de péndulo doble</i>	10
1.3.1 No linealidad del <i>Sistema de péndulo doble</i>	11
1.3.2 Oscilaciones y dificultad para la estabilización del <i>Sistema de péndulo doble</i>	11
1.3.3 Dificultad para el seguimiento específico de la trayectoria del <i>Sistema de péndulo doble</i>	12
1.4. Propuesta de solución	12
1.4.1 Modelamiento No Lineal del <i>Sistema de péndulo doble</i>	13
1.4.2 Estrategia de control no lineal	13
1.5. Objetivos	15
Capítulo II: MODELAMIENTO DEL SISTEMA DE PÉNDULO DOBLE	16
2.1. Estudio del <i>Sistema de Péndulo doble</i>	16
2.2. Modelamiento Físico	17
2.2.1 Obtención analítica de las ecuaciones de LAGRANGE	17
2.2.2 Acoplamiento	22
2.2.3 Resultados de simulación	25
- Resultados de la simulación sin excitación	25
- Resultados de la simulación con excitación sinusoidal	26
2.2.4 Linealización del <i>Sistema de péndulo doble</i>	27
2.2.5 <i>Sistema de péndulo doble</i> en espacio de estado	28
2.2.6 Sistema lineal de doble péndulo	32
2.2.7 Espacio de Estado del Sistema linealizado de doble péndulo	36
2.2.8 Linealización en software	37
2.2.9 Los valores propios de los modelos de espacio de estado	39
2.2.10 Controlabilidad y Observabilidad del Sistema	41
2.2.11 Criterios de Controlabilidad y Observabilidad	42
-Criterio de Kalman	42
-Criterio de Gilbert	43
-Criterio de Hautus	44

2.2.12 Criterios de Controlabilidad y Observabilidad en MATLAB	45
2.2.13 Conclusiones sobre Controlabilidad y Observabilidad del Sistema linealizado	47
Capítulo III: DISEÑO DE CONTROLADORES AVANZADOS PARA UN Sistema de péndulo doble	49
3.1. Diseño de un Controlador lineal cuadrático	49
3.1.1 Obtención de la ley de control óptima	52
3.1.2 Diseño de un LQR en MATLAB	54
3.1.3 Prueba del LQR	56
3.1.4 Simulación del LQR	58
3.1.5 Prueba de las matrices Q y R	60
3.1.6 Prueba del LQR en un punto de operación	61
3.2. Diseño de un Observador y un Controlador con Observador de estado	63
3.2.1 Diseño del Observador de Estado	63
3.2.2 Diseño del Observador de Estado con controlador	65
3.2.3 Convergencia del Observador	65
3.2.4 Elección de la matriz de retroalimentación L	66
3.2.5 Diseño del Observador de Estado con Controlador de Estado	67
3.2.6 Teorema de separación	68
3.2.7 Diseño del Observador de Estado con controlador de estado en MATLAB y SIMULINK	69
3.3. Diseño de un Observador Reducido y un Controlador de estado	74
3.3.1 Diseño del Observador Reducido de Estado con controlador de estado en MATLAB y SIMULINK	76
3.4. Control Lineal Predictivo Basado en Modelo (MPC)	81
3.4.1 Principio básico del Control Lineal Predictivo Basado en Modelo	81
3.4.2 Formulación matemática del Control Predictivo Óptimo .	83
Obtener el estado para el tiempo t_i	84
Calcular la entrada óptima en el intervalo de tiempo (t_i a $t_i + T_p$)	85
Aplicar la señal de entrada Óptima	86
Obtener el estado ahora para el nuevo tiempo t_{i+1}	87
Calcular la entrada óptima en el intervalo de tiempo (t_{i+1} a $t_{i+1} + T_p$)	87
Aplicar la nueva señal de entrada Óptima	88
3.4.3 Formulación lineal del MPC	89
Viabilidad recursiva	90
3.4.4 Diseño Control Lineal Predictivo Basado en Modelo (MPC)	90
Conversión del Sistema continuo a discreto	91
Parámetros del controlador	93
Implementación del MPC lineal en MATLAB	93
Construcción las matrices para la predicción del sistema	94

Construcción de las matrices para la función de costo . . .	96
Construcción de las matrices para la predicción inicial, y los límites inferiores y superiores de la varia- bles de optimización	97
Solución del MPC en cada tiempo de muestreo	97
Los resultados del MPC lineal	98
Prueba MPC lineal bajo otros puntos de operación . . .	101
3.5. Diseño Control No Lineal Predictivo Basado en Modelo (NMPC)	105
3.5.1 Principio básico del Control No-Lineal Predictivo Basado en Modelo	105
Implementación del MPC No-lineal basado en ACADO en MATLAB	107
Los resultados del MPC No-lineal basado en ACADO . .	110
3.5.2 Robustez inherente del NMPC vs MPC	113
Error en el modelo del NMPC - MPC	114
Resultado de prueba de robustez del controlador NMPC	115
3.6. Comparación de los controladores y resultados finales	119
3.6.1 Indicadores de performance de los controladores estadia- dos	121
RMSE para medir la precisión del modelo	121
Indicadores para medir la performance de los controla- dores	122
3.7. Análisis del NMPC basado en ACADO con restricciones y diferen- tes puntos de operación	123
Capítulo IV: CONCLUSIONES Y OBSERVACIONES	130
Capítulo V: RECOMENDACIONES Y PERSPECTIVAS	133
Bibliografía	138
Anexos	144

ÍNDICE DE FIGURAS

<i>Número</i>	<i>Página</i>
0.1. Caos del Sistema de Péndulo Doble.	1
0.2. Sistema real de un Péndulo Doble.	2
2.1. Representación de un <i>Sistema de péndulo doble</i>	16
2.2. Diagrama de bloques del <i>Sistema de péndulo doble</i>	24
2.3. Resultado de la simulación para los ángulos φ_1 y φ_2 con C.I. $x_0^T = [\frac{\pi}{1,01} 0 \frac{\pi}{1,01} 0]$ sin excitación a la entrada.	25
2.4. Resultado de la simulación de una excitación sinusoidal durante 50 segundos.	26
2.5. Resultado de la simulación de una excitación sinusoidal durante dos segundos.	27
2.6. Sistema <i>Controlable</i> y <i>Observable</i>	42
2.7. Representación de un <i>Sistema linearizado de péndulo doble</i> cuando los ángulos de los péndulos hacen un ángulo perpendi- cular.	48
3.1. Función de costo J y el área que hace entre y y y_∞	50
3.2. Lazo cerrado y la ley óptima de control	52
3.3. Modelo en SIMULINK del control del sistema a lazo cerrado.	56
3.4. Resultado de la simulación en SIMULINK del control del sistema a lazo cerrado.	57
3.5. Los resultados de la simulación para el primer punto operación	59
3.6. Los resultados de la simulación para el segundo punto de opera- ción	59
3.7. Respuesta del controlador LQR	62
3.8. Modelo y Controlador del Observador-Luenberger	64
3.9. Observador y Controlador de Estado	68
3.10. Modelo y controlador del Observador	70
3.11. Observador con 1 vez los polos del Sistema a lazo cerrado	72
3.12. Observador con 2.5 veces los polos del Sistema a lazo cerrado	73
3.13. Observador con 10 veces los polos del Sistema a lazo cerrado	73
3.14. Observador Luenberger reducido	76
3.15. Controlador de Estado con Observador Luenberger reducido	77
3.16. Estructura del Observador reducido en bloques en SIMULINK	78

3.17.	Observador completo vs. Observador reducido	78
3.18.	Diferencia entre el Observador completo y reducido	79
3.19.	Oscilaciones en la respuesta del Observador completo	80
3.20.	Observador completo y reducido convergencia	80
3.21.	Ejemplo didáctico para el entendimiento del MPC	82
3.22.	Tiempo de muestreo a lazo cerrado	84
3.23.	Secuencia de formulación del MPC	84
3.24.	Obtención de la señal de control $u(t)$ y variable $x(t)$ en tiempo t	84
3.25.	Obtención del estado $x(t_i)$ en t_i	85
3.26.	Cálculo de la señal de control $u(t)$ para el intervalo $t_i < t \leq t_i + T_p$	85
3.27.	Extracción de la señal de control $u(t)$ para el intervalo acortado de $t_i < t \leq t_{i+1}$	86
3.28.	Obtención del estado $x(t_{i+1})$ en t_{i+1}	87
3.29.	Señal de control $u(t)$ para $t_{i+1} < t \leq t_{i+1} + T_p$	87
3.30.	Extracción de la señal de control $u(t)$ para el intervalo acortado de $t_{i+1} < t \leq t_{i+2}$	88
3.31.	Resultado de simulación del MPC lineal usando el comando QUADPROG	99
3.32.	Resultado de simulación MPC del primer punto de operación	102
3.33.	Resultado de simulación NMPC del tercer punto de operación	103
3.34.	Resultado de simulación NMPC del tercer punto de operación	104
3.35.	Los resultados de la simulación para el primer punto operación	111
3.36.	Resultado de simulación del NMPC basado en ACADO	111
3.37.	Tiempo de computo $210 \mu s$ para el NMPC a lazo cerrado implementado con ACADO	114
3.38.	Estándar MPC nominal versus MPC real	115
3.39.	Resultado de simulación del MPC No-lineal (NMPC) con una perturbación con valores aleatorios basado en ACADO	116
3.40.	CPUTime $228 \mu s$ para el NMPC con perturbación aleatoria	117
3.41.	Ruido blanco Gaussiano con -6 dB	117
3.42.	Resultado de simulación del MPC No-lineal (NMPC) con un ruido blanco Gaussiano de -6dB basado en ACADO	118
3.43.	CPUTime $217 \mu s$ el NMPC con ruido blanco Gaussiano a - 6dB	119
3.44.	Comparación de los controladores LQR, Observador, Observador reducido, MPC y NMPC	120

3.45.	Representación de un <i>Sistema de péndulo doble</i> para un tercer punto de operación	124
3.46.	Resultado de simulación NMPC del tercer punto de operación .	124
3.47.	Representación de un <i>Sistema de péndulo doble</i> para un cuarto punto de operación	125
3.48.	Resultado de simulación NMPC del cuarto punto de operación	126
3.49.	Representación de un <i>Sistema de péndulo doble</i> para un quinto punto de operación	127
3.50.	Resultado de simulación NMPC del quinto punto de operación	127
3.51.	Representación de un <i>Sistema de péndulo doble</i> para un sexto punto de operación	128
3.52.	Resultado de simulación NMPC del sexto punto de operación .	129



ÍNDICE DE TABLAS

<i>Número</i>		<i>Página</i>
2.1.	Variable y unidades para el modelado físico del <i>Sistema de péndulo doble</i>	18
2.2.	Aproximaciones	28
2.3.	Puntos de equilibrio de los Péndulos.	30
2.4.	Dimensión de las matrices	37
2.5.	Puntos de operación y Momentos.	39
2.6.	Controlabilidad de Kalman en diferente puntos de operación . .	45
2.7.	Observabilidad de Kalman en diferente puntos de operación . .	46
2.8.	Controlabilidad de Gilbert en diferente puntos de operación . .	46
2.9.	Observabilidad de Gilbert en diferente puntos de operación . .	46
2.10.	Controlabilidad de Hautus en diferente puntos de operación . .	47
2.11.	Observabilidad de Hautus en diferente puntos de operación . .	47
3.1.	Nomenclatura de un MPC lineal	106
3.2.	Nomenclatura del un MPC no lineal	106
3.3.	Indicador RMSE para el modelo optimal	121
3.4.	Indicadores de la performance de los controladores	123

INTRODUCCIÓN

La dinámica del movimiento del *Sistema de péndulo doble* es complicada debido a las trayectorias independientes de cada péndulo. Mientras que el primer péndulo sigue un movimiento predecible a lo largo de arcos de circunferencia, el segundo péndulo experimenta movimientos caóticos y no definidos. A pesar de que este sistema es relativamente fácil de construir, su estudio y control presentan un desafío debido a este último factor, como se puede observar en la figura 0.1 generada en MATLAB los tipos de movimiento que realiza cada uno de los péndulos. En color azul se muestra el movimiento de circunferencia del primer péndulo altamente predecible y el segundo péndulo en rojo con ese movimiento caótico como antes se mencionó.

En cuanto a la estructura mecánica del *Sistema de péndulo doble*, se presenta la figura 0.2, en donde se muestra una estructura real de un *Sistema de péndulo doble* como objeto de estudio en un laboratorio [1]. Este, consta de dos péndulos con masas m_1 y m_2 , y longitudes l_1 y l_2 , respectivamente. Sólo el primer péndulo está conectado a un pivote con motor que controla el sistema y el segundo péndulo, si se toma como referencia la capa terrestre, se encuentra unido al primer péndulo sin ningún motor de por medio. Debido a la gravedad terrestre, éste sistema experimenta un comportamiento caótico que puede ser modelado mediante ecuaciones acopladas de segundo grado

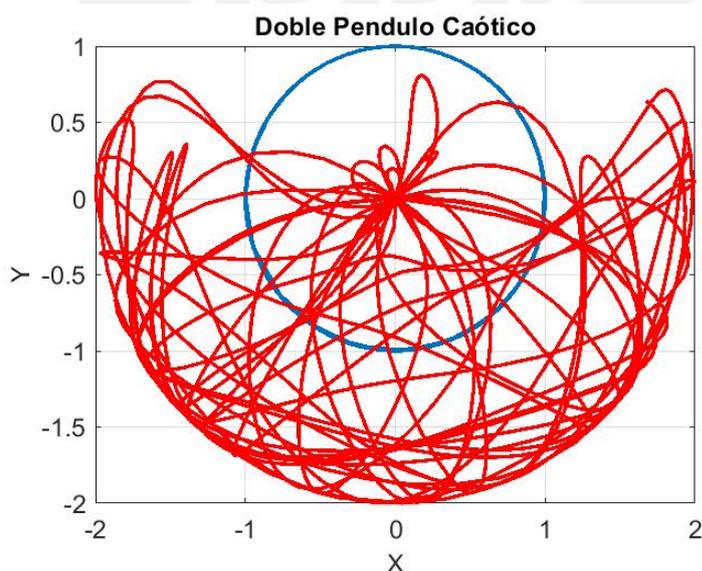


Figura 0.1: Caos del Sistema de Péndulo Doble.

Fuente : Elaboración propia

con una fuerte no linealidad.

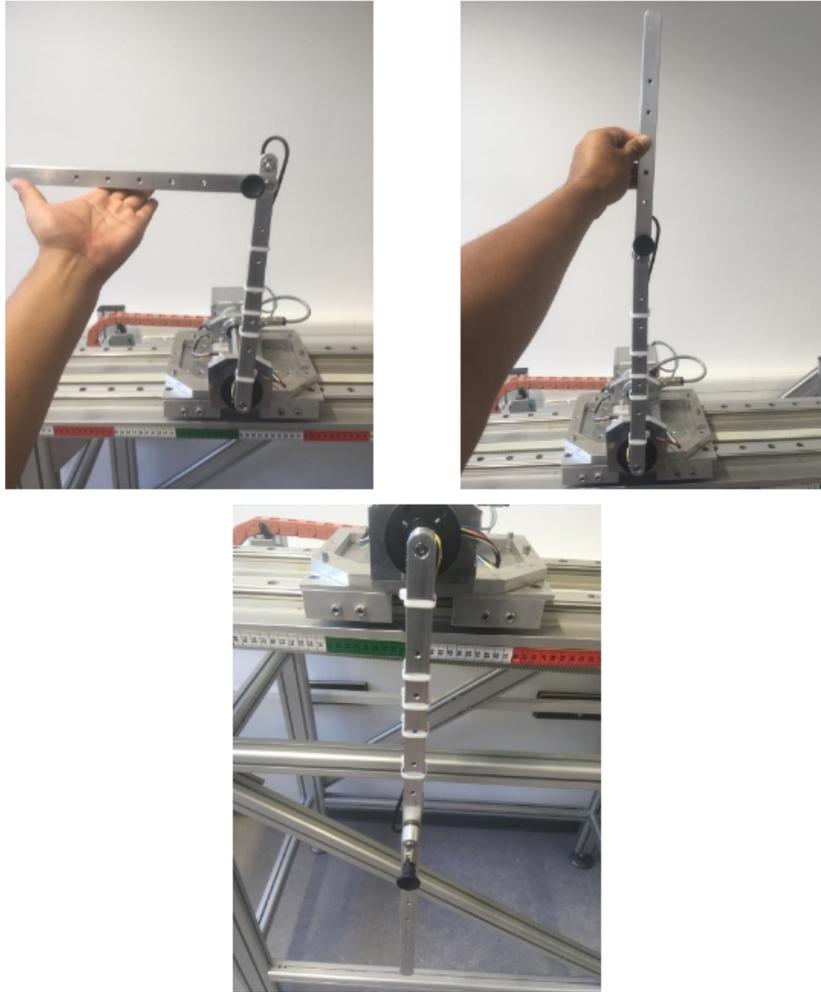


Figura 0.2: Sistema real de un Péndulo Doble.

Fuente : Instituto de Ingeniería de Control y Mecatrónica (rtm) de la TU Darmstadt.

Anteriormente, el uso del *Sistema de péndulo doble* se restringía al ámbito académico. No obstante, un tema interesante a tratar es su aplicación práctica del control y uso de este tipo de sistemas en prótesis robóticas o su uso en el mecanismo de sillas de ruedas eléctricas con capacidad de subir escalones. Aunque esta tesis no se enfoca en dichas aplicaciones prácticas, se centrará en el modelamiento y control del sistema para mejorar su eficiencia energética y robustez. En el futuro, se podría aplicar solución de control óptima presentada en esta tesis a dichas aplicaciones prácticas.

En la presente tesis se propone desarrollar, diseñar e implementar una estrategia de control óptima para dar solución a la dinámica no lineal del *Sistemas de*

péndulo doble caracterizada con movimientos no periódicos e inestables. Para ello se emplea una estrategia de control lineal y no lineal la cual a lo largo del desarrollo de la tesis, muestra cuál de ellos es el más efectivo para el control de este tipo de sistemas. Por ello, el uso del modelo del sistema se propone por medio de la linealización del mismo en un punto de operación o por medio de la descripción por ecuaciones diferenciales.

Para alcanzar los objetivos, se estructura la presente tesis en diferentes fases descritas en diferentes capítulos.

El primer capítulo describe el estado del arte de los diferentes *Sistemas de péndulo doble* y las estrategias de control utilizados. Posteriormente, se detalla la problemática que presenta el sistema y la respectiva propuesta de solución que gobierna esta tesis.

Precisamente, el capítulo 2 muestra el modelamiento del *Sistema de péndulo doble* desde el punto de vista energético así como su respectiva simulación de las ecuaciones diferenciales que describen la dinámica del sistema bajo diferentes excitaciones a la entrada del sistema. Además, se detalla la linealización de la misma para describirlo por medio de las matrices en espacio de estados para el uso de controladores óptimos lineales. Para eso, se prueban diferentes puntos de operaciones por medio de los criterios de observabilidad y controlabilidad.

En cuanto al desarrollo, diseño e implementación de los controladores óptimos avanzados se presentan cada uno de ellos en el capítulo 3. Como se mencionó anteriormente, se desarrollan diferentes tipos de controladores óptimos avanzados dependiendo de la linealidad o no linealidad del modelo descrito. En cuanto al uso modelo lineal, se presenta el controlador LQR, Observador completo y reducido con controlador y el MPC lineal bajo un mismo punto de operación. Bajo esa premisa, se desarrolla el NMPC para el modelo no lineal y se contrasta bajo el mismo punto de operación que el de los controladores lineales. Posteriormente, se prueba este controlador bajo otros puntos de operación y se prueba su robustez.

Finalmente en los capítulos 4 y 5 se presentan las conclusiones y observaciones, y recomendaciones y perspectivas respectivamente bajo los resultados obtenidos en cada fase de la tesis.

Capítulo 1

ESTADO DEL ARTE

1.1. Introducción

En este capítulo aborda los diferentes trabajos relacionados al control de *Sistema de péndulo doble* y sus usos prácticos. Además, se detalla las diferentes estrategias de control aplicadas para este tipo de sistemas que se produjeron en los últimos tiempos, partiendo de su importancia de las mismas para el propósito de esta tesis.

1.2. Estado del Arte de los péndulos dobles

A continuación se hará una breve descripción de los trabajos consultados que utilizan un *Sistema de péndulo doble*:

- **Control óptimo no lineal del péndulo doble subactuado para una grúa (*Nonlinear Optimal Control for the Underactuated Double-Pendulum Overhead Crane*)** El propósito de este trabajo de investigación es proponer un enfoque de control óptimo no lineal para grúas puente basado en un péndulo doble utilizados habitualmente en la industria, las obras de construcción y las operaciones de la cadena de suministro [2]. El control y la estabilización de este tipo de grúas puede ser un reto debido a las no linealidades y a la falta de actuadores en referencia a los grados de libertad.

El enfoque propuesto por los autores implica la linealización del modelo dinámico de la grúa y el sistema de doble péndulo en torno a un punto de operación, utilizando la expansión en serie de Taylor y matrices jacobianas. A continuación, se diseña un controlador de realimentación óptimo estabilizador (H-infinito) para el modelo linealizado de espacio de estados, que es la solución al problema de control óptimo no lineal bajo incertidumbre del modelo y perturbaciones externas. Las ganancias de realimentación del controlador se calculan mediante soluciones repetitivas de la ecuación algebraica de Riccati, y las estabilidad del

mismo se realiza por medio del análisis de Lyapunov. Además, se utiliza el filtro de Kalman H-infinito como estimador de estado robusto para implementar un control basado en la estimación de estado sin medir el vector de estado completo de la grúa y el sistema de doble péndulo. El enfoque propuesto consigue un seguimiento rápido y preciso de los puntos de consigna de referencia para todas las variables de estado.

- **Control de posicionamiento y eliminación de oscilaciones del puente grúa con dinámica de doble péndulo (*Positioning and Swing Elimination Control of the Overhead Crane System with Double-Pendulum Dynamics*)** El objetivo del estudio es abordar la dinámica de doble péndulo para un sistema de puente grúa [3] y desarrollar una estrategia mejorada de control antioscilación para el posicionamiento y la eliminación de oscilaciones. Para ello los autores introdujeron una entrada de control auxiliar, e implementaron una función de Lyapunov.

En cuanto a la estrategia de control, elaboraron un enfoque de control antioscilación no lineal considerando un análisis teórico para la convergencia de los estados del sistema. Las pruebas de simulación confirmaron la eficacia y robustez del enfoque propuesto y demostraron que supera a los métodos de control existentes. El estudio concluye que el planteamiento propuesto tiene importantes implicaciones para el control de sistemas de grúas puente con dinámica de doble péndulo.

- **Control en modo deslizante antioscilación para un péndulo doble tridimensional basado en un observador de estado extendido (*Anti-swing sliding mode control of three-dimensional double pendulum overhead cranes based on extended state observer*)** Éste trabajo se trata sobre el uso de una grúa basado en un péndulo doble en modo de movimiento tridimensional a fin de mejorar la eficiencia del transporte de carga [4]. Sin embargo, el control de los ángulos de oscilación de dos etapas presentan gran dificultad. Para resolver este problema, los autores proponen un método de control coordinado para el carro de la grúa de péndulo doble.

El método utilizado por los autores para el control empieza con un modelo tridimensional de grúa de péndulo doble simplificado. Se emplea además un método de control en modo deslizante, junto con un observador de estado extendido, para posicionar y suprimir la oscilación en

dos etapas de la grúa. El método propuesto tiene una gran robustez y capacidad antiinterferente, y se demuestra que consigue un posicionamiento rápido y preciso a la vez que suprime eficazmente la oscilación en dos etapas. El método se aplica en una planta experimental no lineal y, en comparación con otras propuestas, el control propuesto demuestra una reducción del tiempo de movilidad, una mejora de la eficacia del trabajo y una reducción del riesgo para la seguridad de los operarios.

- **Control de un péndulo doble inverso basado en una rueda que considera la dinámica híbrida (*Regelung eines Radbasierten Inversen Doppelpendels Unter Betrachtung Hybrider Dynamik*)**, es un trabajo relacionado al sistema denominado por su autor como Star Climbing Device (SCD) [5], donde se puede apreciar una aplicación directa respecto al péndulo doble. En este trabajo se aborda el diseño de una silla de ruedas eléctrica capaz de subir/bajar escaleras. Para eso, las ruedas se alinean de tal manera que se asemejan a dos péndulos los que mantendrán a la persona, sin movilidad, y la silla eléctrica en una posición tal que se mantengan estables a pesar del peso de ambos objetos. Por otro lado, se considera de la misma forma las incertidumbres que pudieran ser parte de las perturbaciones como movimientos de la persona en cuestión o algún tipo de fricción que presenten las ruedas con el suelo.

En el caso del controlador usado para este trabajo fue el de Dinámica Cero (Zero Dynamics) cuyo fin de mantener la dinámica interna del sistema en posición de equilibrio, sin movimiento y por ende estable. Para este propósito, se usó un modelo dinámico basado en el método de Lagrange, el cual se necesita linealizar para compensar la dinámica no lineal del SCD. Posteriormente, se utiliza un control adicional por retroalimentación junto con la metodología de desacoplamiento que finalmente logra la estabilización con el método de dinámica cero. Con eso, se garantiza que el objeto se quede en su posición de equilibrio de modo que tanto la silla de ruedas eléctrica junto a la persona con movilidad restringida permanezcan en una posición estable y segura.

- **Control y dinámica de la locomoción de una pierna que funciona como péndulo doble auto-impulsado (*Dynamics and Control of Locomotion of One Leg Walking as SelfImpact Double Pendulum*)**, es otro trabajo relacionado al *Sistema de péndulo doble* [6]. Para eso, se modela la locomoción de una pierna humana. La cual se asemeja a un péndulo

doble auto-impulsado. Además, se modela también las articulaciones que limitan el movimiento a solo muy cortos ángulo de giro. Este último punto es importante a tomarlo en cuenta ya que el *Sistema de péndulo doble* que se estudia en esta presente tesis a diferencia de éste trabajo es que claramente no posee éstas limitaciones.

En cuanto al controlador como se trata de un sistema linealizado, el autor de este trabajo propone un controlador lineal denominado como Retroalimentación Linealizada de Entrada-Salida (Input-Output Feedback Linearization). En este caso, los retroalimentadores se tratan de dos coeficientes utilizados como ganancias de control que regulan cada ángulo, variables del proceso, por separado dándoles autonomía. Asimismo, al trabajar con un sistema linealizado se debe notar que no se consideran las no linealidades propias del movimiento particular de una pierna obviando la dinámica real del sistema. Sin embargo, ofrecen una solución interesante al respecto.

- **Control de un sistema de doble péndulo no afín mediante inversión dinámica y separación de la escala de tiempo (*Control of a Nonaffine Double-Pendulum System via Dynamic Inversion and Time-Scale Separation*)**, se trata de un péndulo doble con entrada no afín, es decir no lineal, que tiene como propósito de control seguir una trayectoria [7]. En cuanto al sistema que utilizan en este trabajo, éste usa dos torques, uno colocado en la base del primer péndulo y el segundo en la juntura de ambos péndulos. Estos dos actuadores a diferencia de nuestro modelo de *Sistema de péndulo doble* de ésta tesis marca una diferencia en cuanto a su estrategia de control ya que tienen una señal de control adicional a nuestro modelo, es decir U_1 y U_2 .

Al respecto, la estrategia de control usada fue propuesta en el año 2006 [8] y fue usada de forma académica por los autores usando un *Sistema de péndulo doble*. Este método se basa en la separación temporal entre la dinámica del sistema y la del controlador para lograr el seguimiento a la trayectoria requerida. Cabe señalar que esta estrategia de control requiere de un sistema lineal por lo que el sistema fue linealizado. Además, se consideraron perturbaciones en la señal de control como también fue probado en la presente tesis teniendo buenos resultados como los nuestros.

- **Síntesis de un control óptimo para la oscilación y la amortiguación de un péndulo doble** (*The optimal control synthesis of the swinging and damping of a double pendulum*), intenta solucionar el problema de balanceo/oscilación de un *Sistema de péndulo doble* [9]. En cuanto a la descripción del sistema a diferencia de nuestro *Sistema de péndulo doble*, donde usamos el ángulo de giro y la velocidad angular. En éste trabajo se hace un cambio, es decir se usa el momento angular y no la velocidad angular. Ya que como se sabe, el elemento derivativo (velocidad angular) proveniente del ángulo de giro genera una gran dificultad para la aplicación de un control óptimo. El autor de este trabajo elimina por lo tanto este término durante el modelamiento. Como se mencionó antes, el modelamiento *Sistema de péndulo doble* que se llevará a cabo a diferencia de este trabajo en mención, sí considera la velocidad angular como subproducto del ángulo de giro, es decir su derivada natural.

Tal como se menciona en el título de este trabajo, se usa un control óptimo usado para solucionar el ratio de cambio del ángulo entre las diferentes secciones es decir el balanceo/oscilación de los ángulos. Para eso, el autor usa del principio máximo Pontryagin. Además como se mencionó anteriormente, al usar el Momento Angular, se elimina sólo la segunda derivada pero se mantiene siempre la primera derivada de la variable de control resolviendo de una forma más sencilla el problema óptimo de control.

- **Estabilización de un péndulo doble invertido con un torque de control** (*On stabilization of an inverted double pendulum with one control torque*), a diferencia del trabajo anterior, este trabajo posee sólo un controlador ubicado en la base. Es decir, sólo del primer péndulo tiene un control establecido así como el modelo del *Sistema de péndulo doble* de esta presenta tesis. Por consiguiente, el segundo péndulo queda libre y conectado por una juntura al primer péndulo [10] presentando un movimiento no periódico.

El modelo también se linealiza como casi todos los trabajos antes mencionados a fin de realizar su control y estabilización. Para eso se asume que los péndulos solo trabajan en un rango pequeño de giro de modo que puede aproximarse a un valor específicos. En cuanto al controlador lineal usado, se trata de una ganancia por retroalimentación. Precisamente, este análisis es parte de nuestra primera etapa de estudio de la

presente tesis teniendo en cuenta que el modelo es muy parecido.

- **Control de un robot humanoide andante con modificación en tiempo real de las posiciones de los pies (*Laufen humanoider Roboter auf regelungstechnischer Basis mit Echtzeitmodifikation der Fußpositionen*),** es otro trabajo relacionado al control del un péndulo doble por medio de un control óptimo cuyo objetivo es controlar la locomoción de un robot humanoide. Este robot humanoide utiliza dos piernas articuladas [11] las cuales se busca, según el autor, que las posiciones de los pies y las piernas sean precisas como la que generaría un humano. Este trabajo se diferencia asimismo con esta presente tesis en la orientación de las coordenadas para su modelamiento que lo convierte en un péndulo doble invertido.

Los controles utilizados para este trabajo fueron el PC (Preview control) y el MPC (Model Predictive Model) pero en su estado lineal. Para eso, se utiliza un proceso de linealización del sistema; lo que limita el trabajo a sólo ciertos puntos de operación. Tal como se explica en este trabajo, el autor de este trabajo simplificó la no linealidad de su sistema haciendo que los pasos que da el robot articulado sean pequeños; de modo que su estabilidad no sea un problema. Por otro lado, se consideraron también las posibles perturbaciones o obstáculos que el robot, en su movimiento, encuentren en su camino, de modo que el robot humanoide pueda ser capaz de sobrepasar con éxito estos obstáculos.

Como se vieron en los proyectos antes mencionados, todos los modelos utilizan un *Sistema de péndulo doble* en su variante inversa y no inversa. Se podría mencionar que los modelos del *Sistema de péndulo doble* de algunos de estos trabajos varíen en su modelamiento ya sea por el cambio de coordenadas o que se asumen aproximaciones en los ángulos de giro para su simplificarlo. Aún así con estas consideraciones, estos trabajos son validos como materia de estudios en el marco de su aplicación partiendo del hecho que los sistemas utilizan alguna dinámica de un péndulo doble y utilizan estrategias de control con buenos resultados a fin para resolver estos tipos de sistemas caóticos. Ahora bien, cabe mencionar que casi todos trabajos presentan una modelización de los sistemas en su forma lineal. Es decir, el control de los *Sistemas de péndulo doble* se realizó basado en un sistemas linealizados; de modo que los controladores usados son de naturaleza lineal dando como resultado que los

puntos de operación o trabajo sean cortos y muy limitados que no abarquen todo el espectro de la dinámica de los reales *Sistemas de péndulo doble* como si lo haría un sistema no lineal y que sí es parte del estudio de esta tesis como se verá más adelante.

Por otro lado, la motivación que presentan muchos de estos proyectos en materia de su aplicación va más allá de la parte académica. En efecto, muchos de ellos toman en cuenta la aplicación humana para el bien y beneficio de la calidad de vida de personas. Por ejemplo, en el primer trabajo presentado en este capítulo, el autor menciona que este tipo de trabajo va a mejorar la calidad de vida de personas con movilidad reducida dando a entender ésta como su motivación específica en mención. El segundo trabajo, por otro lado, tiene como propósito y motivación el control de la locomoción de una pierna a fin de usarlo en una prótesis robótica que en un futuro pueda reemplazar la pérdida de este miembro en alguna persona en necesidad. El tercer, cuarto y quinto trabajo presentan un estudio puramente académico y aplicativo para un *Sistema de péndulo doble*. No cabe ninguna duda que su aplicación vaya más del mero ámbito de la investigación sino también para un beneficio mas altruista. En cuanto al último trabajo mencionado, el autor explica que su trabajo, a modo de aplicación, puede ayudar a proteger de alguna manera a las personas en ambientes peligrosos o no apropiados para ellos ya que se utilizaría un robot humanoide que pueda evitar obstáculos que un simple humano no lo lograría con efectividad. De los antes visto, se concluye y deja por sentado que el estudio del *Sistema de péndulo doble* tiene un gran valor ya que su aplicación va mas allá de su parte académica ya que también puede y muy ciertamente va a mejorar la calidad de vida de las personas.

1.3. Problemática de control del *Sistema de péndulo doble*

Los sistemas basados en un *Sistema de péndulo doble* presentan dificultad para su control y su estabilización. El movimiento caótico que produce el segundo péndulo se hace resaltar enormemente a diferencia del movimiento circular que se produce por el primer péndulo. Además del movimiento caótico ya mencionado, se presentan otros problemas y son los siguientes:

1.3.1 No linealidad del *Sistema de péndulo doble*

El *Sistema de péndulo doble* presenta no linealidades ya que en su modelamiento, se encuentran términos cuadráticos o funciones de segundo grado con términos de cosenos y senos que hacen difícil su solución matemática; para dar por un ejemplo de ello, al cambio mínimo en estos ángulos pueden considerarse quasi lineales pero al ser de valores más grandes como condición inicial, pueden generar recorridos muchos más caóticos y analíticamente difíciles de manejar, y por ende difícil de controlar. Si bien es cierto que este tipo de sistema es parte de muchos estudios académicos y muchos estudios han abordado la solución de la misma desde un punto de vista lineal aproximando los ángulos a valores pequeños, como por ejemplo, resultando es un sistema con ecuaciones puramente lineales. Sin embargo, al linearizar el *Sistema de péndulo doble* se deja de lado la dificultad no lineal del sistema y su verdadera dinámica ya que el sistema linealizado sólo será válido para un pequeño espacio de ángulo de giro; de modo que se obvia su verdadero comportamiento difiriendo bastante a uno real [12].

1.3.2 Oscilaciones y dificultad para la estabilización del *Sistema de péndulo doble*

En efecto, además del comportamiento caótico antes descrito debemos añadir las oscilaciones y la dificultad de estabilización que tiene un *Sistema de péndulo doble*. Debemos entender primeramente que al tener dos ángulos de giros, tenemos por lo tanto dos grados de libertad que al ser posicionados en un punto específico de equilibrio, debería existir dos controladores que gobiernen dichos ángulos de giro a fin de estabilizar eficientemente el *Sistema de péndulo doble* [13]. Sin embargo, cabe señalar que el problema radica en que la mayoría de *Sistemas de péndulo doble* e incluyendo el nuestro, posee más grados de libertad que entradas de control. En efecto, nuestro *Sistema de péndulo doble* si bien posee 2 ángulos de giro, es decir 2 grados de libertad o parámetros que nos ayudan a describir las demás variables, sólo depende de una única señal de entrada control ubicada en el pivote del *Sistema de péndulo doble*. Este tiene como tarea de estabilizar y posicionar los dos péndulos en un determinado punto de equilibrio y mantenerlos allí. Este hace por consiguiente que sea su control muy desafiante al momento de elegir una estrategia de control ya que éste tiene que ser no sólo robusto sino también capaz de

vencer las oscilaciones [14]. Adicionalmente a la dificultad antes mencionada debemos describir otra la problemática, las oscilaciones que en su mayoría pueden originarse en el actuador que gobierna la entrada de control, es decir el motor que genera el momento de torque al todo el *Sistema de péndulo doble* por medio de su desgaste en el tiempo y la fricción entre las superficies y rotores del motor que se traduce en oscilaciones y/o perturbaciones. Estos que se modelan como un ruido continuo que afecta al buen funcionamiento del sistema en su totalidad o parcialmente que la estrategia de control sea capaz de disminuir a su mínima expresión a fin de tener un buen control óptimo y robusto; empero como se mencionó antes, es un tema bastante complicado de solucionar.

1.3.3 Dificultad para el seguimiento específico de la trayectoria del *Sistema de péndulo doble*

Según la literatura consultada, los tipos de problemas que se plantean para el *Sistema de péndulo doble* no contemplan un punto de referencia a donde el péndulo doble tendría que direccionarse, en muchos casos solo se plantea que el sistema llegue a un punto de equilibrio estable. Sin embargo, si se le agrega puntos de referencia diferentes a los puntos de equilibrio estables, la dificultad aumenta ya que el nuevo objetivo es que la estrategia de control sea robusto y capaz de mantener estos puntos de referencia diferentes a los de equilibrio y mantenerlos allí [15]. Esto incurre es un problema adicional a la dificultad ya establecida previamente en los puntos anteriores que no es considerado en muchos de los trabajos.

1.4. Propuesta de solución

Para dar un panorama amplio sobre la solución a las problemáticas asociadas al *Sistema de péndulo doble* antes mencionadas, se deben establecer éstos como puntos importantes a tomar en cuenta. En efecto, como primer punto se debe comenzar por considerar las no linealidades del *Sistema de péndulo doble*, como segundo punto se debe considerar también las oscilaciones propias que presenta el sistema, y los agentes externos que las producen y finalmente la trayectoria de seguimiento para valores de referencia arbitrarios más allá de los puntos de equilibrio.

1.4.1 Modelamiento No Lineal del *Sistema de péndulo doble*

Nuestra propuesta se basa en trabajar con un modelo de péndulo doble no lineal que verdaderamente represente su dinámica real; entonces para modelar el sistema, se tiene dos alternativas ya que podemos plantearlo desde un punto de vista energético o desde un punto de vista basado en fuerzas que soportan los péndulos. La diferencia de uno contra el otro es su simplicidad o complejidad para describir las ecuaciones matemáticas del modelo que gobiernan al *Sistema de péndulo doble* respectivamente. En efecto, en esta tesis analizaremos el sistema desde el punto de vista energético debido a su estandarización y facilidad de cálculo ya que sólo se hará uso de algunas derivadas (método Lagrangiano) [16] y no el cálculo de fuerzas que gobierna al sistema (método de Newton) [17].

El sistema por lo tanto, será descrito por ecuaciones diferenciales ordinarias o ODEs (Ordinary Differential Equations) de segundo grado, los que considerarán la No Linealidad propia del *Sistema de péndulo doble* en todo su rango de acción. Por consiguiente, su respectiva solución será dada, considerando todo el espectro del control No lineal, a base de métodos de integración que puede ser bastante compleja por la solución numérica que se requiere y asimismo computacionalmente costosa. De allí su importancia de resolver un verdadero/real *Sistema de péndulo doble* puramente No Lineal cobra mucha importancia [18] ya que no muchos autores consideran esta no linealidad y optan por linealizarlo a fin de obviar los puntos antes mencionados.

1.4.2 Estrategia de control no lineal

Comenzaremos con una estrategia de control bastante exitosa para la solución de este sistema no lineal mediante un modelo linealizado. Se puede mencionar al Control Lineal Cuadrático (LQR) como un método bastante exitoso para su control, estudiado y utilizado mucho en la parte académica [19]. El LQR usa un modelo lineal para predecir el comportamiento del sistema, el cual pueda representar su dinámica en un cierto rango de operación. Sin embargo, su control y estudio se limita de gran manera ya que no se cubre la no lineal propia del *Sistema de péndulo doble* ni las restricciones que puedan considerarse tanto en las variables de control o en la misma señal de control. Al usar el LQR, el *Sistema de péndulo doble* es resuelto como un problema óptimo de control (OCP); ya que la solución de la misma, requerirá la búsqueda de una entrada

de control óptima, tal que se pueda alcanzar un costo mínimo de energía o que los valores de posición de los péndulos converjan a un punto de referencia o trayectoria deseada en el menor tiempo posible. La solución de ésta OCP mediante el LQR deberá utilizar necesariamente un modelo lineal por lo que para su pleno uso, se deberá linealizar el sistema en un punto específico de operación por algún método de linealización como la expansión de la serie de Taylor de las ODEs considerando un punto específico de operación [20], el cual presentaría, como se mencionó anteriormente, limitaciones.

Además de la linealización propia del sistema para así usar el LQR, tenemos que entender que este control óptimo se basa en un modelo determinístico donde éste predice su comportamiento a futuro en un espacio de tiempo determinado (Horizonte finito) . Por lo tanto, este control óptimo no contempla perturbaciones o mismatch de los modelos. De modo que a fin de tener aún mejores resultados considerando las limitaciones previamente mencionadas, el LQR debe combinar un control a lazo cerrado y/o un Observador completo o reducido que prediga las variables no medibles [21] como se probará en la presente tesis.

Por esa razón, se presentará en esta tesis un análisis de las estrategias de control óptimas no lineales para un *Sistema de péndulo doble* considerando restricciones, robustez contra el mismatch y perturbaciones, y un horizonte de predicción infinito que garantice estabilidad del sistema [22]. Así, se propone como solución al NMPC (Nonlinear Model Predictive Control) considerándolo como un Receding Horizon Control (RHC) que construye una función de control que predice el comportamiento del sistema con un horizonte de predicción indeterminado (Infinite Horizon) o que lo aproxime al mismo a fin de garantizar una buena optimización y de posicionamiento en el punto de referencia deseado como por ejemplo mantener el *Sistema de péndulo doble* en forma vertical a lo largo del tiempo, consumiendo la mínima cantidad de energía [23] a lo largo de todo el rango de acción del *Sistema de péndulo doble* sin limitaciones alguna.

Ahora bien, esta estrategia de control no lineal tiene que ser comparada a fin de asegurar su efectividad frente a las otras estrategias de control que ya tuvieron mucho éxito para la solución de este tipo de sistema. Sin embargo, precisamente estas estrategias de control manejan un modelo lineal del sistema por lo que se tiene que idear un punto en común entre ella. En efecto, todas las estrategias de control lineal junto con la estrategias de control no

lineal pueden ser solo probadas en un punto común de operación. Para eso se linealizará el *Sistema de péndulo doble* y se controlará el sistema lineal para un LQR, Observador completo con Controlador, Observador Reducido con Controlador y un MPC (Model Predictive Controller). En cuando a la estrategia de control no lineal, el NMPC no será linealizado, sino que su rango de giro del sistema de péndulo doble se limitará a un punto inicial y punto final, tal que se asemeje a sus pares lineales, pero manteniendo la dinámica no lineal del *Sistema de péndulo doble* en su totalidad.

1.5. Objetivos

Diseñar diferentes estrategias de control basados en el método Lagrangiano de energía y probar cuál de ellos es la que mejor performance ofrece en la solución del problema óptimo de control (OCP) considerando las restricciones, el menor consumo de energía y el tiempo de establecimiento al punto de equilibrio.

Los objetivos específicos de este trabajo son:

- Modelar y simular el *Sistema de péndulo doble* basado en energías (Método Lagrangiano)
- Linealizar el modelo en distintos puntos de operación y obtener sus respectivos espacios de estado para posteriormente analizar la observabilidad y controlabilidad.
- Diseñar estrategias de control para estabilizar el sistema lineal (LQR, Observador y MPC) y no lineal de péndulo doble (NMPC).
- Realizar las simulaciones a fin de verificar el performance de las estrategias de control comparando los resultados de las estrategias de control.
- Probar la inherente robustez del NMPC frente a incertidumbres.

Capítulo 2

MODELAMIENTO DEL SISTEMA DE PÉNDULO DOBLE

El *Sistema de péndulo doble* se modelará por medio de diferentes tipos de modelado, como por ejemplo, el modelado físico o el modelado por bloques en Simulink entre otros. Los cuales serán descritos en detalle en este capítulo. En primer lugar como se mencionó anteriormente, se desarrolla el modelamiento físico del *Sistema de péndulo doble*. Concretamente, se deduce las ecuaciones del movimiento del péndulo doble mediante el método Lagrangiano para luego simularlo y obtener las ecuaciones que describan el sistema. En segundo lugar, se obtiene el modelo del sistema por medio de MATLAB/SIMULINK, para ello se emplea las funciones de bloque (*Fcn-Blocks*) y funciones de MATLAB (*Script* y *S-Function*). Y por último, se simula los diferentes modelos obtenidos; para eso, se considera condiciones iniciales con y sin excitación a fin de comparar el comportamiento de los diferentes modelos obtenidos.

2.1. Estudio del *Sistema de Péndulo doble*

La representación del *Sistema de Péndulo doble* [1] con las variables que se utilizarán en el modelado se puede ver en la figura 2.1.

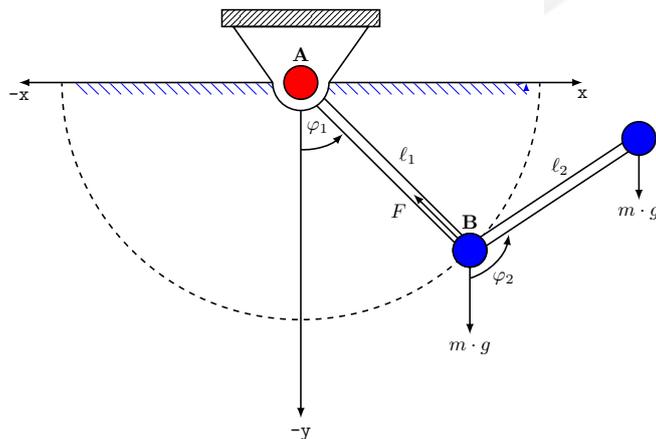


Figura 2.1: Representación de un *Sistema de péndulo doble*.
Fuente : Elaboración propia

En el contexto de esta tesis, el primer péndulo está fijo en el punto A y posee un rodamiento (*pivote*), que es excitado por un motor. En cuanto al segundo péndulo, éste está conectado con la parte final del primer péndulo y posee la misma característica del primer rodamiento pero sin motor. Es importante notar dos consideraciones, los rodamientos están influenciados por fricciones respectivamente y el flujo de aire será omitido para el modelamiento del sistema. Las longitudes del péndulo que serán parte del modelamiento, son las distancias entre el centro de gravedad (*punto medio*) y el rodamiento de los pivotes para cada uno de los péndulos. El movimiento de los péndulos está dado por los ángulos $\varphi_1(t)$ y $\varphi_2(t)$.

Entonces, procedemos a asignar las variables de entrada y salida. El torque $M(t)$ (a través de un motor) en el punto A actúa como entrada y los ángulos $\varphi_1(t)$ y $\varphi_2(t)$ se consideran como salida. Aquí es importante tener en cuenta que la dirección positiva de los ángulos $\varphi_1(t)$, $\varphi_2(t)$ y del torque $M(t)$ son en sentido contrario a las agujas del reloj. .

Las variables y unidades necesarias para el modelamiento físico o la obtención de la ecuación diferencial del *Sistema de péndulo doble* se pueden ver en la tabla 2.1 [1].

2.2. Modelamiento Físico

2.2.1 Obtención analítica de las ecuaciones de LAGRANGE

En forma general, la modelización física de cualquier sistema se describe en forma matemática con mayor frecuencia por ecuaciones diferenciales lineales ordinarias así como parciales o en su defecto en ecuaciones diferenciales no lineales [24]. En cuanto a la construcción de las ecuaciones diferenciales, existen varios métodos a emplear y en esta tesis se utilizarán las ecuaciones de LAGRANGE. Como se mencionó anterior en marco de esta tesis, las ecuaciones de movimiento no se obtienen a través el principio de d'Alembert (*Generalización de la segunda ley de Newton*), ya que conlleva a considerar el modelamiento de fuerzas [25] que no son necesarias para el movimiento del péndulo. En cambio, se emplea las ecuaciones de Lagrange para las ecuaciones de movimiento [25]. Los n grados de libertad de un sistema se especifica por la n coordenadas generalizadas [26], Los ángulos $\varphi_1(t)$ y $\varphi_2(t)$ se representan

Tabla 2.1: Variable y unidades para el modelado físico del *Sistema de péndulo doble*.

Variable	Designación	Valor/Unidad
$x_1(t), x_2(t), y_1(t), y_2(t)$	Posición	m
$M(t)$	Momento (Torque)	Nm
$M_{R1}(t)$	Torque por fricción del rodamiento del motor de accionamiento $M_{R1}(t) = R_{P1}\dot{\varphi}_1$	Nm
$M_{R2}(t)$	Torque por fricción del rodamiento del péndulo $M_{R2}(t) = R_{P2}(\dot{\varphi}_2 - \dot{\varphi}_1)$	Nm
$\varphi_1(t)$	Ángulo del primer péndulo	rad
$\varphi_2(t)$	Ángulo del segundo péndulo	rad
$J_1(t)$	Momento de Inercia del primer péndulo en relación con el eje del punto medio $l_1/2$	kg m ²
$J_2(t)$	Momento de Inercia del primer péndulo en relación con el eje del punto medio $l_2/2$	kg m ²
l_1	Longitud del primer péndulo	0,2 m
l_2	Longitud del segundo péndulo	0,2 m
g	Aceleración de gravedad	9,81 m/s ²
m_1	Masa del primer péndulo	0,3 kg
m_2	Masa del segundo péndulo	0,3 kg
$R_{P1}(t)$	Coefficiente de fricción	$1 \cdot 10^{-2} \frac{Nm}{rad \cdot s^{-1}}$
$R_{P2}(t)$	Coefficiente de fricción	$1 \cdot 10^{-3} \frac{Nm}{rad \cdot s^{-1}}$

con las coordenadas generalizadas:

$$q_1 = \varphi_1 \quad \text{und} \quad q_2 = \varphi_2 \quad (2.1)$$

Como se puede notar en el punto 2.1, existen 2 grados de libertad representados por los ángulos que gobiernan los péndulos.

Las posiciones del péndulo (en el centro de gravedad) se describen a continuación:

$$x_1 = \frac{l_1}{2} \sin(\varphi_1) \quad y_1 = -\frac{l_1}{2} \cos(\varphi_1) \quad (2.2)$$

$$x_2 = \frac{l_2}{2} \sin(\varphi_2) + l_1 \sin(\varphi_1) \quad y_2 = -\frac{l_2}{2} \cos(\varphi_2) - l_1 \cos(\varphi_1) \quad (2.3)$$

La derivación con respecto al tiempo

$$\dot{x}_1 = \frac{l_1}{2} \cdot \cos(\varphi_1) \cdot \dot{\varphi}_1 \quad \dot{y}_1 = \frac{l_1}{2} \cdot \sin(\varphi_1) \cdot \dot{\varphi}_1 \quad (2.4)$$

$$\dot{x}_2 = \frac{l_2}{2} \cdot \cos(\varphi_2) \cdot \dot{\varphi}_2 + l_1 \cdot \cos(\varphi_1) \cdot \dot{\varphi}_1 \quad \dot{y}_2 = \frac{l_2}{2} \cdot \sin(\varphi_2) \cdot \dot{\varphi}_2 + l_1 \cdot \sin(\varphi_1) \cdot \dot{\varphi}_1 \quad (2.5)$$

Ahora se determina la energía cinética T, la cual se obtiene por medio de la suma de las ecuaciones de movimiento que gobierna cada péndulo en las 2 dimensiones para la traslación (x - y) y rotación (φ).

$$T = \sum_{i=1}^2 \left(\frac{m_i}{2} (\dot{x}_i^2 + \dot{y}_i^2) + \frac{J_i}{2} \left(\sum_{j=1}^i \dot{\varphi}_j \right)^2 \right) \quad (2.6)$$

A continuación se acomoda y simplifica la ecuación de la siguiente manera:

$$T = \frac{1}{2} \cdot m_1 \cdot v_1^2 + \frac{1}{2} \cdot J_1 \cdot w_1^2 + \frac{1}{2} \cdot m_2 \cdot v_2^2 + \frac{1}{2} \cdot J_2 \cdot w_2^2 \quad (2.7)$$

Con:

$$v_1^2 = \dot{x}_1^2 + \dot{y}_1^2 \quad v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 \quad (2.8)$$

$$w_1^2 = \dot{\varphi}_1^2 \quad w_2^2 = \dot{\varphi}_2^2 \quad (2.9)$$

$$J_1 = \frac{1}{12} \cdot m_1 \cdot l_1^2 \quad J_2 = \frac{1}{12} \cdot m_2 \cdot l_2^2 \quad (2.10)$$

De la misma manera se obtiene la energía potencial U de los péndulos por medio de la suma de la energía.

$$U = \int_0^{y_i} m_i \cdot g \, dy = m_i \cdot g \cdot y_i \quad (2.11)$$

De allí se obtiene la energía potencia de cada péndulo por medio de la integral y queda de la siguiente manera:

$$U = m_1 \cdot g \cdot y_1 + m_2 \cdot g \cdot y_2 \quad (2.12)$$

Definimos por la tanto la función de LAGRANGE como la diferencia de la energía cinética y la energía potencial:

$$L = T - U \quad (2.13)$$

Y usamos ahora la función de LAGRANGE como base para las ecuaciones de Lagrange de segundo tipo para el ángulo de giro y su respectiva derivada, la velocidad angular para cada péndulo:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}_1} \right) - \frac{\partial L}{\partial \varphi_1} = M - M_{R1} \quad \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}_2} \right) - \frac{\partial L}{\partial \varphi_2} = -M_{R2} \quad (2.14)$$

Se procede ahora a ordenar los términos que corresponden a la ecuación 2.13. Entonces ordenamos los sumandos:

$$\begin{aligned} \frac{1}{2} \cdot m_1 \cdot v_1^2 &= \frac{1}{2} \cdot m_1 \cdot (\dot{x}_1^2 + \dot{y}_1^2) \\ &= \frac{1}{2} \cdot m_1 \cdot \left(\frac{l_1}{2} \right)^2 \cdot \dot{\varphi}_1^2 \cdot \cos^2(\varphi_1) + \frac{1}{2} \cdot m_1 \cdot \left(\frac{l_1}{2} \right)^2 \cdot \dot{\varphi}_1^2 \cdot \sin^2(\varphi_1) \\ &= \frac{1}{2} \cdot m_1 \cdot \left(\frac{l_1}{2} \right)^2 \cdot \dot{\varphi}_1^2 \end{aligned} \quad (2.15)$$

$$\begin{aligned} \frac{1}{2} \cdot m_2 \cdot v_2^2 &= \frac{1}{2} \cdot m_2 \cdot (\dot{x}_2^2 + \dot{y}_2^2) \\ &= \frac{1}{2} \cdot m_2 \cdot [l_1^2 \cdot \dot{\varphi}_1^2 + \left(\frac{l_2}{2} \right)^2 \cdot \dot{\varphi}_2^2 + 2 \cdot l_1 \cdot \frac{l_2}{2} \cdot \dot{\varphi}_1 \cdot \dot{\varphi}_2 \cdot (\sin\varphi_1 \cdot \sin\varphi_2 + \cos\varphi_1 \cdot \cos\varphi_2)] \\ &= \frac{1}{2} \cdot m_2 \cdot [l_1^2 \cdot \dot{\varphi}_1^2 + \left(\frac{l_2}{2} \right)^2 \cdot \dot{\varphi}_2^2 + 2 \cdot l_1 \cdot \frac{l_2}{2} \cdot \dot{\varphi}_1 \cdot \dot{\varphi}_2 \cdot \cos(\varphi_1 - \varphi_2)] \end{aligned} \quad (2.16)$$

La ecuación de LAGRANGE sería finalmente:

$$\begin{aligned} L &= T - U \\ &= \frac{1}{2} \cdot m_1 \cdot \left(\frac{l_1}{2} \right)^2 \cdot \dot{\varphi}_1^2 + \frac{1}{2} \cdot J_1 \cdot \dot{\varphi}_1^2 \\ &+ \frac{1}{2} \cdot m_2 \cdot [l_1^2 \cdot \dot{\varphi}_1^2 + \left(\frac{l_2}{2} \right)^2 \cdot \dot{\varphi}_2^2 + 2 \cdot l_1 \cdot \frac{l_2}{2} \cdot \dot{\varphi}_1 \cdot \dot{\varphi}_2 \cdot \cos(\varphi_1 - \varphi_2)] \\ &+ \frac{1}{2} \cdot J_2 \cdot \dot{\varphi}_2^2 + m_1 \cdot g \cdot \frac{l_1}{2} \cdot \cos\varphi_1 + m_2 \cdot g \cdot \left(l_1 \cdot \cos\varphi_1 + \frac{l_2}{2} \cdot \cos\varphi_2 \right) \end{aligned} \quad (2.17)$$

Las ecuaciones de movimiento 2.14 obtenemos los términos de cada velocidad de ángulo φ_1 y φ_2 . Empezamos con la derivada de la función de LAGRANGE con respecto a la velocidad de ángulo φ_1 y su derivada en el tiempo:

$$\frac{\partial L}{\partial \dot{\varphi}_1} = m_1 \cdot \left(\frac{l_1}{2} \right)^2 \cdot \dot{\varphi}_1 + J_1 \cdot \dot{\varphi}_1 + m_2 \cdot [l_1^2 \cdot \dot{\varphi}_1 + l_1 \cdot \frac{l_2}{2} \cdot \dot{\varphi}_2 \cdot \cos(\varphi_1 - \varphi_2)] \quad (2.18)$$

y la segunda derivada en el tiempo sería:

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}_1} \right) &= m_1 \cdot \left(\frac{l_1}{2} \right)^2 \cdot \ddot{\varphi}_1 + J_1 \cdot \ddot{\varphi}_1 + m_2 \cdot l_1^2 \cdot \ddot{\varphi}_1 + m_2 \cdot l_1 \cdot \left(\frac{l_2}{2} \right) \cdot \ddot{\varphi}_2 \cdot \cos(\varphi_1 - \varphi_2) \\ &- m_2 \cdot l_1 \cdot \left(\frac{l_2}{2} \right) \cdot \dot{\varphi}_2 \cdot \text{sen}(\varphi_1 - \varphi_2) \cdot (\dot{\varphi}_1 - \dot{\varphi}_2) \end{aligned} \quad (2.19)$$

Y el segundo termino de la primera ecuación 2.14:

$$\frac{\partial L}{\partial \varphi_1} = -m_2 \cdot l_1 \cdot \left(\frac{l_2}{2}\right) \cdot \dot{\varphi}_1 \cdot \dot{\varphi}_2 \cdot \sin(\varphi_1 - \varphi_2) - m_1 \cdot g \cdot \frac{l_1}{2} \cdot \sin \varphi_1 - m_2 \cdot g \cdot l_1 \cdot \sin \varphi_1 \quad (2.20)$$

Ahora procedemos a obtener la derivada de la función de LAGRANGE con respecto a la velocidad de ángulo $\dot{\varphi}_2$ y su derivada en el tiempo:

$$\frac{\partial L}{\partial \dot{\varphi}_2} = m_2 \cdot \left(\frac{l_2}{2}\right)^2 \cdot \dot{\varphi}_2 + J_2 \cdot \dot{\varphi}_2 + m_2 \cdot l_1 \cdot \left(\frac{l_2}{2}\right) \cdot \dot{\varphi}_1 \cdot \cos(\varphi_1 - \varphi_2) \quad (2.21)$$

y la segunda derivada en el tiempo sería:

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}_2} \right) &= m_2 \cdot \left(\frac{l_2}{2}\right)^2 \cdot \ddot{\varphi}_2 + J_2 \cdot \ddot{\varphi}_2 + m_2 \cdot l_1 \cdot \left(\frac{l_2}{2}\right) \cdot \ddot{\varphi}_1 \cdot \cos(\varphi_1 - \varphi_2) \\ &\quad - m_2 \cdot l_1 \cdot \left(\frac{l_2}{2}\right) \cdot \dot{\varphi}_1 \cdot \sin(\varphi_1 - \varphi_2) \cdot (\dot{\varphi}_1 - \dot{\varphi}_2) \end{aligned} \quad (2.22)$$

Y el segundo termino de la segunda ecuación 2.14:

$$\frac{\partial L}{\partial \varphi_2} = m_2 \cdot l_1 \cdot \left(\frac{l_2}{2}\right) \cdot \dot{\varphi}_1 \cdot \dot{\varphi}_2 \cdot \sin(\varphi_1 - \varphi_2) - m_2 \cdot g \cdot \left(\frac{l_2}{2}\right) \cdot \sin \varphi_2 \quad (2.23)$$

Entonces se forman las ecuaciones de cada ecuación de movimiento. Se toman las ecuaciones intermedias 2.19 y 2.20 así como las ecuaciones 2.22 y 2.23:

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}_1} \right) - \frac{\partial L}{\partial \varphi_1} &= (m_1 \cdot \left(\frac{l_1}{2}\right)^2 + J_1 + m_2 \cdot l_1^2) \cdot \ddot{\varphi}_1 + (m_2 \cdot l_1 \cdot \left(\frac{l_2}{2}\right) \cdot \cos(\varphi_1 - \varphi_2)) \cdot \ddot{\varphi}_2 \\ &\quad + m_2 \cdot l_1 \cdot \left(\frac{l_2}{2}\right) \cdot \sin(\varphi_1 - \varphi_2) \cdot \dot{\varphi}_2^2 + (m_1 \cdot \left(\frac{l_1}{2}\right) + m_2 \cdot l_1) \cdot g \cdot \sin \varphi_1 \end{aligned} \quad (2.24)$$

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}_2} \right) - \frac{\partial L}{\partial \varphi_2} &= (m_2 \cdot l_1 \cdot \left(\frac{l_2}{2}\right) \cdot \cos(\varphi_1 - \varphi_2)) \cdot \ddot{\varphi}_1 + (m_2 \cdot \left(\frac{l_2}{2}\right)^2 + J_2) \cdot \ddot{\varphi}_2 \\ &\quad - m_2 \cdot l_1 \cdot \left(\frac{l_2}{2}\right) \cdot \sin(\varphi_1 - \varphi_2) \cdot \dot{\varphi}_1^2 + m_2 \cdot \left(\frac{l_2}{2}\right) \cdot g \cdot \sin \varphi_2 \end{aligned} \quad (2.25)$$

Para una mejor visión de las ecuaciones de movimiento se procede a hacer asignar coeficientes:

$$a_{11} = m_1 \cdot \left(\frac{l_1}{2}\right)^2 + J_1 + m_2 \cdot l_1^2 \quad (2.26)$$

$$a_{12} = m_2 \cdot l_1 \cdot \left(\frac{l_2}{2}\right) \quad (2.27)$$

$$a_{13} = [m_1 \cdot \left(\frac{l_1}{2}\right) + m_2 \cdot l_1] \cdot g \quad (2.28)$$

$$a_{21} = m_2 \cdot \left(\frac{l_2}{2}\right)^2 + J_2 \quad (2.29)$$

$$a_{22} = m_2 \cdot \left(\frac{l_2}{2}\right) \cdot g \quad (2.30)$$

Obteniendo la ecuación de movimiento 1:

$$a_{11} \cdot \ddot{\varphi}_1 + a_{12} \cdot \cos(\varphi_1 - \varphi_2) \cdot \ddot{\varphi}_2 + a_{12} \cdot \sin(\varphi_1 - \varphi_2) \cdot \dot{\varphi}_2^2 + a_{13} \cdot \sin(\varphi_1) = M - M_{R1} \quad (2.31)$$

Y la ecuación de movimiento 2:

$$a_{12} \cdot \cos(\varphi_1 - \varphi_2) \cdot \ddot{\varphi}_1 + a_{21} \cdot \ddot{\varphi}_2 - a_{12} \cdot \sin(\varphi_1 - \varphi_2) \cdot \dot{\varphi}_1^2 + a_{22} \cdot \sin(\varphi_2) = -M_{R2} \quad (2.32)$$

2.2.2 Acoplamiento

Como se puede apreciar, existe un acoplamiento en el sistema ya que las variables de estado depende uno del otro como se pueden apreciar en las ecuaciones 2.33 y 2.34 respectivamente. Por lo tanto, acomodamos las ecuaciones con respecto a las derivadas de mayor orden respectivamente para poder apreciar mejor este caso. $M_{R1} = R_{P1} \cdot \dot{\varphi}_1$ y $M_{R2} = R_{P2} \cdot (\dot{\varphi}_2 - \dot{\varphi}_1)$

$$\ddot{\varphi}_1 = -\frac{a_{12}}{a_{11}} \cdot \cos(\varphi_1 - \varphi_2) \cdot \ddot{\varphi}_2 - \frac{a_{12}}{a_{11}} \cdot \sin(\varphi_1 - \varphi_2) \cdot \dot{\varphi}_2^2 - \frac{a_{13}}{a_{11}} \cdot \sin(\varphi_1) + \frac{1}{a_{11}} \cdot M - \frac{1}{a_{11}} \cdot M_{R1} \quad (2.33)$$

$$\ddot{\varphi}_2 = -\frac{a_{12}}{a_{21}} \cdot \cos(\varphi_1 - \varphi_2) \cdot \ddot{\varphi}_1 + \frac{a_{12}}{a_{21}} \cdot \sin(\varphi_1 - \varphi_2) \cdot \dot{\varphi}_1^2 - \frac{a_{22}}{a_{21}} \cdot \sin(\varphi_2) - \frac{1}{a_{21}} \cdot M_{R2} \quad (2.34)$$

Ahora reemplazamos la ecuación 2.34 en la ecuación 2.33 :

$$\begin{aligned} \ddot{\varphi}_1 = & -\frac{a_{12}}{a_{11}} \cdot \cos(\varphi_1 - \varphi_2) \cdot \left[-\frac{a_{12}}{a_{21}} \cdot \cos(\varphi_1 - \varphi_2) \cdot \ddot{\varphi}_1 + \frac{a_{12}}{a_{21}} \cdot \sin(\varphi_1 - \varphi_2) \cdot \dot{\varphi}_1^2 \right. \\ & \left. - \frac{a_{22}}{a_{21}} \cdot \sin(\varphi_2) - \frac{1}{a_{21}} \cdot M_{R2} \right] - \frac{a_{12}}{a_{11}} \cdot \sin(\varphi_1 - \varphi_2) \cdot \dot{\varphi}_2^2 - \frac{a_{13}}{a_{11}} \cdot \sin(\varphi_1) \\ & + \frac{1}{a_{11}} \cdot M - \frac{1}{a_{11}} \cdot M_{R1} \end{aligned}$$

Donde obtenemos la ecuacion diferencial de segundo orden para el primer ángulo desacoplado:

$$\begin{aligned} \ddot{\varphi}_1 = & -\frac{a_{12}}{a_{11} \cdot a_{21}} \cdot \sin(\varphi_1 - \varphi_2) \cdot [a_{12} \cdot \cos(\varphi_1 - \varphi_2) \cdot \dot{\varphi}_1^2 + a_{21} \cdot \dot{\varphi}_2^2] \\ & + \frac{a_{12}}{a_{11} \cdot a_{21}} \cdot \cos(\varphi_1 - \varphi_2) \cdot [a_{22} \cdot \sin(\varphi_2) + R_{P2} \cdot (\dot{\varphi}_2 - \dot{\varphi}_1)] \\ & - \frac{a_{13}}{a_{11}} \cdot \sin(\varphi_1) + \frac{1}{a_{11}} \cdot M - \frac{1}{a_{11}} \cdot R_{P1} \cdot \dot{\varphi}_1 \end{aligned}$$

$$\frac{a_{11} \cdot a_{21} - a_{12}^2 \cdot \cos^2(\varphi_1 - \varphi_2)}{a_{11} \cdot a_{21}} \quad (2.35)$$

Ahora reemplazamos la ecuación 2.33 en la ecuación 2.34 :

$$\begin{aligned} \ddot{\varphi}_2 = & -\frac{a_{12}}{a_{21}} \cdot \cos(\varphi_1 - \varphi_2) \cdot \left[-\frac{a_{12}}{a_{11}} \cdot \cos(\varphi_1 - \varphi_2) \cdot \ddot{\varphi}_2 - \frac{a_{12}}{a_{11}} \cdot \sin(\varphi_1 - \varphi_2) \cdot \dot{\varphi}_2^2 \right. \\ & \left. - \frac{a_{13}}{a_{11}} \cdot \sin(\varphi_1) + \frac{1}{a_{11}} \cdot M - \frac{1}{a_{11}} \cdot M_{R1} \right] + \frac{a_{12}}{a_{21}} \cdot \sin(\varphi_1 - \varphi_2) \cdot \dot{\varphi}_1^2 - \frac{a_{22}}{a_{21}} \cdot \sin(\varphi_2) \\ & - \frac{1}{a_{21}} \cdot M_{R2} \end{aligned}$$

y finalmente obtenemos la ecuacion desacoplada para el segundo ángulo:

$$\begin{aligned} \ddot{\varphi}_2 = & \frac{a_{12}}{a_{11} \cdot a_{21}} \cdot \sin(\varphi_1 - \varphi_2) \cdot [a_{12} \cdot \cos(\varphi_1 - \varphi_2) \cdot \dot{\varphi}_2^2 + a_{11} \cdot \dot{\varphi}_1^2] \\ & + \frac{a_{12}}{a_{11} \cdot a_{21}} \cdot \cos(\varphi_1 - \varphi_2) \cdot [a_{13} \cdot \sin(\varphi_1) - M + R_{P1} \cdot \dot{\varphi}_1] - \frac{a_{22}}{a_{21}} \cdot \sin(\varphi_2) \\ & - \frac{1}{a_{21}} \cdot R_{P2} \cdot (\dot{\varphi}_2 - \dot{\varphi}_1) \end{aligned}$$

$$\frac{a_{11} \cdot a_{21} - a_{12}^2 \cdot \cos^2(\varphi_1 - \varphi_2)}{a_{11} \cdot a_{21}} \quad (2.36)$$

En la figura 2.2 se presenta representación del diagrama de bloques *Sistema de Péndulo doble*. Se nota claramente que el sistema es acoplado.

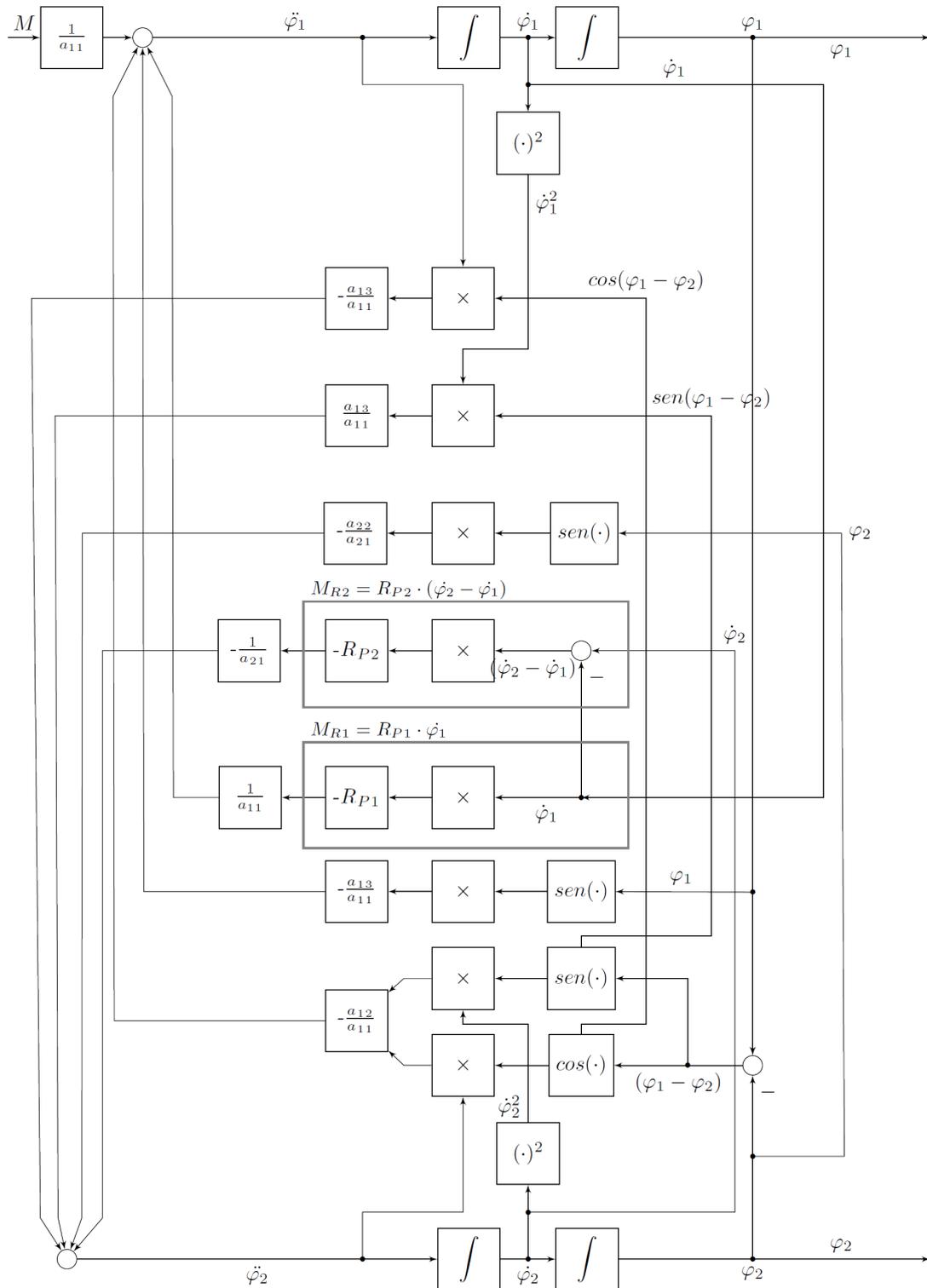


Figura 2.2: Diagrama de bloques del *Sistema de péndulo doble*.

Fuente : Elaboración propia

2.2.3 Resultados de simulación

La configuración de la simulación así como la implementación de los modelos en Matlab junto con los algoritmos se encuentran en el anexo de la presente tesis. A continuación se presentan los resultados:

- Resultados de la simulación sin excitación

En primera instancia, todos los modelos tendrán una condición inicial $x_0^T = \left[\frac{\pi}{1,01} 0 \frac{\pi}{1,01} 0 \right]$. Se define el vector de las variables $x = \left[\varphi_1 \dot{\varphi}_1 \varphi_2 \dot{\varphi}_2 \right]$ y se asigna una entrada sin excitación ($M = 0$) a fin ser simulado sin ninguna influencia y ver la respuesta de los ángulos φ_1 y φ_2 mostrados en la figura 2.3 para cada modelo. Para este caso se mostrará con una línea continua el resultado de Fcn-Block, una línea discontinua para el modelo con Matlab-Function y finalmente línea de puntos para el modelo con S-Function.

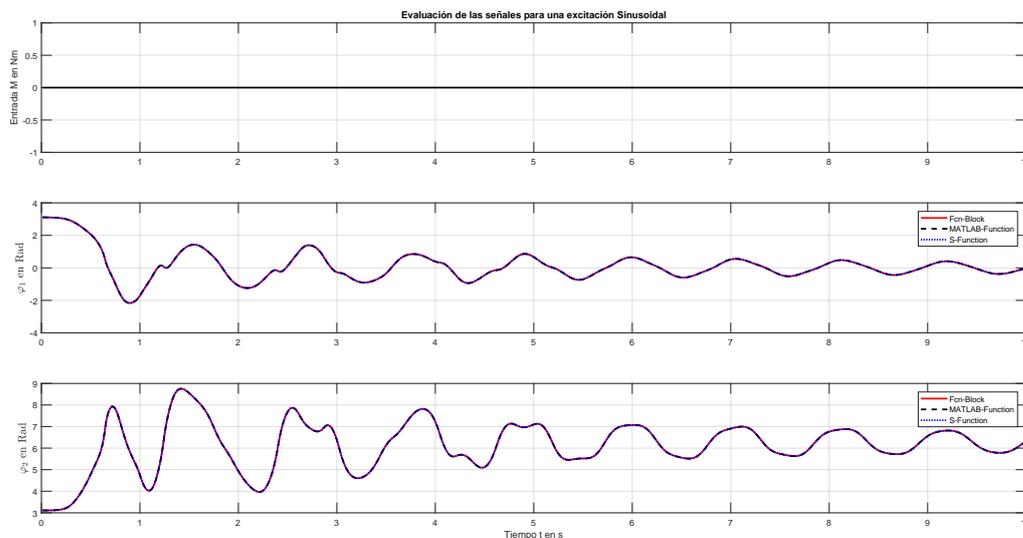


Figura 2.3: Resultado de la simulación para los ángulos φ_1 y φ_2 con C.I.

$$x_0^T = \left[\frac{\pi}{1,01} 0 \frac{\pi}{1,01} 0 \right] \text{ sin excitación a la entrada.}$$

Fuente : Elaboración propia

Como se puede apreciar en los resultados sin excitación, desde las posiciones iniciales de los péndulos se dejan caer ya que no existe ninguna fuerza motriz que haga una rotación de los péndulos; por lo que la gravedad juega un papel importante. Como se puede apreciar en la gráfica del ángulo φ_1 tiene un movimiento amortiguado casi regular en el que todos los modelos convergen

a su posición de equilibrio natural, es decir cuando el péndulo cuelga hacia abajo influenciado por la gravedad. Sobre el segundo péndulo, la gráfica del ángulo φ_2 tiene un movimiento mas caótico pero al igual que el primer ángulo llega a las misma posición de equilibrio debido a la fuerza de gravedad.

- Resultados de la simulación con excitación sinusoidal

Ahora los modelos serán excitados con una señal sinusoidal con $M(t) = 0.1 \cdot \sin(\omega \cdot t)$ y $\omega = 1 \frac{rad}{s}$ para el espacio de tiempo $t = 0$ hasta $t = 50s$

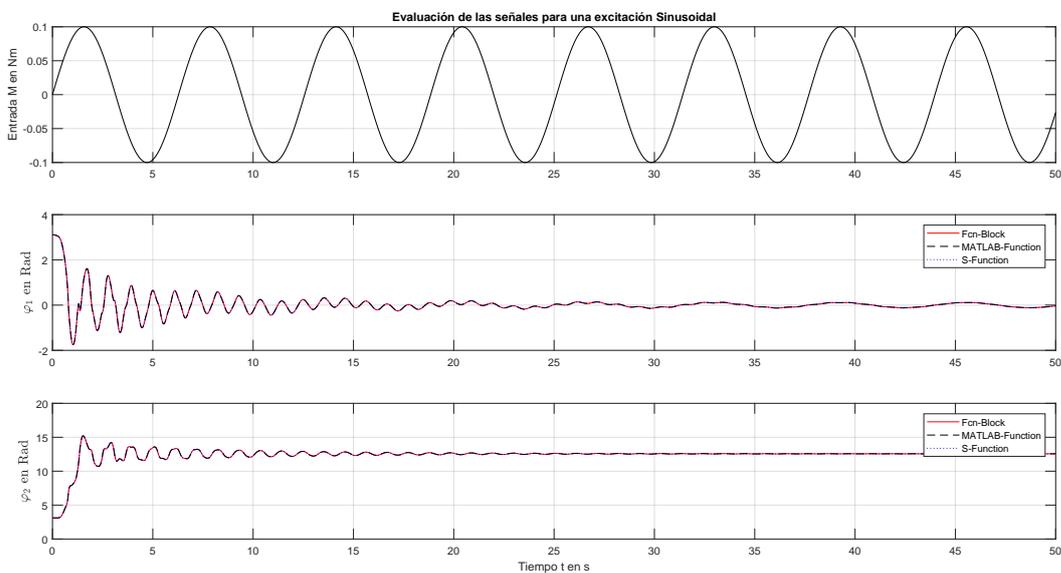


Figura 2.4: Resultado de la simulación de una excitación sinusoidal durante 50 segundos.

Fuente : Elaboración propia

Como se puede apreciar, la figura 2.4 los tres resultados de la simulación son representados. En el primero se muestra señal de entrada sinusoidal de amplitud 0.1 y frecuencia $1 \frac{rad}{s}$. Después , la siguiente gráfica muestra la posición del ángulo φ_1 y la ultima gráfica, muestra la segunda posición del ángulo φ_2 .

Como se puede apreciar, la primera posición de ángulo tiene casi el mismo recorrido que la señal sinusoidal en la entrada. Es muy cierto decir entonces que el momento en la entrada tiene una influencia directa en la primera posición de ángulo. La segunda posición de ángulo se mueve a una mayor frecuencia (cerca de 6 veces mas rápido) y pequeña amplitud (cerca de 10 veces pequeña). En otras palabras, la segunda posición tiene un valor de 4π

para el segundo ángulo casi 0 radianes; por lo que podemos decir, que los péndulos hacen una forma o especie de línea recta pero si nota más de cerca este resultado, el ángulo φ_1 oscila desde cerca a los 25 segundos en sincronía con la entrada sinusoidal con amplitud parecida pero de forma quasi periódica. En cuanto al segundo ángulo llega a su estado natural o de equilibrio oscilando en forma quasi simultanea al primer pendulo. Algo que se debe tambien añadir es que los primeros segundos de la simulación dichos ángulos empiezan de una posición inicial cercana a los 90 grados $x_0^T = [\frac{\pi}{1,01} \ 0 \ \frac{\pi}{1,01} \ 0]$ por lo que existe ya de por sí un momento que los péndulos tienen en un principio.

Para comprender mejor este comportamiento de los dos péndulos, vamos a simular los modelos por un intervalo de 2 segundos, esto se puede apreciar en la figura 3.5. En caso que el segundo péndulo no sea estimulado, seria el resultado de la simulación de la primera posición de ángulo también sinusoidal. Ahora, si agregamos el segundo péndulo, este tendría una influencia directa en la posición del primero ángulo, casi intuitivamente.

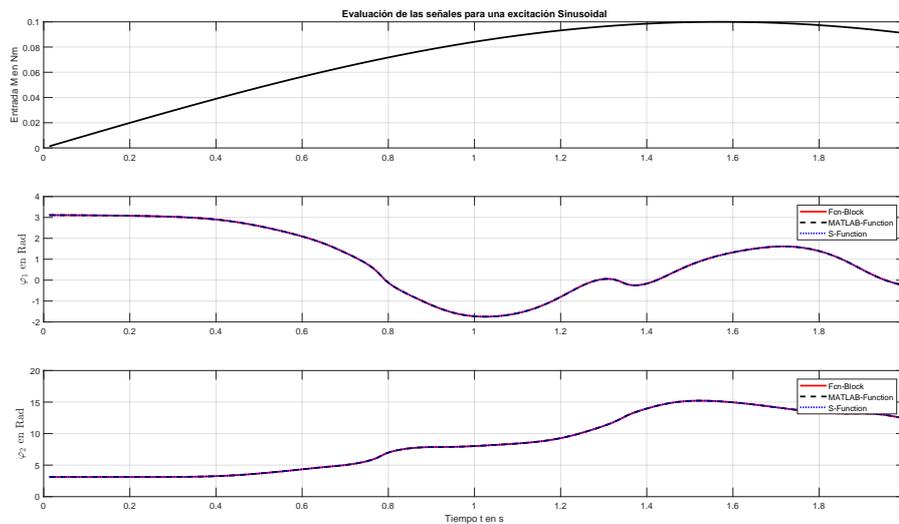


Figura 2.5: Resultado de la simulación de una excitación sinusoidal durante dos segundos.

Fuente : Elaboración propia

2.2.4 Linealización del Sistema de péndulo doble

Como se puede apreciar las ecuaciones 2.31 y 2.32 existen componentes trigonométricas (senos y cosenos) y así como términos cuadráticos. Por lo tanto,

todos estos componentes hacen que las ecuaciones y el comportamiento del *Sistema de péndulo doble* no sea lineal.

En efecto, sistemas como el del péndulo doble así como muchos otros son procesos no lineales cuyo comportamiento se puede describir mediante modelos lineales con un rango limitado y específico. En algunos casos una aproximación puede ser suficiente para prescindir de la no linealidad. Para eso se considera una aproximación que transforma la no linealidad a una estructura matemática más sencilla [27]. En nuestro caso se podría considerar los ángulos φ_1 y φ_2 como muy pequeños tal como se puede apreciar en la siguiente tabla:

Tabla 2.2: Aproximaciones

Ángulo	Aproximación	
$\varphi_1 \ll \frac{\pi}{2}$	$\sin \varphi_1 \approx \varphi_1$	$\cos \varphi_1 \approx 1$
$\varphi_2 \ll \frac{\pi}{2}$	$\sin \varphi_2 \approx \varphi_2$	$\cos \varphi_2 \approx 1$

Sin embargo, en nuestro caso, para realizar una buena modelización y evitar imprecisiones en el modelo provenientes de una aproximación, el sistema de doble péndulo deberá trabajar a diferentes puntos de operación, considerando ángulos de diferentes valores según la posición de los péndulos; por lo tanto, se necesitará linearizar el sistema. Los puntos de operación útiles son las dos posiciones en reposo, donde el doble péndulo cuelga hacia abajo y la otra en donde se mantiene en posición vertical. Otro punto de operación es cuando el péndulo 2 forma una perpendicular al péndulo 1; claro esta notar, que el péndulo 1 está situado en el eje x. Y posteriormente se consideraran otros puntos de trabajo que sean convenientes para la correcta simulación del modelo linealizado. En el siguiente apartado se mostrará una representación matemática la cual puede ser útil para aplicar estrategias de control pero con una condición de linealidad del sistema para este fin.

2.2.5 Sistema de péndulo doble en espacio de estado

El sistema puede describirse a continuación mediante ecuaciones diferenciales lineales de primer orden y, en consecuencia, puede describirse en la representación del espacio de estado de la siguiente forma:

$$\dot{x} = Ax + Bu \quad (2.37)$$

$$y = Cx + Du \quad (2.38)$$

El sistema dinámico no lineal será descrito como lineal en un espacio de estado y mediante un sistema de ecuaciones diferenciales:

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}) \quad (2.39)$$

$$\underline{y} = \underline{h}(\underline{x}, \underline{u}) \quad (2.40)$$

Los puntos de operación (PO) en la mayoría se encuentran, por ejemplo en el punto de reposo o equilibrio del sistema donde no existe ninguna dinámica. En efecto, el controlador tratará siempre de llevar la dinámica del sistema en equilibrio y mantenerlo allí. Para eso hacemos que nuestro sistema dinámico no lineal sea igualado a 0 a fin de obtener vector de salida en función del vector de entrada [24]. Y se representa de la siguiente forma:

$$\underline{0} = \dot{\underline{x}}_{PO} = \underline{f}(\underline{x}_{PO}, \underline{u}_{PO}) \quad (2.41)$$

Otro punto a considerar cuando el sistema este en punto de equilibrio o sin dinámica a fin de hallar los puntos de operación. Entonces se considera la velocidad y aceleración angular igual a 0. Entonces se tiene: $(\dot{\varphi}_1, \dot{\varphi}_2 = 0)$ y $(\ddot{\varphi}_1, \ddot{\varphi}_2 = 0)$ respectivamente. Además se considera $M = 0$. De las ecuaciones 2.35 y 2.36 se obtiene lo siguiente:

Por lo tanto de la primera ecuación quedaría de la siguiente manera:

$$0 = \frac{a_{12}}{a_{11} \cdot a_{21}} \cdot \cos(\varphi_1 - \varphi_2) \cdot [a_{22} \cdot \sin(\varphi_2)] - \frac{a_{13}}{a_{11}} \cdot \sin(\varphi_1) \quad (2.42)$$

Y la segunda ecuación quedaría de esta manera:

$$0 = \frac{a_{12}}{a_{11} \cdot a_{21}} \cdot \cos(\varphi_1 - \varphi_2) \cdot [a_{13} \cdot \sin(\varphi_1)] - \frac{a_{22}}{a_{21}} \cdot \sin(\varphi_2) \quad (2.43)$$

Como se puede apreciar, existen 2 ecuaciones 2.42 y 2.43 y dos incógnitas (φ_1 y φ_2). Hallamos entonces los puntos de operación. Como se explicó anteriormente, se lleva al sistema en el punto de equilibrio para esto podemos tener

2 puntos, considerando que el inicio del movimiento del péndulo es cuando los ángulos (φ_1 y φ_2) son iguales 0. Se calculan los ángulos por medio del comando SOLVE en MATLAB®/ Simulink. Se obtiene por lo tanto los ángulos que hacen que el *Sistema de péndulo doble* se encuentre en equilibrio. Los resultados se muestran en la tabla 2.4:

Tabla 2.3: Puntos de equilibrio de los Péndulos.

Variable	Designación	Descripción
φ_1	0	Péndulo 1 posicionado hacia abajo
φ_2	0	Péndulo 2 posicionado hacia abajo
φ_1	π	Péndulo 1 posicionado hacia arriba
φ_2	π	Péndulo 2 posicionado hacia arriba

Como se puede notar cuando los ángulos son $\varphi_1 = \varphi_2 = 0^\circ$, los péndulos se encuentran en su estado natural, es decir en reposo sin movimiento y se le considera "*Estable*". Por otro lado cuando los ángulos se encuentran a 180° (π), los péndulos se encuentran en una posición "*Inestable*" y es exactamente donde trataremos de controlar nuestro sistema.

Ahora bien es importante considerar la diferencia entre el vector de variable de estado y el vector de la variable de estado en el punto de operación, así también como en los vectores de entrada y salida para controlar las desviaciones de cada valor real contra el valor alrededor del punto de operación. Esta diferencia se expresa media un delta y se representa de la siguiente forma:

$$\begin{aligned}\Delta \underline{x} &= \underline{x} - \underline{x}_{PO} \\ \Delta \underline{u} &= \underline{u} - \underline{u}_{PO} \\ \Delta \underline{y} &= \underline{y} - \underline{y}_{PO}\end{aligned}\tag{2.44}$$

Así también como el vector de salida en el punto de operación:

$$\underline{y}_{PO} = \underline{h}(\underline{x}_{PO}, \underline{u}_{PO})\tag{2.45}$$

La forma de representación del sistema, es decir en espacio de estado y tener la forma de la ecuaciones 2.37 y 2.38, se deberá obtener la linealización de forma matemática por medio del Jacobiano que trata sobre las diferencias parciales de las ecuaciones que corresponden a la función del sistema $\underline{f}(\underline{x}, \underline{u})$

y la función de salida $\underline{h}(\underline{x}, \underline{u})$ entre la variable de estado y la variable de entrada [1]. Esta operación se muestra de la siguiente forma :

$$\begin{aligned} \underline{A} &= \left. \frac{\partial \underline{f}(\underline{x}, \underline{u})}{\partial \underline{x}} \right|_{\text{Punto de operación}} \\ \underline{B} &= \left. \frac{\partial \underline{f}(\underline{x}, \underline{u})}{\partial \underline{u}} \right|_{\text{Punto de operación}} \\ \underline{C} &= \left. \frac{\partial \underline{h}(\underline{x}, \underline{u})}{\partial \underline{x}} \right|_{\text{Punto de operación}} \\ \underline{D} &= \left. \frac{\partial \underline{h}(\underline{x}, \underline{u})}{\partial \underline{u}} \right|_{\text{Punto de operación}} \end{aligned} \quad (2.46)$$

Posteriormente, debemos tener en cuenta que tenemos un sistema multivariable por lo que se desarrolla la función vectorial \underline{f} en la serie de Taylor en los puntos de operación $\underline{x}_{PO}, \underline{u}_{PO}$ [1]. La matriz resultante se denomina matriz jacobiana y se ve de la siguiente manera.

En donde :

$$\left(\frac{\partial \underline{f}}{\partial \underline{x}} \right)_{\substack{\underline{x} = \underline{x}_{PO} \\ \underline{u} = \underline{u}_{PO}}} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}_{\substack{\underline{x} = \underline{x}_{PO} \\ \underline{u} = \underline{u}_{PO}}} = \underline{A} \quad (2.47)$$

y

$$\left(\frac{\partial \underline{f}}{\partial \underline{u}} \right)_{\substack{\underline{x} = \underline{x}_{PO} \\ \underline{u} = \underline{u}_{PO}}} = \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \dots & \frac{\partial f_1}{\partial u_m} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial u_1} & \dots & \frac{\partial f_n}{\partial u_m} \end{pmatrix}_{\substack{\underline{x} = \underline{x}_{PO} \\ \underline{u} = \underline{u}_{PO}}} = \underline{B} \quad (2.48)$$

Entonces se obtiene la ecuación del modelo lineal :

$$(\Delta \dot{\underline{x}}) = \underline{A}\Delta \underline{x} + \underline{B}\Delta \underline{u} \quad (2.49)$$

y para la ecuación de salida:

$$\Delta \underline{y} = \underline{C}\Delta \underline{x} + \underline{D}\Delta \underline{u} \quad (2.50)$$

con :

$$\left(\frac{\partial \underline{h}}{\partial \underline{x}} \right)_{\substack{\underline{x} = \underline{x}_{PO} \\ \underline{u} = \underline{u}_{PO}}} = \underline{C} \quad (2.51)$$

y

$$\left(\frac{\partial h}{\partial \underline{u}} \right)_{\substack{\underline{x} = \underline{x}_{PO} \\ \underline{u} = \underline{u}_{PO}}} = \underline{D} \quad (2.52)$$

2.2.6 Sistema lineal de doble péndulo

Para la linealización debemos desarrollar la serie de Taylor en el punto de operación. En nuestro caso, debemos notar el vector de estado con las variables de estado \underline{x} para el *Sistema de péndulo doble*, así como el vector de variable de entrada \underline{u} .

Para el vector \underline{x} de variables de estado:

$$\underline{x} = \begin{bmatrix} \varphi_1 \\ \dot{\varphi}_1 \\ \varphi_2 \\ \dot{\varphi}_2 \end{bmatrix} \xrightarrow{\text{derivada}} \dot{\underline{x}} = \begin{bmatrix} \dot{\varphi}_1 \\ \ddot{\varphi}_1 \\ \dot{\varphi}_2 \\ \ddot{\varphi}_2 \end{bmatrix} \quad (2.53)$$

y para el vector de entrada \underline{u} :

$$\underline{u} = \begin{bmatrix} M \end{bmatrix} \quad (2.54)$$

Entonces linealizamos las ecuaciones. Empezamos con la ecuación 2.31.

Se desarrolla la parte derecha de la ecuación y se iguala a 0. Con $M_{R1} = R_{P1} \cdot \dot{\varphi}_1$

$$\underbrace{a_{11} \cdot \ddot{\varphi}_1}_{f1} + \underbrace{a_{12} \cdot \cos(\varphi_1 - \varphi_2) \cdot \ddot{\varphi}_2}_{f2} + \underbrace{a_{12} \cdot \sin(\varphi_1 - \varphi_2) \cdot \dot{\varphi}_2^2}_{f3} + \underbrace{a_{13} \cdot \sin(\varphi_1)}_{f4} - \underbrace{M}_{f5} + \underbrace{M_{R1}}_{f6} = 0$$

$$\underbrace{f(M, \varphi_1, \dot{\varphi}_1, \ddot{\varphi}_1, \varphi_2, \dot{\varphi}_2, \ddot{\varphi}_2)}_{f(M, \varphi_1, \dot{\varphi}_1, \ddot{\varphi}_1, \varphi_2, \dot{\varphi}_2, \ddot{\varphi}_2)} \quad (2.55)$$

en una serie de Taylor en el punto de operación, se obtiene

$$0 = \left. \frac{\partial f(M, \varphi_1, \dot{\varphi}_1, \ddot{\varphi}_1, \varphi_2, \dot{\varphi}_2, \ddot{\varphi}_2)}{\partial \varphi_1} \right|_{PO} \cdot \Delta \varphi_1 + \left. \frac{\partial f(M, \varphi_1, \dot{\varphi}_1, \ddot{\varphi}_1, \varphi_2, \dot{\varphi}_2, \ddot{\varphi}_2)}{\partial \dot{\varphi}_1} \right|_{PO} \cdot \Delta \dot{\varphi}_1 + \left. \frac{\partial f(M, \varphi_1, \dot{\varphi}_1, \ddot{\varphi}_1, \varphi_2, \dot{\varphi}_2, \ddot{\varphi}_2)}{\partial \varphi_2} \right|_{PO} \cdot \Delta \varphi_2 + \left. \frac{\partial f(M, \varphi_1, \dot{\varphi}_1, \ddot{\varphi}_1, \varphi_2, \dot{\varphi}_2, \ddot{\varphi}_2)}{\partial \dot{\varphi}_2} \right|_{PO} \cdot \Delta \dot{\varphi}_2 + \left. \frac{\partial f(M, \varphi_1, \dot{\varphi}_1, \ddot{\varphi}_1, \varphi_2, \dot{\varphi}_2, \ddot{\varphi}_2)}{\partial M} \right|_{PO} \cdot \Delta M \quad (2.56)$$

Ahora se procede a linealizar las ecuación 2.55 por partes:

$$\begin{aligned}
 & 1) \overbrace{a_{11} \cdot \ddot{\varphi}_1}^{f1} \xrightarrow{lin.} a_{11} \cdot \left[\left. \frac{\partial f1}{\partial \ddot{\varphi}_1} \right|_{PO} \cdot \Delta \ddot{\varphi}_1 \right] \\
 & \hspace{15em} = a_{11} \cdot \Delta \ddot{\varphi}_1 \\
 & 2) \overbrace{a_{12} \cdot \cos(\varphi_1 - \varphi_2) \cdot \ddot{\varphi}_2}^{f2} \xrightarrow{lin.} a_{12} \cdot \left[\left. \frac{\partial f2}{\partial \varphi_1} \right|_{PO} \cdot \Delta \varphi_1 + \left. \frac{\partial f2}{\partial \varphi_2} \right|_{PO} \cdot \Delta \varphi_2 + \left. \frac{\partial f2}{\partial \ddot{\varphi}_2} \right|_{PO} \cdot \Delta \ddot{\varphi}_2 \right] \\
 & \hspace{15em} = a_{12} \cdot [-\sin(\varphi_{1PO} - \varphi_{2PO}) \cdot \ddot{\varphi}_{2PO} \cdot \Delta \varphi_1 + \sin(\varphi_{1PO} - \varphi_{2PO}) \cdot \ddot{\varphi}_{2PO} \cdot \Delta \varphi_2 \\
 & \hspace{18em} + \cos(\varphi_{1PO} - \varphi_{2PO}) \cdot \Delta \ddot{\varphi}_2] \\
 & 3) \overbrace{a_{12} \cdot \sin(\varphi_1 - \varphi_2) \cdot \dot{\varphi}_2^2}^{f3} \xrightarrow{lin.} a_{12} \cdot \left[\left. \frac{\partial f3}{\partial \varphi_1} \right|_{PO} \cdot \Delta \varphi_1 + \left. \frac{\partial f3}{\partial \varphi_2} \right|_{PO} \cdot \Delta \varphi_2 + \left. \frac{\partial f3}{\partial \dot{\varphi}_2} \right|_{PO} \cdot \Delta \dot{\varphi}_2 \right] \\
 & \hspace{15em} = a_{12} \cdot [\cos(\varphi_{1PO} - \varphi_{2PO}) \cdot \dot{\varphi}_{2PO}^2 \cdot \Delta \varphi_1 - \cos(\varphi_{1PO} - \varphi_{2PO}) \cdot \dot{\varphi}_{2PO}^2 \cdot \Delta \varphi_2 \\
 & \hspace{18em} + \sin(\varphi_{1PO} - \varphi_{2PO}) \cdot 2 \cdot \dot{\varphi}_{2PO} \cdot \Delta \dot{\varphi}_2] \\
 & 4) \overbrace{a_{13} \cdot \sin(\varphi_1)}^{f4} \xrightarrow{lin.} a_{13} \cdot \left[\left. \frac{\partial f4}{\partial \varphi_1} \right|_{PO} \cdot \Delta \varphi_1 \right] \\
 & \hspace{15em} = a_{13} \cdot \cos(\varphi_{1PO}) \cdot \Delta \varphi_1 \\
 & 5) \overbrace{-M}^{f5} \xrightarrow{lin.} \left[\left. \frac{\partial f5}{\partial M} \right|_{PO} \cdot \Delta M \right] \\
 & \hspace{15em} = -\Delta M \\
 & 6) \overbrace{M_{R1} = R_{P1} \cdot \dot{\varphi}_1}^{f6} \xrightarrow{lin.} \left[\left. \frac{\partial f6}{\partial \dot{\varphi}_1} \right|_{PO} \cdot \Delta \dot{\varphi}_1 \right] \\
 & \hspace{15em} = R_{P1} \cdot \Delta \dot{\varphi}_1
 \end{aligned} \tag{2.57}$$

Entonces ordenamos la primera ecuación de movimiento linealizada.

$$\begin{aligned}
 \Delta \ddot{\varphi}_1 = & \Delta \varphi_1 \cdot \left[\frac{a_{12}}{a_{11}} \cdot \sin(\varphi_{1PO} - \varphi_{2PO}) \cdot \ddot{\varphi}_{2PO} - \frac{a_{12}}{a_{11}} \cdot \cos(\varphi_{1PO} - \varphi_{2PO}) \cdot \dot{\varphi}_{2PO}^2 - \frac{a_{13}}{a_{11}} \cdot \cos(\varphi_{1PO}) \right] \\
 & + \Delta \varphi_2 \cdot \left[-\frac{a_{12}}{a_{11}} \cdot \sin(\varphi_{1PO} - \varphi_{2PO}) \cdot \ddot{\varphi}_{2PO} + \frac{a_{12}}{a_{11}} \cdot \cos(\varphi_{1PO} - \varphi_{2PO}) \cdot \dot{\varphi}_{2PO}^2 \right] \\
 & + \Delta \dot{\varphi}_1 \cdot \left[-\frac{1}{a_{11}} \cdot R_{P1} \right] \\
 & + \Delta \ddot{\varphi}_2 \cdot \left[-\frac{a_{12}}{a_{11}} \cdot \cos(\varphi_{1PO} - \varphi_{2PO}) \right] \\
 & + \Delta \dot{\varphi}_2 \cdot \left[-\frac{a_{12}}{a_{11}} \cdot \sin(\varphi_{1PO} - \varphi_{2PO}) \cdot 2 \cdot \dot{\varphi}_{2PO} \right] \\
 & + \Delta M \cdot \left[\frac{1}{a_{11}} \right]
 \end{aligned} \tag{2.58}$$

y de forma análoga se linealiza la ecuación 2.32. Con $M_{R2} = R_{P2} \cdot (\varphi_2 - \varphi_1)$

$$\underbrace{a_{12} \cdot \cos(\varphi_1 - \varphi_2) \cdot \ddot{\varphi}_1}_{g^1} + \underbrace{a_{21} \cdot \ddot{\varphi}_2}_{g^2} - \underbrace{a_{12} \cdot \sin(\varphi_1 - \varphi_2) \cdot \dot{\varphi}_1^2}_{g^3} + \underbrace{a_{22} \cdot \sin(\varphi_2)}_{g^4} + \underbrace{M_{R2}}_{g^5} = 0$$

$$g(M, \varphi_1, \dot{\varphi}_1, \ddot{\varphi}_1, \varphi_2, \dot{\varphi}_2, \ddot{\varphi}_2)$$

$$(2.59)$$

en una serie de Taylor para la segunda ecuación de movimiento en el punto de operación, se obtiene

$$0 = \left. \frac{\partial g(M, \varphi_1, \dot{\varphi}_1, \ddot{\varphi}_1, \varphi_2, \dot{\varphi}_2, \ddot{\varphi}_2)}{\partial \varphi_1} \right|_{PO} \cdot \Delta \varphi_1 + \left. \frac{\partial g(M, \varphi_1, \dot{\varphi}_1, \ddot{\varphi}_1, \varphi_2, \dot{\varphi}_2, \ddot{\varphi}_2)}{\partial \dot{\varphi}_1} \right|_{PO} \cdot \Delta \dot{\varphi}_1 + \left. \frac{\partial g(M, \varphi_1, \dot{\varphi}_1, \ddot{\varphi}_1, \varphi_2, \dot{\varphi}_2, \ddot{\varphi}_2)}{\partial \varphi_2} \right|_{PO} \cdot \Delta \varphi_2 + \left. \frac{\partial g(M, \varphi_1, \dot{\varphi}_1, \ddot{\varphi}_1, \varphi_2, \dot{\varphi}_2, \ddot{\varphi}_2)}{\partial \dot{\varphi}_2} \right|_{PO} \cdot \Delta \dot{\varphi}_2 + \left. \frac{\partial g(M, \varphi_1, \dot{\varphi}_1, \ddot{\varphi}_1, \varphi_2, \dot{\varphi}_2, \ddot{\varphi}_2)}{\partial M} \right|_{PO} \cdot \Delta M$$

$$(2.60)$$

Ahora se procede a linealizar las ecuación 2.59 por partes:

$$1) \underbrace{a_{12} \cdot \cos(\varphi_1 - \varphi_2) \cdot \ddot{\varphi}_1}_{g^1} \xrightarrow{lin.} a_{12} \cdot \left[\left. \frac{\partial g^1}{\partial \varphi_1} \right|_{PO} \cdot \Delta \varphi_1 + \left. \frac{\partial g^1}{\partial \varphi_2} \right|_{PO} \cdot \Delta \varphi_2 + \left. \frac{\partial g^1}{\partial \ddot{\varphi}_1} \right|_{PO} \cdot \Delta \ddot{\varphi}_1 \right]$$

$$= a_{12} \cdot [-\sin(\varphi_{1PO} - \varphi_{2PO}) \cdot \ddot{\varphi}_{1PO} \cdot \Delta \varphi_1 + \sin(\varphi_{1PO} - \varphi_{2PO}) \cdot \ddot{\varphi}_{1PO} \cdot \Delta \varphi_2 + \cos(\varphi_{1PO} - \varphi_{2PO}) \cdot \Delta \ddot{\varphi}_1]$$

$$2) \underbrace{a_{21} \cdot \ddot{\varphi}_2}_{g^2} \xrightarrow{lin.} a_{21} \cdot \left[\left. \frac{\partial g^2}{\partial \ddot{\varphi}_2} \right|_{PO} \cdot \Delta \ddot{\varphi}_2 \right]$$

$$= a_{21} \cdot \Delta \ddot{\varphi}_2$$

$$3) \underbrace{a_{12} \cdot \sin(\varphi_1 - \varphi_2) \cdot \dot{\varphi}_1^2}_{g^3} \xrightarrow{lin.} a_{12} \cdot \left[\left. \frac{\partial g^3}{\partial \varphi_1} \right|_{PO} \cdot \Delta \varphi_1 + \left. \frac{\partial g^3}{\partial \varphi_2} \right|_{PO} \cdot \Delta \varphi_2 + \left. \frac{\partial g^3}{\partial \dot{\varphi}_1} \right|_{PO} \cdot \Delta \dot{\varphi}_1 \right]$$

$$= a_{12} \cdot [\cos(\varphi_{1PO} - \varphi_{2PO}) \cdot \dot{\varphi}_{1PO}^2 \cdot \Delta \varphi_1 - \cos(\varphi_{1PO} - \varphi_{2PO}) \cdot \dot{\varphi}_{1PO}^2 \cdot \Delta \varphi_2 + \sin(\varphi_{1PO} - \varphi_{2PO}) \cdot 2 \cdot \dot{\varphi}_{1PO} \cdot \Delta \dot{\varphi}_1]$$

$$4) \underbrace{a_{22} \cdot \sin(\varphi_2)}_{g^4} \xrightarrow{lin.} a_{22} \cdot \left[\left. \frac{\partial g^4}{\partial \varphi_2} \right|_{PO} \cdot \Delta \varphi_2 \right]$$

$$= a_{22} \cdot \cos(\varphi_{2PO}) \cdot \Delta \varphi_2$$

$$5) \underbrace{M_{R2} = R_{P2} \cdot (\varphi_2 - \varphi_1)}_{g^5} \xrightarrow{lin.} R_{P2} \cdot \left[\left. \frac{\partial g^5}{\partial \varphi_2} \right|_{PO} \cdot \Delta \varphi_2 - \left. \frac{\partial g^5}{\partial \varphi_1} \right|_{PO} \cdot \Delta \varphi_1 \right]$$

$$= R_{P2} \cdot (\Delta \varphi_2 - \Delta \varphi_1)$$

$$(2.61)$$

Entonces ordenamos la segunda ecuación de movimiento linealizada.

$$\begin{aligned}
\Delta\ddot{\varphi}_2 = & \Delta\varphi_1 \cdot \left[\frac{a_{12}}{a_{21}} \cdot \sin(\varphi_{1PO} - \varphi_{2PO}) \cdot \ddot{\varphi}_{1PO} - \frac{a_{12}}{a_{21}} \cdot \cos(\varphi_{1PO} - \varphi_{2PO}) \cdot \dot{\varphi}_{1PO}^2 \right] \\
& + \Delta\varphi_2 \cdot \left[-\frac{a_{12}}{a_{21}} \cdot \sin(\varphi_{1PO} - \varphi_{2PO}) \cdot \ddot{\varphi}_{1PO} + \frac{a_{12}}{a_{21}} \cdot \cos(\varphi_{1PO} - \varphi_{2PO}) \cdot \dot{\varphi}_{1PO}^2 - \frac{a_{22}}{a_{21}} \cdot \cos(\varphi_{2PO}) \right] \\
& + \Delta\dot{\varphi}_1 \cdot \left[-\frac{a_{12}}{a_{21}} \cdot \sin(\varphi_{1PO} - \varphi_{2PO}) \cdot 2 \cdot \dot{\varphi}_{1PO} + \frac{1}{a_{21}} \cdot R_{P2} \right] \\
& + \Delta\ddot{\varphi}_1 \cdot \left[-\frac{a_{12}}{a_{21}} \cdot \cos(\varphi_{1PO} - \varphi_{2PO}) \right] \\
& + \Delta\dot{\varphi}_2 \cdot \left[-\frac{1}{a_{21}} \cdot R_{P2} \right]
\end{aligned} \tag{2.62}$$

Ahora bien, formaremos el modelo lineal a partir de las ecuaciones antes halladas asignando constantes. Comenzamos con la ecuación 2.58:

$$\begin{aligned}
X_1 = & \left[\frac{a_{12}}{a_{11}} \cdot \sin(\varphi_{1PO} - \varphi_{2PO}) \cdot \ddot{\varphi}_{2PO} - \frac{a_{12}}{a_{11}} \cdot \cos(\varphi_{1PO} - \varphi_{2PO}) \cdot \dot{\varphi}_{2PO}^2 - \frac{a_{13}}{a_{11}} \cdot \cos(\varphi_{1PO}) \right] \\
X_2 = & \left[-\frac{a_{12}}{a_{11}} \cdot \sin(\varphi_{1PO} - \varphi_{2PO}) \cdot \ddot{\varphi}_{2PO} + \frac{a_{12}}{a_{11}} \cdot \cos(\varphi_{1PO} - \varphi_{2PO}) \cdot \dot{\varphi}_{2PO}^2 \right] \\
X_3 = & \left[-\frac{1}{a_{11}} \cdot R_{P1} \right] \\
X_4 = & \left[-\frac{a_{12}}{a_{11}} \cdot \cos(\varphi_{1PO} - \varphi_{2PO}) \right] \\
X_5 = & \left[-\frac{a_{12}}{a_{11}} \cdot \sin(\varphi_{1PO} - \varphi_{2PO}) \cdot 2 \cdot \dot{\varphi}_{2PO} \right] \\
X_6 = & \left[\frac{1}{a_{11}} \right]
\end{aligned} \tag{2.63}$$

Entonces tenemos la primera ecuación de movimiento simplificada:

$$\Delta\ddot{\varphi}_1 = \Delta\varphi_1 \cdot X_1 + \Delta\varphi_2 \cdot X_2 + \Delta\dot{\varphi}_1 \cdot X_3 + \Delta\ddot{\varphi}_2 \cdot X_4 + \Delta\dot{\varphi}_2 \cdot X_5 + \Delta M \cdot X_6 \tag{2.64}$$

De la misma forma, formaremos el modelo lineal a partir de la segunda ecuación de movimiento. También asignando constantes a fin de simplificar el análisis 2.62:

$$\begin{aligned}
Y_1 = & \left[\frac{a_{12}}{a_{21}} \cdot \sin(\varphi_{1PO} - \varphi_{2PO}) \cdot \ddot{\varphi}_{1PO} - \frac{a_{12}}{a_{21}} \cdot \cos(\varphi_{1PO} - \varphi_{2PO}) \cdot \dot{\varphi}_{1PO}^2 \right] \\
Y_2 = & \left[-\frac{a_{12}}{a_{21}} \cdot \sin(\varphi_{1PO} - \varphi_{2PO}) \cdot \ddot{\varphi}_{1PO} + \frac{a_{12}}{a_{21}} \cdot \cos(\varphi_{1PO} - \varphi_{2PO}) \cdot \dot{\varphi}_{1PO}^2 - \frac{a_{22}}{a_{21}} \cdot \cos(\varphi_{2PO}) \right] \\
Y_3 = & \left[-\frac{a_{12}}{a_{21}} \cdot \sin(\varphi_{1PO} - \varphi_{2PO}) \cdot 2 \cdot \dot{\varphi}_{1PO} + \frac{1}{a_{21}} \cdot R_{P2} \right] \\
Y_4 = & \left[-\frac{a_{12}}{a_{21}} \cdot \cos(\varphi_{1PO} - \varphi_{2PO}) \right] \\
Y_5 = & \left[-\frac{1}{a_{21}} \cdot R_{P2} \right]
\end{aligned} \tag{2.65}$$

Entonces tenemos la segunda ecuación de movimiento simplificada:

$$\Delta\ddot{\varphi}_2 = \Delta\varphi_1 \cdot Y_1 + \Delta\varphi_2 \cdot Y_2 + \Delta\dot{\varphi}_1 \cdot Y_3 + \Delta\dot{\varphi}_1 \cdot Y_4 + \Delta\dot{\varphi}_2 \cdot Y_5 \quad (2.66)$$

Como se puede apreciar, en las 2 ecuaciones de movimiento simplificadas 2.64 y 2.66 las derivadas de mayor orden depende una de otra. Por lo tanto, finalmente reemplazamos una ecuación en la otra y obtenemos las siguientes ecuaciones de movimientos. La primera ecuación de movimiento linealizada es la siguiente:

$$\begin{aligned} \Delta\ddot{\varphi}_1 = & \Delta\varphi_1 \cdot \left[\frac{X_1 + Y_1 \cdot X_4}{1 - Y_4 \cdot X_4} \right] + \Delta\varphi_2 \cdot \left[\frac{X_2 + Y_2 \cdot X_4}{1 - Y_4 \cdot X_4} \right] + \Delta\dot{\varphi}_1 \cdot \left[\frac{X_3 + Y_3 \cdot X_4}{1 - Y_4 \cdot X_4} \right] \\ & + \Delta\dot{\varphi}_2 \cdot \left[\frac{X_5 + Y_5 \cdot X_4}{1 - Y_4 \cdot X_4} \right] + \Delta M \cdot \left[\frac{X_6}{1 - Y_4 \cdot X_4} \right] \end{aligned} \quad (2.67)$$

y la segunda ecuación de movimiento linealizada es la siguiente:

$$\begin{aligned} \Delta\ddot{\varphi}_2 = & \Delta\varphi_1 \cdot \left[\frac{Y_1 + X_1 \cdot Y_4}{1 - Y_4 \cdot X_4} \right] + \Delta\varphi_2 \cdot \left[\frac{Y_2 + X_2 \cdot Y_4}{1 - Y_4 \cdot X_4} \right] + \Delta\dot{\varphi}_1 \cdot \left[\frac{Y_3 + X_3 \cdot Y_4}{1 - Y_4 \cdot X_4} \right] \\ & + \Delta\dot{\varphi}_2 \cdot \left[\frac{Y_5 + X_5 \cdot Y_4}{1 - Y_4 \cdot X_4} \right] + \Delta M \cdot \left[\frac{X_6 \cdot Y_4}{1 - Y_4 \cdot X_4} \right] \end{aligned} \quad (2.68)$$

2.2.7 Espacio de Estado del Sistema linealizado de doble péndulo

Espacio de Estado para el modelo linealizado del Péndulo Doble

Para el vector $\Delta\underline{x}$ de variables de estado:

$$\Delta\underline{x} = \begin{bmatrix} \Delta\varphi_1 \\ \Delta\dot{\varphi}_1 \\ \Delta\varphi_2 \\ \Delta\dot{\varphi}_2 \end{bmatrix} \xrightarrow{\text{derivada}} \Delta\dot{\underline{x}} = \begin{bmatrix} \Delta\dot{\varphi}_1 \\ \Delta\ddot{\varphi}_1 \\ \Delta\dot{\varphi}_2 \\ \Delta\ddot{\varphi}_2 \end{bmatrix} \quad (2.69)$$

Entonces se obtiene la ecuación del modelo linealizado del sistema :

$$\begin{aligned} \Delta\dot{\underline{x}} &= A \cdot \Delta\underline{x} + B \cdot \Delta\underline{M} \\ \Delta\underline{y} &= C \cdot \Delta\underline{x} + D \cdot \Delta\underline{M} \end{aligned} \quad (2.70)$$

En donde las matrices A, B y C serían de la siguiente forma:

$$\begin{bmatrix} \Delta\dot{\varphi}_1 \\ \Delta\ddot{\varphi}_1 \\ \Delta\dot{\varphi}_2 \\ \Delta\ddot{\varphi}_2 \end{bmatrix} = \overbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ \left[\frac{X_1 + Y_1 \cdot X_4}{1 - Y_4 \cdot X_4} \right] & \left[\frac{X_3 + Y_3 \cdot X_4}{1 - Y_4 \cdot X_4} \right] & \left[\frac{X_2 + Y_2 \cdot X_4}{1 - Y_4 \cdot X_4} \right] & \left[\frac{X_5 + Y_5 \cdot X_4}{1 - Y_4 \cdot X_4} \right] \\ 0 & 0 & 0 & 1 \\ \left[\frac{Y_1 + X_1 \cdot Y_4}{1 - Y_4 \cdot X_4} \right] & \left[\frac{Y_3 + X_3 \cdot Y_4}{1 - Y_4 \cdot X_4} \right] & \left[\frac{Y_2 + X_2 \cdot Y_4}{1 - Y_4 \cdot X_4} \right] & \left[\frac{Y_5 + X_5 \cdot Y_4}{1 - Y_4 \cdot X_4} \right] \end{bmatrix}}^A \begin{bmatrix} \Delta\varphi_1 \\ \Delta\dot{\varphi}_1 \\ \Delta\varphi_2 \\ \Delta\dot{\varphi}_2 \end{bmatrix} + \overbrace{\begin{bmatrix} 0 \\ \left[\frac{X_6}{1 - Y_4 \cdot X_4} \right] \\ 0 \\ \left[\frac{X_6 \cdot Y_4}{1 - Y_4 \cdot X_4} \right] \end{bmatrix}}^B \left[\Delta M \right]$$

$$\Delta y = \overbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}^C \begin{bmatrix} \Delta\varphi_1 \\ \Delta\dot{\varphi}_1 \\ \Delta\varphi_2 \\ \Delta\dot{\varphi}_2 \end{bmatrix} + \overbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}^D \begin{bmatrix} \Delta M \end{bmatrix} \quad (2.71)$$

Las dimensiones serían de la siguiente manera:

Tabla 2.4: Dimensión de las matrices

Matriz	Descripción	Dimensión
<u>A</u>	Matriz del Sistema/Estado	(4,4) - Matriz
<u>B</u>	Matriz de Control/Entrada	(4,1) - Matriz
<u>C</u>	Matriz de Observación/Salida	(2,4) - Matriz
<u>D</u>	Matriz de Transmisión directa	(2,1) - Matriz

2.2.8 Linealización en software

En MATLAB se puede linealizar ecuaciones diferenciales no lineales a través de código en forma simbólica; de modo que las matrices A, B, C y D se puedan obtener. Tal como en la parte analítica 2.39 y 2.40 el sistema puede ser descrito mediante ecuaciones diferenciales de primero orden y son de la forma:

$$\begin{aligned} \underline{\dot{x}} &= \underline{f(x, u)} \\ \underline{y} &= \underline{h(x, u)} \end{aligned} \quad (2.72)$$

Cuando f y h así como los vectores son impresos en forma simbólica, se puede entonces obtener las matrices de espacio de estado con el comando `jacobian` y se muestra en el siguiente algoritmo 2.1

Algoritmo 2.1: Comando `jacobian`

```
1 >>A_ap=jacobian(f,x);
2 >>B_ap=jacobian(f,u);
3 >>C_ap=jacobian(h,x);
4 >>D_ap=jacobian(h,u);
```

Ahora bien, el sistema debe ser en 3 tres puntos de operación linealizado. Los puntos operación convenientes son las dos posiciones de equilibrio del sistema tal como se vio en la tabla 2.5, en donde el doble péndulo esta posicionado

o colgado hacia abajo (PO_1) y cuando el péndulo se mantiene en posición vertical o hacia arriba (PO_2). El tercer punto de operación (PO_3) hace que los péndulos hagan un ángulo perpendicular entre sí. Algo muy importante a notar es que el Péndulo 1 está situado en el eje X. Ahora bien, el tercer Punto de Operación se eligió a criterio propia ya que la idea es demostrar que el sistema en ese punto no será controlable.

A continuación se muestran las operaciones para el cálculo de Momento M_{OP} para cada uno de los puntos de operación en el algoritmo 2.2 en MATLAB:

Algoritmo 2.2: Puntos de Operación

```

1 %% Puntos de Operacion
2 syms phi1 phi2 dphi1 dphi2 M
3 % Punto de Operacion 1
4 x_po1 = [0 0 0 0];
5 [f, h] = nonlinear_model();
6 f_po1 = subs(f(2), {phi1, dphi1, phi2, dphi2}, {x_po1(1), x_po1(2)
7 , x_po1(3), x_po1(4)});
8 M_po1 = solve(f_po1==0, M);
9 % Punto de Operacion 2
10 x_po2 = [pi 0 pi 0];
11 [f, h] = nonlinear_model();
12 f_po2 = subs(f(2), {phi1, dphi1, phi2, dphi2}, {x_po2(1), x_po2(2)
13 , x_po2(3), x_po2(4)});
14 M_po2 = solve(f_po2==0, M);
15 % Punto de Operacion 3
16 x_po3 = [pi/2 0 pi 0];
17 [f, h] = nonlinear_model();
18 f_po3 = subs(f(2), {phi1, dphi1, phi2, dphi2}, {x_po3(1), x_po3(2)
19 , x_po3(3), x_po3(4)});
20 M_po3 = solve(f_po3==0, M);

```

El Momento M_{PO} para los dos primeros puntos de operación son cero, ya que son posiciones de equilibrio del sistema. Si el Momento M_{PO} no es cero, la desviación corresponde a todos los demás puntos que no representan posiciones de equilibrio. Este hecho se refleja en el resultado del punto de operación PO_3 donde está el Momento $M_{PO_3} = 0.8823 Nm \neq 0$.

Luego, el sistema se linealiza alrededor de los tres puntos de operación. Para eso se usa la función que se muestra en el siguiente algoritmo 2.3. Los valores de retorno son las matrices (A,B,C,D) en cada punto de operación respectivo.

Tabla 2.5: Puntos de operación y Momentos.

Puntos de Operación	Momento	Descripción
$PO_1 = [0;0;0;0]$	$M_{PO1} = 0Nm$	Péndulo 1 y 2 posicionados hacia abajo
$PO_2 = [\pi;0;\pi;0]$	$M_{PO2} = 0Nm$	Péndulo 1 y 2 posicionados hacia arriba
$PO_3 = [\frac{\pi}{2};0;\pi;0]$	$M_{PO3} = 0.8829Nm$	Péndulo 1 y 2 forman ángulo perpendicular

Algoritmo 2.3: Linealización en los Puntos de Operación

```

1 %% Linealización
2 PO1 = [x_po1 M_po1];
3 [ A1,B1,C1,D1 ] = linealizacion( f,h,PO1 );
4 PO2 = [x_po2 M_po2];
5 [ A2,B2,C2,D2 ] = linealizacion( f,h,PO2 );
6 PO3 = [x_po3 M_po3];
7 [ A3,B3,C3,D3 ] = linealizacion( f,h,PO3 );

```

Los resultados se muestran en las siguientes matrices:

$$\begin{aligned}
 A_{PO1} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -126 & 0 & 63 & 0 \\ 0 & 0 & 0 & 1 \\ 189 & 0 & -168 & 0 \end{bmatrix} & B_{PO1} &= \begin{bmatrix} 0 \\ 143 \\ 0 \\ -214 \end{bmatrix} & C_{PO1} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} & D_{PO1} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
 A_{PO2} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 126 & 0 & -63 & 0 \\ 0 & 0 & 0 & 1 \\ -189 & 0 & 168 & 0 \end{bmatrix} & B_{PO2} &= \begin{bmatrix} 0 \\ 143 \\ 0 \\ -214 \end{bmatrix} & C_{PO2} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} & D_{PO2} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
 A_{PO3} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 74 & 0 \end{bmatrix} & B_{PO3} &= \begin{bmatrix} 0 \\ 63 \\ 0 \\ 0 \end{bmatrix} & C_{PO3} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} & D_{PO3} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}
 \end{aligned}$$

2.2.9 Los valores propios de los modelos de espacio de estado

Una de las condiciones más importantes para hallar los valores/vectores propios de un sistema de ecuaciones lineales y no lineales es que sean cuadradas $n \times n$. Ahora bien en nuestro caso, lo que determina la estabilidad de un sistema son sus valores propios de un sistema linealizado en sus puntos de operación.

Por ejemplo, si son reales o imaginarios o si son menores y/o mayores a 0 [24]. Se hallan de la siguiente manera:

$$\lambda \underline{x} = \underline{A}x \quad (2.73)$$

Como se puede apreciar en la ecuación 2.73 el término \underline{x} que recibe la denominación vector propio de los valores propios λ . Ahora para calcular los n valores propios λ_i de una matriz \underline{A} con dimensión $n \times n$, se deberá utilizar la determinante igualada a cero y se representa así:

$$\det(\underline{A} - \lambda \underline{I}) = 0 \quad (2.74)$$

Luego del cálculo de la determinante de una matriz $n \times n$ se van a obtener los n diferentes valores propios λ_i ; de la misma forma existen también n vectores propios \underline{x}_i lineales e independientes donde se hallan a través de la ecuación lineal homogénea :

$$(\underline{A} - \lambda_i \underline{I}) \underline{x}_i = 0 \quad (2.75)$$

En MATLAB el algoritmo sería así:

Algoritmo 2.4: Cálculo de los valores propios

```
1 %% Valores propios
2 lambda1=double(eig(A1));
3 lambda2=double(eig(A2));
4 lambda3=double(eig(A3));
```

Y los resultados de los valores propios son :

$$\lambda_1 = \begin{bmatrix} 16.0744i \\ 5.9929i \\ -16.0744i \\ -5.9929i \end{bmatrix} \quad \lambda_2 = \begin{bmatrix} 5.9929 \\ 16.0744 \\ -5.9929 \\ -16.0744 \end{bmatrix} \quad \lambda_3 = \begin{bmatrix} 0 \\ 0 \\ -8.5776 \\ 8.5776 \end{bmatrix}$$

Los valores propios del sistema linealizado en el primer punto de operación son puramente imaginarios conjugados. Eso quiere decir que el comportamiento del sistema corresponde al de un oscilador armónico y por lo tanto es críticamente estable. Los valores propios del segundo punto operación tienen partes reales positivas. Esto significa que el sistema es inestable en este

punto para ambos péndulos. El tercer punto de trabajo es críticamente estable de una varilla del péndulo, pero inestable debido a la otra.

Si se tuviera en cuenta la fricción, se amortiguarían los movimientos del péndulo. En el primer punto de operación, la oscilación decaería. Los valores propios ya no serían puramente imaginarios. Cuanto mayor fuera la fricción, más se amortiguaría la oscilación y menor sería el ángulo entre la línea de sobregiro constante y el eje real. Así que la amortiguación sólo depende de la parte real.

2.2.10 Controlabilidad y Observabilidad del Sistema

Las características de un sistema puede acotarse en qué tanto controlable y observable es. Es decir en el caso que no sea controlable, no sería posible diseñar un sistema de control. Y en cuanto en el caso que no sea observable, no se podría saber lo que ocurre en un sistema durante el proceso de control; es decir, no se podría saber si un valor de una variable de estado sobrepasaría, por ejemplo, el límite permitido ocasionando pérdidas o falta de control del sistema en el peor de los casos.

Analizamos las notaciones para controlabilidad y observabilidad (citado de [24]): "Un sistema dinámico se denomina completamente *controlable* cuando desde cada estado inicial \underline{x}_0 por medio de una adecuada variable de entrada $\underline{u}(t)$, dentro de un intervalo de tiempo finito $[0, T_e]$, en cada variable deseada $\underline{x}_{deseada}$ puede ser transmitida. Y un sistema dinámico se denomina completamente *observable* cuando se puede inferir todas las variables de estado $\underline{x}(t_0)$ a partir de la marcha de las variables de entrada $\underline{u}(t)$ y salida $\underline{y}(t)$ dentro de un intervalo de tiempo finito $[t_0, t_e]$ ".

En la figura 2.6 se ve un diagrama de bloques en donde las 2 notaciones se cumple. Desde la entrada $U(s)$ se puede llegar a $X1(s)$ y $X2(s)$. De la misma forma desde cualquier de las salidas ($Y(s)$) se puede llegar también a $X1(s)$ y $X2(s)$.

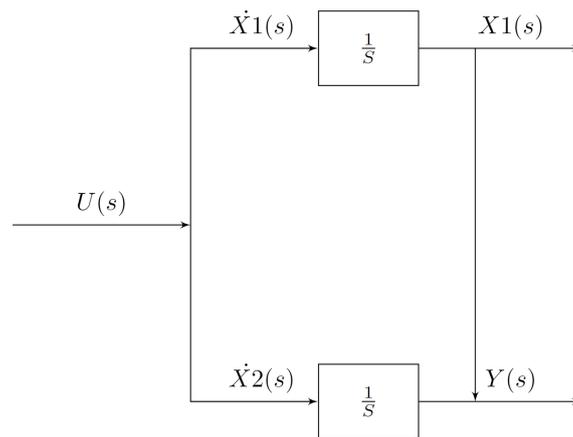


Figura 2.6: Sistema *Controlable y Observable*
Fuente : Elaboración propia

2.2.11 Criterios de Controlabilidad y Observabilidad

-Criterio de Kalman

En cuanto al criterio de Kalman, existen un criterio para controlabilidad y otro para observabilidad. En cuanto al criterio de estabilidad se tiene la siguiente matriz S_S de dimensión $(n \times n \cdot m)$ en donde:

$$S_S = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \quad (2.76)$$

Para considerar que el sistema (A, B) es completamente controlable, la matriz S_S tendrá que tener Rango n [28], para MATLAB se usa la función `ctrb`:

$$\text{Rang}(S_S) = n$$

En cuanto al criterio de observabilidad se tiene la siguiente matriz S_B de dimensión $(r \cdot n \times n)$ en donde:

$$S_B = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (2.77)$$

Para considerar que el sistema (A, C) es completamente observable, la matriz S_B tendrá que tener Rango n [28], para MATLAB se usa la funcion `obsv`:

$$\text{Rang}(S_B) = n$$

-Criterio de Gilbert

Antes de poder entrar a fondo sobre el criterio de Gilbert, se debe entender la *Transformación de un sistema en formas Canónicas*. Para eso, debemos entender la diagonalización en forma canónica; en donde la matriz A será diagonalizada en primera instancia por medio de la *Transformación de Semejanza*, en donde se encuentran todos los n -valores propios $(\lambda_1, \lambda_2, \dots, \lambda_n)$ del sistema en una diagonal principal de la matriz resultante \tilde{A} [24]. Donde se cumple:

$$\tilde{A} = \text{diag} \lambda_i = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \lambda_n \end{bmatrix} \quad (2.78)$$

Pero para eso debemos hacer uso de la matriz de transformación T de modo que transformemos el sistema original como se mostró en la ecuación 2.37 a su forma forma canónica donde:

$$\begin{aligned} \tilde{x} &= Tx, \quad T \in \mathbb{R}^{n \times n} \\ \dot{\tilde{x}} &= T\dot{x} = T(Ax + Bu) \\ &= \underbrace{TAT^{-1}}_{\tilde{A}} \tilde{x} + \underbrace{TB}_{\tilde{B}} \tilde{u} \end{aligned} \quad (2.79)$$

Por lo tanto, la matriz semejante para la que la Matriz A tengo una forma diagonal se vería de la siguiente forma:

$$\tilde{A} = TAT^{-1}$$

en donde la matriz de transformación T de n (estandarizado) contendrá vectores propios v_i [29] y se define de la siguiente manera:

$$T = \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}$$

Los vectores propios v_i cumple la condición ya mostrada en la ecuación 2.75 en donde:

$$Av_i = \lambda_i v_i = 0 \iff (A - \lambda_i I) \cdot v_i = 0$$

Asimismo, la matriz de entrada y la matriz de salida serán:

$$\tilde{B} = T^{-1}B$$

$$\tilde{C} = CT$$

Un prerequisite para la existencia de las matrices semejantes $\tilde{A}, \tilde{B}, \tilde{C}$ es que la matriz T sea regular [29].

Entonces con lo visto anteriormente se puede definir el criterio de *Controlabilidad* y *Observabilidad* de Gilbert. En cuanto al criterio de controlabilidad (citado de [28]) se dice que " el sistema $(diag \lambda_i, \tilde{B})$, cuyo modelo de espacio y estado esta en forma canónica, es completamente controlable , cuando la matriz \tilde{B} no tiene ceros en sus filas y cuando los p filas \tilde{b}_i^T de la matriz \tilde{B} que pertenecen a las variables de estado canónico de un valor propio múltiplos de p , son linealmente independientes " .

Y en cuanto al criterio de observabilidad de gilbert (citado de [28]) se puede también definir que " el sistema $(diag \lambda_i, \tilde{C})$, cuyo modelo de espacio y estado esta en forma canónica, es completamente observable , cuando la matriz \tilde{C} no tiene ceros en sus columnas y cuando los p columnas \tilde{c}_i de la matriz \tilde{C} que pertenecen a las variables de estado canónico de un valor propio múltiplos de p , son linealmente independientes " .

-Criterio de Hautus

El criterio de controlabilidad de Hautus se define (citado de [28]), .el sistema (A,B) es exactamente completamente controlable, cuando la condición

$$Rang[\lambda_i I - A \quad B] = n$$

para todos los valores propios $\lambda_i (i = 1, 2, \dots, n)$ de la matriz A es satisfecha " .

Y en cuanto al criterio de observabilidad de Hautus se define (citado de [28]), " el sistema (A,C) es exactamente completamente observable, cuando la condición

$$Rang \begin{bmatrix} \lambda_i I - A \\ C \end{bmatrix} = n$$

para todos los valores propios $\lambda_i (i = 1, 2, \dots, n)$ de la matriz A es satisfecha".

Una vez explicado los criterios se calcula los criterios de Kalman, Gilbert y Hautus en MATLAB usando los puntos de operación de la tabla 2.5.

2.2.12 Criterios de Controlabilidad y Observabilidad en MATLAB

Comenzamos por lo tanto con el criterio de Kalman. En el algoritmo 2.5 muestra la función que usaremos para hallar su controlabilidad, por medio de un función realizada en matlab en donde se arma la matriz kalmaniana de controlabilidad mostrada en la ecuación 2.76. Y en la tabla 2.6 mostraremos los resultados de la función `checkCtrbKalman(A,B)`, que hemos programado versus el comando `crtb(A,B)` propio del MATLAB.

Algoritmo 2.5: Controlabilidad de Kalman

```
1 %% Funcion Controlabilidad Kalman
2 function [ ] = checkCtrbKalman( A,B )
```

Tabla 2.6: Controlabilidad de Kalman en diferente puntos de operación

	<code>checkCtrbKalman(A,B)</code>	<code>crtb(A,B)</code>
Sistema en OP1	Controlable	Controlable
Sistema en OP2	Controlable	Controlable
Sistema en OP3	No controlable	No controlable

De la misma forma se compara la función `checkObsvKalman(A, C)`, la cual contiene la matriz kalmaniana de observabilidad de la ecuación 2.77 realizada en MATLAB y el comando `obsv(A, C)` propio de MATLAB mostrados en el algoritmo 2.6 y la tabla 2.7 respectivamente.

Algoritmo 2.6: Observabilidad de Kalman

```
1 %% Funcion Observabilidad Kalman
2 function [ ] = checkObsvKalman( A,C )
```

Continuamos con el criterio de Gilbert. En el algoritmo 2.7 muestra la función que usaremos para hallar su controlabilidad. Y en la tabla 2.8 mostraremos los resultados de la función `checkCtrbGilbert(A,B)`, que hemos programado considerando el criterio explicado de Gilbert versus el comando `crtb(A,B)` en MATLAB.

Tabla 2.7: Observabilidad de Kalman en diferente puntos de operación

	checkObsvkalman(A,C)	obsv(A,C)
Sistema en OP1	Observable	Observable
Sistema en OP2	Observable	Observable
Sistema en OP3	Observable	Observable

Algoritmo 2.7: Controlabilidad de Gilbert

```

1 %% Función Controlabilidad Gilbert
2 function [ ] = checkCtrbGilbert( A,B )

```

Tabla 2.8: Controlabilidad de Gilbert en diferente puntos de operación

	checkCtrbGilbert(A,B)	ctrb(A,B)
Sistema en OP1	Controlable	Controlable
Sistema en OP2	Controlable	Controlable
Sistema en OP3	Matriz Transformación no Regular	No controlable

Consiguientemente como el caso anterior, ahora se compara la función `checkObsvGilbert(A, C)` y el comando `obsv(A, C)` de MATLAB mostrados en el algoritmo 2.8 y la tabla 2.9 respectivamente.

Algoritmo 2.8: Observabilidad de Gilbert

```

1 %% Función Observabilidad Gilbert
2 function [ ] = checkObsvGilbert( A,C )

```

Tabla 2.9: Observabilidad de Gilbert en diferente puntos de operación

	checkObsvGilbert(A,C)	obsv(A,C)
Sistema en OP1	Observable	Observable
Sistema en OP2	Observable	Observable
Sistema en OP3	Matriz Transformación no Regular	No observable

Para el punto de operación 3 (**OP3**) debido a que posee doble valor propio por lo tanto, la matriz de transformación no es regular. Este resultado se ve para el caso de controlabilidad y observabilidad para el criterio de Gilbert.

Finalmente veremos el criterio de Hautus. En el algoritmo 2.9 muestra la función que usaremos para hallar su controlabilidad. Y en la tabla 2.10 mostraremos los resultados de la función `checkCtrbHautus(A,B)`, que hemos programado versus el comando `crtb(A,B)` en MATLAB.

Algoritmo 2.9: Controlabilidad de Hautus

```

1 %% Funcion Controlabilidad Hautus
2 function [ ] = checkCtrbHautus( A,B )

```

Tabla 2.10: Controlabilidad de Hautus en diferente puntos de operación

	checkCtrbHautus(A,B)	ctrb(A,B)
Sistema en OP1	Controlable	Controlable
Sistema en OP2	Controlable	Controlable
Sistema en OP3	No controlable	No controlable

Como ultimo punto en esta sección, ahora se compara la función `checkObsvHautus(A, C)` y el comando `obsv(A, C)` de MATLAB mostrados en el algoritmo 2.10 y la tabla 2.11 respectivamente.

Algoritmo 2.10: Observabilidad de Hautus

```

1 %% Funcion Observabilidad Hautus
2 function [ ] = checkObsvHautus( A,C )

```

Tabla 2.11: Observabilidad de Hautus en diferente puntos de operación

	checkObsvHautus(A,C)	obsv(A,C)
Sistema en OP1	Observable	Observable
Sistema en OP2	Observable	Observable
Sistema en OP3	Observable	Observable

2.2.13 Conclusiones sobre Controlabilidad y Observabilidad del Sistema linealizado

El sistema linealizado es *Observable* para los tres puntos de operación. La *Controlabilidad* sólo está garantizada para los sistemas alrededor de los puntos de operación **OP1** y **OP2**. Para el tercer punto de operación **OP3** hay una excepción, como queda claro por el criterio de Kalman, Hautus y el comando `ctrb` de MATLAB.

Físicamente esto puede explicarse de tal manera que en un ángulo de $\varphi_1 = \frac{\pi}{2}$ el primer péndulo no se puede realizar ningún movimiento en dirección x, lo que sí sería necesario a fin de equilibrar la segunda barra con un ángulo de $\varphi_2 = \pi$ ya que se espera que en este punto no haya ninguna dinámica en el sistema.

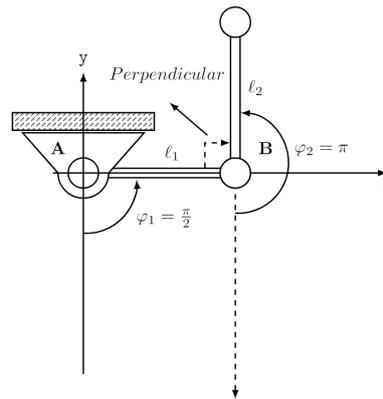


Figura 2.7: Representación de un *Sistema linearizado de péndulo doble* cuando los ángulos de los péndulos hacen un ángulo perpendicular.

Ahora qué pasaría si se consideraría la fricción a la controlabilidad y la observabilidad. La Observabilidad no se vería afectada. Sin embargo, la Controlabilidad del sistema en el tercer punto de operación se vería afectada. Un cambio de ángulo $\varphi_1 = \frac{\pi}{2} + \Delta w$ y así mismo $\varphi_1 \neq 0$, provoca que un momento sea transmitido en el punto B por la relación $M_{R2}(t) = R_{P2}(\varphi_2 - \varphi_1)$. Así pues, φ_1 también puede utilizarse para controlar φ_2 , y viceversa.

Capítulo 3

DISEÑO DE CONTROLADORES AVANZADOS PARA UN SISTEMA DE PÉNDULO DOBLE

En el capítulo anterior se analizó el modelamiento, la linealización y el análisis de la observabilidad, y controlabilidad del Sistema de péndulo doble. Este análisis previo nos ayudó entender la dinámica del sistema y con esta información poder construir las estrategias de control. En efecto, en el presente capítulo se van a probar diferentes tipos de controladores para el sistema; para eso se van asumir diferentes suposiciones a fin de tener un resultado cercano a la realidad. En la primera suposición se tiene que todas las variables son medibles. Por lo tanto, se puede diseñar un con *Control cuadrático lineal* (LQR). En el caso que sea diferentes, es decir, que alguna de las variables no sean medibles; podrá usarse un *Observador*. Posteriormente, se probará un control óptimo a fin de resolver el problema óptimo de control (OCP) para el sistema en su forma lineal y no lineal.

3.1. Diseño de un Controlador lineal cuadrático

El diseño de este controlador nos permite calcular la diferencia entre la variable controlada y (salida de la planta) y el valor final estacionario y_∞ durante todo el período de tiempo. Para eso, usar el control cuadrático de área; cuyo objetivo es minimizar y optimizar en forma rápida dichas áreas mediante la función de Costo J [24]. En la figura 3.1 se muestra la función de costo y el área que hace respuesta del sistema con su valor final. En gráfico es sacado de la fuente [24].

Las características de este control que se puede deducir de la figura 3.2 sobre el valor de la función de costo J [24]:

- Cuando más la variable de control y se aleje del valor final estacionario y_∞ , mayor será el valor de J .
- Una gran diferencia entre y y y_∞ se pondera más que las pequeñas (debido al cuadrado).

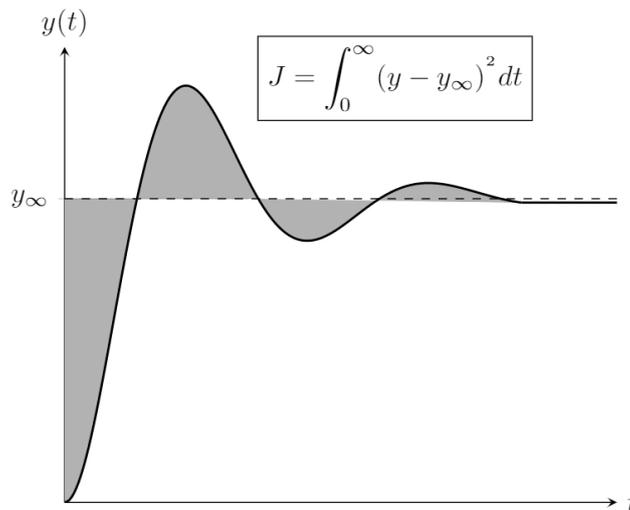


Figura 3.1: Función de costo J y el área que hace entre y y y_∞ .
Fuente : Systemdynamik und Regelungstechnik II, J. Adamy

Debido a las características antes mencionadas, el control cuadrático de área J es una gran solución para describir la calidad de una respuesta en el tiempo.

Para que la integral converga tenemos que tener en cuenta que el termino del valor final estacionario tienda a cero $y_\infty \rightarrow 0$, de otro modo tampoco se podría realizar algún tipo de control óptima si no se cumple esta condición [30]. Adicionalmente, para ponderar salida en la función de costo se utiliza una la matriz de ponderación q_y . Por lo tanto se tiene:

$$J = \int_0^{\infty} [q_y y(t)^2] dt \quad (3.1)$$

Y para el caso donde haya más salidas, la notación seria de la siguiente forma:

$$J = \int_0^{\infty} [\underline{y}(t)^T \underline{Q}_y \underline{y}(t)] dt \quad (3.2)$$

Ahora bien. nuestro objetivo es calcular la trayectoria $\underline{x}(t)$ desde un punto de inicio \underline{x}_0 en un estado final \underline{x}_∞ y no la salida. Por lo tanto, de la planta en espacio de estado se tiene $\underline{y}(t) = \underline{C} \underline{x}(t)$ y la especial caso de la matriz de ponderación $\underline{Q} = \underline{C}^T \underline{Q}_y \underline{C}$ [30] se tiene una función de costo respecto a la trayectoria $\underline{x}(t)$. Entonces se tiene:

$$J = \int_0^{\infty} [\underline{x}(t)^T \underline{Q} \underline{x}(t)] dt \quad (3.3)$$

Cabe señalar que la matriz \underline{Q} , es aquella que estabiliza el sistema con la mayor o menor variación de las variables de estado [31]. A fin de redondear la idea sobre la matriz de ponderación en mención se presenta la deducción de la misma para el caso de una variable :

$$J = \int_0^{\infty} (q_1(x_1 - x_{1\infty})^2 + q_2(x_2 - x_{2\infty})^2 + \dots + q_n(x_n - x_{n\infty})^2) dt \quad (3.4)$$

y el de más variables :

$$J = \int_0^{\infty} (\underline{x} - \underline{x}_{\infty})^T \underline{Q} (\underline{x} - \underline{x}_{\infty}) dt \quad (3.5)$$

con:

$$\underline{Q} = \begin{bmatrix} q_1 & \dots & \dots & 0 \\ \vdots & q_2 & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & \dots & q_n \end{bmatrix}$$

A fin de simplificar la ecuación se hace un cambio de coordenadas:

$$\tilde{\underline{x}} = \underline{x} - \underline{x}_{\infty}$$

Al igual que para la parte del valor final estacionario y_{∞} , el valor de la variable de estado estacionario x_{∞} también tendrá que tender a 0 a fin de lograr la convergencia de la integral se tiene $x_{\infty} = 0$. Entonces tenemos al final la integral de la función de costo en función a la trayectoria:

$$J = \int_0^{\infty} \tilde{\underline{x}}^T \underline{Q} \tilde{\underline{x}} dt \quad (3.6)$$

Y para un sistema MIMO , cual es materia de nuestro cálculo sería:

$$J = \int_0^{\infty} \underline{x}^T \underline{Q} \underline{x} dt \quad (3.7)$$

Por otro lado es necesario cuantificar la energía que se necesita durante el proceso de control; para eso, es también prescindible definir una función de costo en relación al consumo de energía, la cual depende cuadráticamente por variable manipulada u [24]. Y se define de la siguiente manera:

$$J_E = \int_0^{\infty} Ru^2 dt \quad \text{con } R=\text{const.} \quad (3.8)$$

Y en forma general para un Sistema MIMO:

$$J = \int_0^{\infty} \underline{u}^T \underline{R} \underline{u} dt \quad (3.9)$$

Finalmente para se define la ecuación para una Trayectoria o de Energía, se tiene la ecuación final para el valor de coste:

$$J = \int_0^{\infty} \underline{x}^T \underline{Q} \underline{x} dt + \underline{u}^T \underline{R} \underline{u} dt \quad (3.10)$$

El objetivo es que J sea mínima. Para lograr eso Q y R deben ser son positivos, definitivos y simétricos [30].

3.1.1 Obtención de la ley de control óptima

Proveniente de la siguiente figura donde se muestra el lazo de control cerrado se puede deducir la ley óptima de control:

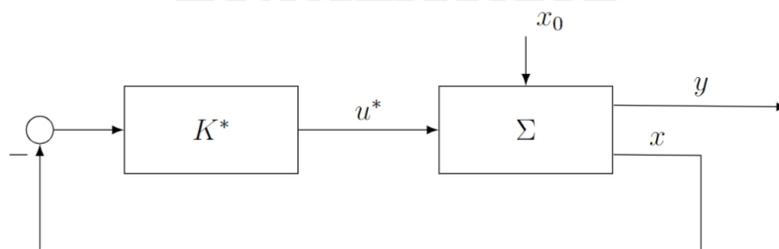


Figura 3.2: Lazo cerrado y la ley óptima de control
Fuente : Elaboración propia

Donde la ley de control tiene por lo tanto la forma:

$$\underline{u} = -\underline{K} \underline{x} \quad (3.11)$$

Para hallar el valor de K de la ley de control, se deberá adecuar la integral de la función de costo. Partimos del sistema en espacio de estado donde usaremos ley de control obtenida en la ecuación 3.11 donde:

$$\begin{aligned}\dot{\underline{x}} &= \underbrace{(A - BK)}_{\tilde{A}} \underline{x} \\ \underline{y} &= C\underline{x}\end{aligned}\quad (3.12)$$

Asimismo, en la función de costo 3.13 se inserta la ley control y se tiene lo siguiente:

$$J = \int_0^{\infty} \underline{x}^T \underbrace{(Q + K^T R K)}_{\tilde{Q}} \underline{x} dt \quad (3.13)$$

Y haciendo uso de la ecuación de Lyapunov [30]:

$$P\tilde{A} + \tilde{A}^T P = -\tilde{Q} \quad (3.14)$$

Considerando $\tilde{A} = A - BK$ y $\tilde{Q} = Q + K^T R K$ se insertan en la ecuación de Lyapunov 3.14 se obtiene la matriz óptima de control, que resulta de la solución del problema de optimización.

$$P(A - BK) + (A - BK)^T P = -(Q + K^T R K) \quad (3.15)$$

De la ecuación 3.15 se procede a obtener K mediante las derivadas de $K \frac{\partial K^T}{\partial k_{ij}}$ y $\frac{\partial K}{\partial k_{ij}}$ [30] correspondientemente se tiene :

$$\underline{K}^* = \underline{K} = \underline{K}_{opt} = \underline{R}^{-1} \underline{B}^T \underline{P} \quad (3.16)$$

y finalmente , se halla \mathbf{P} por medio de la ecuación matricial algebraica de RICCATI considerando la ecuación 3.16 en la ecuación 3.15:

$$\underline{A}^T \underline{P} + \underline{P} \underline{A} - \underline{P} \underline{B} \underline{R}^{-1} \underline{B}^T \underline{P} = -\underline{Q} \quad (3.17)$$

Una condición para que exista una solución única es que el sistema controlado sea completamente controlable [24]. Otra cosa interesante sobre este procedimiento es que los polos del sistema de control cerrado no se den por defecto, sino serán determinados por medio de las matrices **Q** y **R**. En cuanto a las variables \underline{x} y la entradas \underline{u} según su valor influirán en la función de costo.

Como se puede deducir, las ecuaciones 3.16 y 3.17 previamente halladas, son fundamentales para la obtención de la ley de control K . En efecto, para obtenerla debemos primero calcular P de la Matriz de Ricatti y este valor sustituirlo en la ecuación de la matriz K , de modo que finalmente obtengamos la ley de control óptima que buscamos [30]. A continuación, en el próximo apartado se desarrolla estos cálculos en MATLAB.

3.1.2 Diseño de un LQR en MATLAB

La parte analítica antes vista se ha sintetizado en MATLAB y será parte de nuestro desarrollo del LQR en software. Si bien existe un comando para el diseño de un controlador cuadrático lineal (LQR), construimos una función llamada `CALCULALQR` mostrada en el algoritmo 3.1 para asegurar las condiciones que requiere este controlador. Usaremos entonces las matrices **A**, **B**, **Q** y **R** para calcular esta función y asimismo serán parte de un análisis en primera instancia.

Algoritmo 3.1: Funcion LQR

```
1 function [ K, poleRK ] = calculaLQR( A,B,Q,R )
```

Como se mencionó antes, para asegurar que la función pueda ser correctamente calculada se debe cumplir condiciones como por ejemplo la controlabilidad del sistema y es mostrado en el algoritmo 3.2 a continuación:

Algoritmo 3.2: Controlabilidad

```
1 % Controlabilidad
2 Ms = ctrb(A,B);
3 if rank(Ms) == length(A)
4     disp('El sistema es completamente controlable')
5 else
6     disp('El sistema no es controlable')
```

```
7 end
```

Otra condición es las matrices \mathbf{Q} y \mathbf{R} sean *Simétricos y Definidos positivos*. El algoritmo es casi igual para ambos casos por lo que se presentara a continuación el código para la matriz para \mathbf{Q} . Estas dos condiciones se muestran en el algoritmo 2.2:

Algoritmo 3.3: Matriz simétrica y definida positiva

```
1 % Test de la simetria de Q
2 if Q' == Q
3     disp('La matriz Q es simetrica')
4 else
5     disp('La matriz Q no es simetrica')
6 end
7 % Test de la definicion positiva de Q
8 lambda_Q = eig(Q);
9 posdef_Q = 0;
10 for i=1:length(Q)
11     if lambda_Q(i) > 0
12         posdef_Q = posdef_Q + 1;
13     end
14 end
15 if posdef_Q == length(Q)
16     disp('La matriz Q es positiva definida')
17 else
18     disp('La matriz Q no es positiva definida')
19 end
```

Finalmente, una vez cumplido el análisis de las condiciones antes señaladas, se utiliza el comando LQR de MATLAB para obtener los valores de la ley de control y los polos a lazo cerrado:

Algoritmo 3.4: Calculo de la regla de control para el LQR

```
1 % Calculo de la ley de control
2 [K, poleRK] = lqr(A, B, Q, R);
```

3.1.3 Prueba del LQR

Ahora para poder simular y probar el algoritmo, debemos obtener las variables de estado ($\varphi_1, \dot{\varphi}_1, \varphi_2, \dot{\varphi}_2$) para diferentes puntos de operación. A fin de obtener dichos resultados y probarlos en nuestra función `FUNCTION [K, POLERK] = CALCULALQR(A,B,Q,R)`. Por eso se arma el modelo del sistema a lazo cerrado en SIMULINK. El modelo se puede mostrar en la figura 3.3.

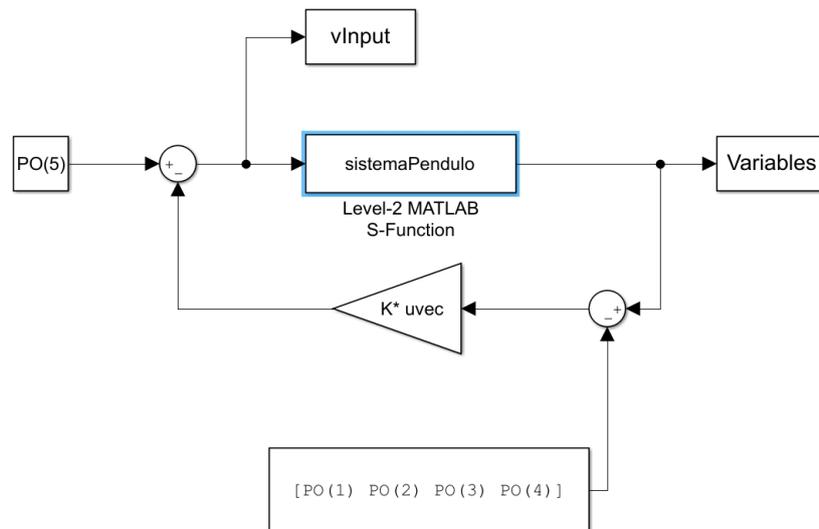


Figura 3.3: Modelo en SIMULINK del control del sistema a lazo cerrado.

Fuente : Elaboración propia

El modelo de SIMULINK de la figura 3.3 por medio del bloque S-function denominado como SISTEMAPENDULO lee la matriz de control \mathbf{K} , los datos del péndulo (g, m_1, m_2, l_1, l_2), los puntos de operación \mathbf{PO} y el estado inicial x_0 del WORKSPACE. En este modelo, las cuatro variables de estado (**Variables**) serán devueltos al WORKSPACE de la misma forma que la variable manipulada (**vInput**). Además, es muy importante considerar que no se tiene en cuenta la fricción (R_{p1}, R_{p2}).

Algoritmo 3.5: Cálculos de los POs

```

1 % Calculo de los PO (Punto de operacion)
2 x_PO = [pi/3 0 pi 0];
3 f_PO = subs(f(2), {phi1, dphi1, phi2, dphi2},
4             {x_po(1), x_po(2), x_po(3), x_po(4)});
5 M_po = solve(f_po==0, M);
6 PO = double([x_po M_po]);
7 % Condicion Inicial
8 x0 = [-pi/3 0 pi 0];

```

De el algoritmo 3.6 los PO(1), PO(2), PO(3) y PO(4) corresponden a los puntos donde se usa la linealización y se usan así mismo como los puntos de referencia o equilibrio donde se quiera que los péndulos llegen. El PO(5) se obtiene luego de hallar el punto donde el sistema está en equilibrio y se forma el vector que se usará en la 3.3. Posteriormente estos puntos OP, se usan para linealizar el modelo de la siguiente forma:

Algoritmo 3.6: Uso de las POs para la linealización del modelo

```
1 % Linealizacion
2 [ A,B,C,D ] = linealizacion( f,h,PO);
```

Aquí es también importante señalar que la matriz de control \mathbf{K} fue calculada a partir del modelo linealizado. Esto significa que la matriz de control \mathbf{K} depende de la diferencia entre el modelo linealizado y el modelo no linealizado (no lineal) que contiene el la S-function SISTEMAPENDULO ya que las variables estados que va a la retroalimentación son restados de los puntos de operación. Hay que notar que el punto de operación PO(5) corresponde al Momento M y se calcula como se vió anteriormente cuando el sistema está en equilibrio.

A fin de comprobar la linealización del modelo en SIMULINK en el siguiente punto de referencia $x_{ref} = [\frac{\pi}{3}; 0; \pi; 0]$. Se consigue entonces un Momento $M_{PO} = 0.7646$ con $x_0 = [-\frac{\pi}{3}; 0; \pi; 0]$

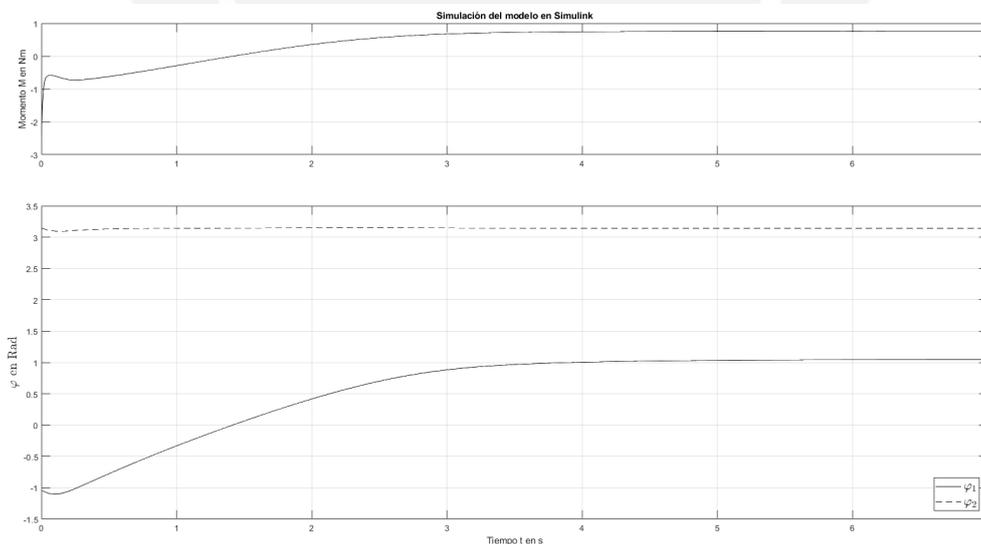


Figura 3.4: Resultado de la simulación en SIMULINK del control del sistema a lazo cerrado.

Fuente : Elaboración propia

La figura 3.4 muestra los resultados de la simulación. Aquí se puede ver que los ángulos de la variable de estado inicial seleccionados convergen con el

punto de operación linealizado en $\varphi_1 = \frac{\pi}{3}$ y $\varphi_2 = \pi$ a los 4 segundos aproximadamente. Por lo tanto, podemos decir que el modelo linealizado resulto ser exitoso así como la ley de control óptima que logra este resultado.

3.1.4 Simulación del LQR

En la parte teórica, las matrices \mathbf{Q} y \mathbf{R} juegan una parte importante para el calculo del algoritmo. De modo que la matriz \mathbf{Q} es la ponderación para las variables de estado y la matriz \mathbf{R} es la ponderación para la entrada [32]. Como la matriz \mathbf{Q} es una matriz diagonal, las variables de estado se ponderan de esta manera:

$$x_i \text{ con } \sqrt{q_{ii}} \text{ ponderado con } \mathbf{Q} = \begin{bmatrix} q_{11} & 0 & 0 & 0 \\ 0 & q_{22} & 0 & 0 \\ 0 & 0 & q_{33} & 0 \\ 0 & 0 & 0 & q_{44} \end{bmatrix} \quad (3.18)$$

Para ver la influencia de las dos matrices de ponderación, se mostrarán dos resultados de simulación a continuación. Los resultados de la simulación con $x_{PO} = \left[\frac{\pi}{3} \ 0 \ 0 \ 0 \right]$ y $x_0 = \left[-\frac{\pi}{3} \ 0 \ 0 \ 0 \right]$ así como $x_{PO} = \left[\frac{\pi}{3} \ 0 \ \pi \ 0 \right]$ y $x_0 = \left[-\frac{\pi}{3} \ 0 \ \pi \ 0 \right]$ muestran en las figuras respectivamente. Cabe señalar que cuando se usan los POs, estos señalan los puntos donde se linealizado del modelo y sirve así mismo como los puntos de referencia donde se espera que los péndulos lleguen.

La línea sólida muestra la mayor ponderación en la variable de estado φ_1 por lo tanto la línea sólida en el gráfico- φ_1 converge más rápidamente a su posición de reposo. Además, la línea sólida en el gráfico- φ_2 la que converge más lentamente.

Ahora un comportamiento diferente se muestra con la de la línea punteada. La línea de punteada es la respuesta a una ponderación más fuerte en en la variable de estado φ_2 . En este caso φ_2 converge más rápido que φ_1 .

Como se observa en la figura, que sólo el segundo ángulo en el punto de operación y en su estado inicial ha cambiado de 0 en π . Aquí no se ve una gran diferencia como en la figura 3.5. El Momento M se ajusta con el control de tal manera que el segundo péndulo no debe salir de la posición inestable

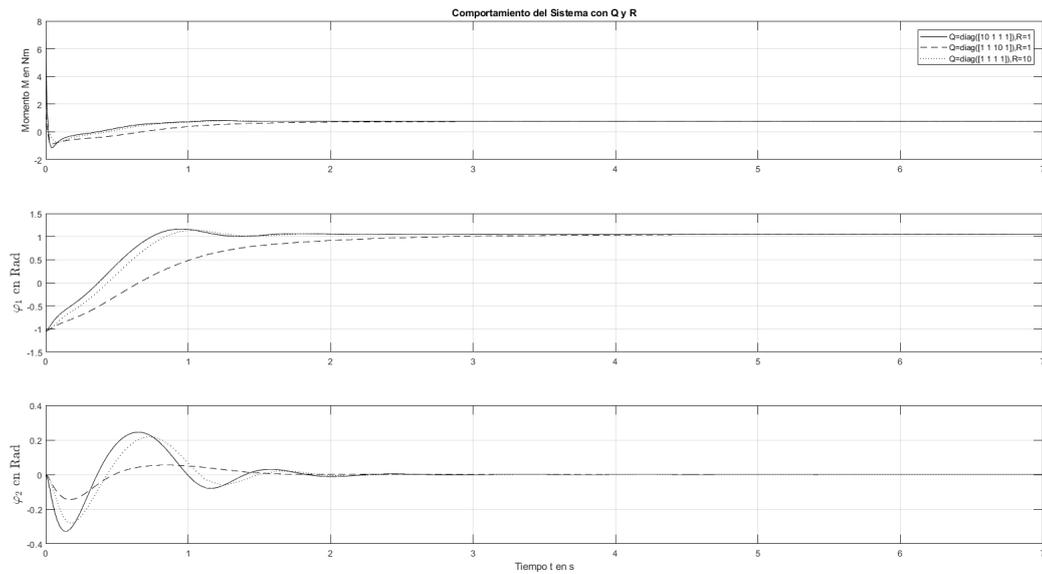


Figura 3.5: Los resultados de la simulación con $x_{PO} = \left[\frac{\pi}{3} \ 0 \ 0 \ 0 \right]$
y $x_O = \left[-\frac{\pi}{3} \ 0 \ 0 \ 0 \right]$
Fuente : Elaboración propia

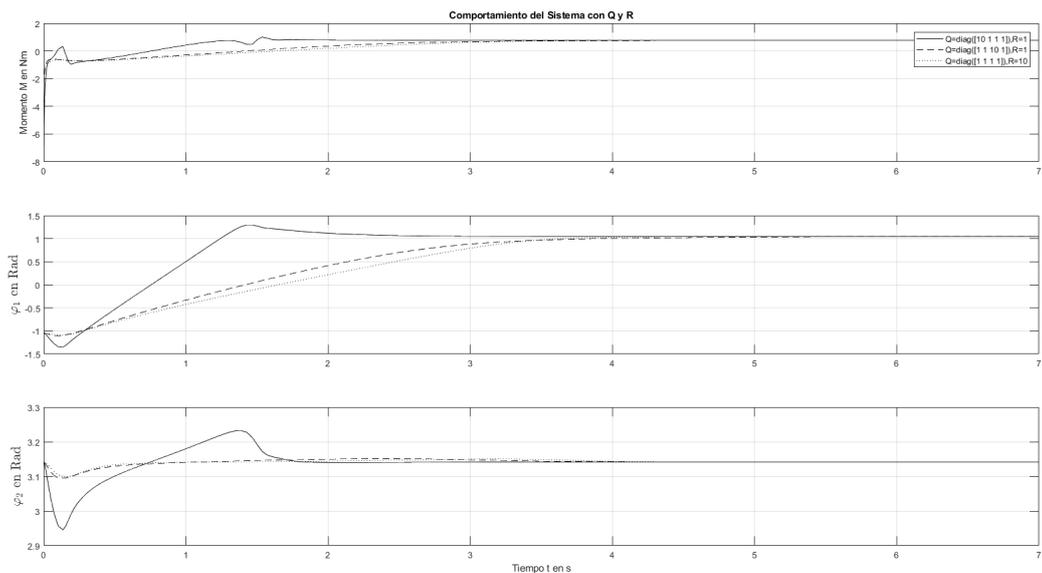


Figura 3.6: Los resultados de la simulación con $x_{PO} = \left[\frac{\pi}{3} \ 0 \ \pi \ 0 \right]$ y
 $x_O = \left[-\frac{\pi}{3} \ 0 \ \pi \ 0 \right]$
Fuente : Elaboración propia

de reposo . Por lo tanto, la única diferencia se encuentra al principio del comportamiento del Momento.

3.1.5 Prueba de las matrices **Q** y **R**

La elección de los valores de las matrices **Q** y **R** producen cambios positivos o negativos en la función de Costo. Si elegimos valores inadecuados para **Q** y **R** como por ejemplo $\mathbf{Q} = \text{diag}(100 \ 1 \ 1 \ 1)$ y $\mathbf{R} = 1$ para $x_0 = \left[-\frac{\pi}{3} \ 0 \ \pi \ 0\right]$ tendremos problemas numéricos durante la simulación. A fin de analizar la causa, se deberá calcular los valores propios del bucle de control a lazo cerrado y las variables manipuladas \underline{u} al principio de la simulación para diferentes casos y así tener mas certeza de los valores correctos para la matriz de ponderación **Q** y **R**. Se mostrará a continuación lo antes mencionado en 4 casos diferentes y estos son:

1. $\mathbf{Q} = \text{diag}([10 \ 1 \ 1 \ 1])$, $\mathbf{R} = 1$
2. $\mathbf{Q} = \text{diag}([1 \ 1 \ 10 \ 1])$, $\mathbf{R} = 1$
3. $\mathbf{Q} = \text{diag}([1 \ 1 \ 1 \ 1])$, $\mathbf{R} = 10$
4. $\mathbf{Q} = \text{diag}([100 \ 1 \ 1 \ 1])$, $\mathbf{R} = 1$

Para ellos se utiliza el siguiente algoritmo,

Algoritmo 3.7: Valores propios y variable manipulada

```

1  % Calculo de las variables manipuladas
2  u1 = PO(5) - K1 * (x0' - [PO(1); PO(2); PO(3); PO(4)]);
3  u2 = PO(5) - K2 * (x0' - [PO(1); PO(2); PO(3); PO(4)]);
4  u3 = PO(5) - K3 * (x0' - [PO(1); PO(2); PO(3); PO(4)]);
5  u4 = PO(5) - K4 * (x0' - [PO(1); PO(2); PO(3); PO(4)]);
6
7  % Calculo de los valores propios
8  lambda1 = eig(double(A - B * K1));
9  lambda2 = eig(double(A - B * K2));
10 lambda3 = eig(double(A - B * K3));
11 lambda4 = eig(double(A - B * K4));
12
13 % Matrices Q y R
14 Q4 = diag([100 1 1 1]);
15 R4 = 1;
16 [K4, r] = calculaLQR( double(A), double(B), Q4, R4 );

```

se calcula los valores propios (λ) y la variable manipulada (\mathbf{u}) al comienzo de la simulación.

$$\lambda_1 = \begin{bmatrix} -90.9471 \\ -7.19 + 2.62i \\ -7.19 - 2.62i \\ -3.1992 \end{bmatrix} \quad \lambda_2 = \begin{bmatrix} -90.9606 \\ -7.3 + 2.52i \\ -7.3 - 2.52i \\ -1.1017 \end{bmatrix} \quad \lambda_3 = \begin{bmatrix} -29.0943 \\ -6.9 + 2.43i \\ -6.9 - 2.43i \\ -1.8629 \end{bmatrix} \quad \lambda_4 = \begin{bmatrix} -90.6298 \\ -8.52 + 4i \\ -8.52 - 4i \\ -6.6699 \end{bmatrix}$$

Analizando los resultados, es interesante notar que en la parte imaginaria de los valores propios complejos y conjugados; en el cuarto caso, la parte imaginaria es significativamente mayor (4) que los otros valores (alrededor de 2.5). Esto tiene como consecuencia que el péndulo puede tener en este caso, una gran oscilación. Este efecto también puede verse en las variables manipuladas al principio de la simulación como se muestra a continuación:

$$u_1 = -6.8472, \quad u_2 = -2.4494, \quad u_3 = -1.2973, \quad u_4 = -21.1243$$

La variable manipulada en el cuarto caso es significativamente mayor que las otras, por lo que el péndulo oscila muy fuertemente en el cuarto caso. Debido a un exceso en la oscilación, el péndulo deja su rango linealizado. Dado que la matriz de control K sólo funciona alrededor del punto de operación (rango linealizado), se producen problemas numéricos ya antes mencionados. De modo que se puede concluir que valores mayores en la diagonal \mathbf{Q} pueden producir este error.

3.1.6 Prueba del LQR en un punto de operación

A fin de comparar el LQR y los demás controladores que se presentarán en los siguientes apartados, se decide por lo tanto probar el controlador LQR en un punto de operación y condiciones iniciales específicos que serán usados en todos y cada uno de los controladores. Estos parámetros son los siguientes:

$$\mathbf{Q} = \text{diag}([1 \ 1 \ 1 \ 1])$$

$$\mathbf{R} = 1$$

$$\mathbf{X}_0 = [3 \ 0 \ \pi \ 0]$$

$$\mathbf{X}_{p0} = [\pi \ 0 \ \pi \ 0] \tag{3.19}$$

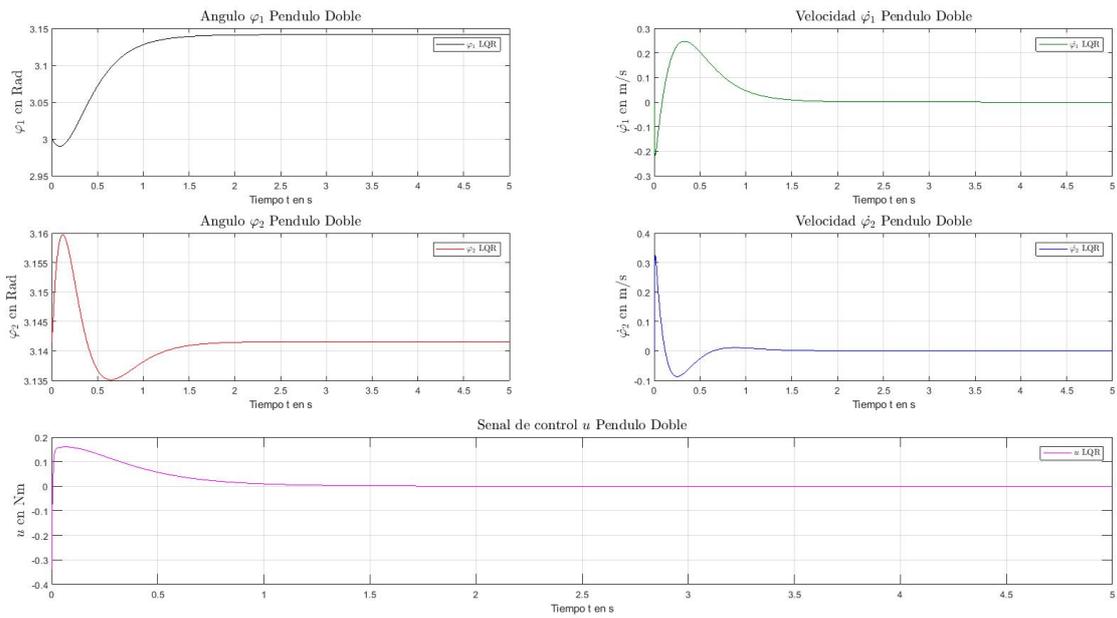


Figura 3.7: Respuesta del controlador LQR

Fuente : Elaboración propia

Como se puede observar la respuesta del controlador para el primer ángulo φ_1 converge en π a los 2.8 segundos y el segundo ángulo φ_2 a los 2.3 segundos. En cuanto a las velocidades angulares $\dot{\varphi}_1$ y $\dot{\varphi}_2$, éstos lo hacen en 2.2 y 1.7 segundos respectivamente. En cuanto a la entrada de control u que tiene un valor máximo de 0.16 Nm y a los 1.6 segundos llega a 0 Nm.

3.2. Diseño de un Observador y un Controlador con Observador de estado

En el capítulo anterior se diseñó un controlador asumiendo que todas las variables son medibles. Sin embargo, éste no sería el caso en la práctica, porque se pueden medir los ángulos de los péndulos con razonable dificultad, pero no las velocidades.

Algunas de las razones [24] por la que algunas variables no puedan ser medidas son:

- La variable a medir no es un elemento físico.
- La medición puede llevar tiempo debido a la dinámica lenta de la variable a medir.
- Es necesario realizar la medición en un laboratorio especializado.
- La técnica de medición puede ser peligrosa o el material a medir puede ser perjudicial para la persona que la realiza.
- La medición requerida implicaría un gasto elevado

En nuestro caso, una posibilidad para calcular las variables de estado faltantes sería derivar la posición y el ángulo numéricamente. Sin embargo, esto puede causar problemas debido a que podría presentarse un comportamiento diferencial como resultante de la derivación, la cual a su vez muy susceptible al ruido.

Por lo antes dicho, una segunda posibilidad y la que será parte de este capítulo es el diseño de un *Observador de Estado* para el sistema.

3.2.1 Diseño del Observador de Estado

Vamos a analizar el *Observador Luenberger* para la representación del espacio de estado. David Luenberger, creador de esta teoría, tuvo la idea de formar una estructura que contenga el modelo del sistema controlado y el sistema controlado en paralelo donde sean alimentados por una misma entrada \underline{u} pero con diferentes salidas provenientes del sistema controlado \underline{y} y el modelo del sistema controlado $\tilde{\underline{y}}$. Estas salidas posteriormente se restan ($\underline{y} - \tilde{\underline{y}}$) y formarán parte de la retroalimentación que tiene como fin minimizar la diferencia entre el modelo y el sistema controlado ajustando su valor en cada bucle de control hasta que la desviación sea casi cero. El valor a ajustar se le denomina matriz

L como se puede apreciar en la figura 3.8 [24]. En la figura también se puede apreciar la estructura antes mencionada donde en la parte superior representa el sistema de control, en la parte media representa matriz de retroalimentación del observador L (Luenberger) y en la parte inferior el observador o el modelo del sistema de control. La parte media y de abajo representan por lo tanto el observador como se puede observar en la figura con líneas punteadas. Por lo tanto para definir con claridad, la matriz Luenberger L depende de la retroalimentación que viene de la diferencia entre el valor de salida real y y el valor de salida estimado \tilde{y} y que a su vez minimiza la diferencia entre las variables de estado real y estimado $\underline{x} - \tilde{x}$ [28].

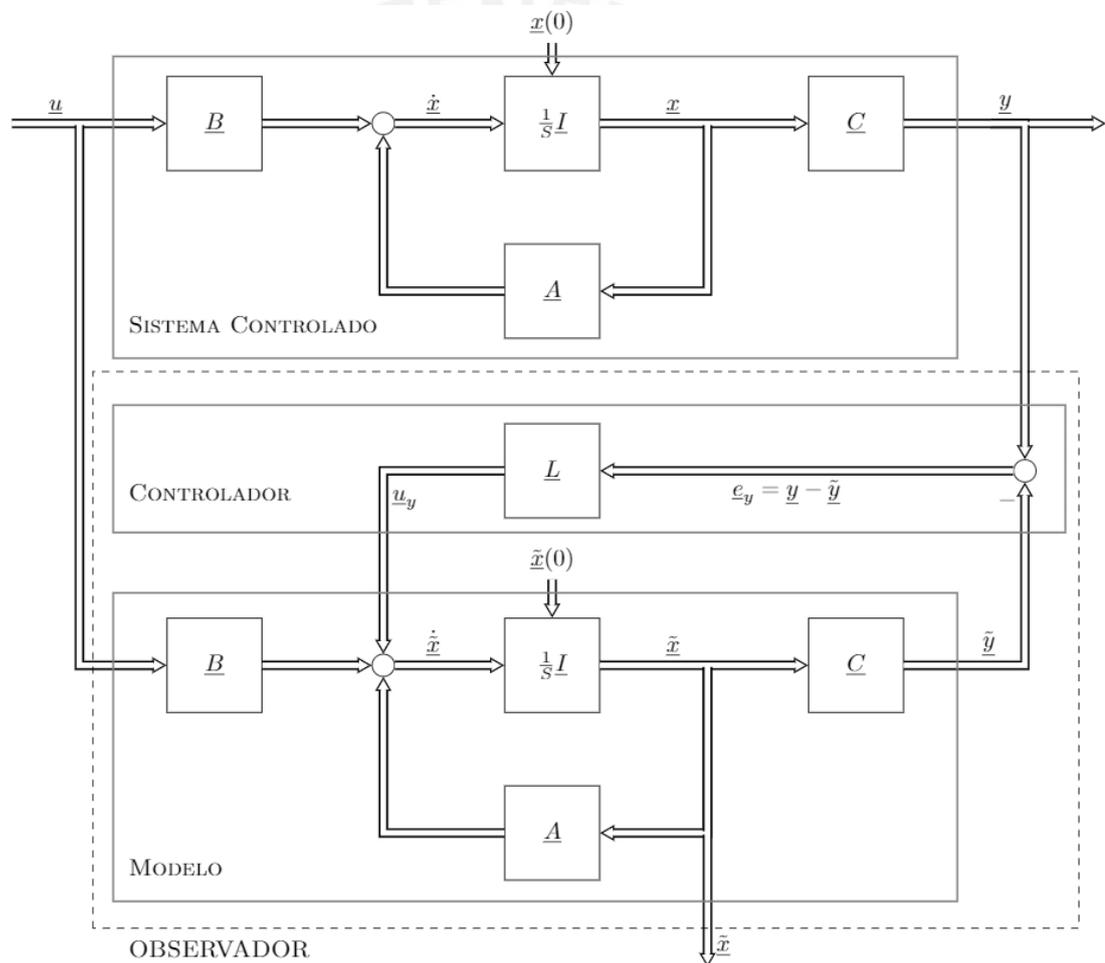


Figura 3.8: Modelo y Controlador del Observador-Luenberger

Fuente: Systemdynamik und Regelungstechnik II, J. Adamy

3.2.2 Diseño del Observador de Estado con controlador

Como se vió anteriormente, la existencia de la matriz Luenberger \mathbf{L} ayuda a que el modelo y el sistema controlado tener una diferencia mínima que equipara las salidas tanto estimada como la real. Ahora bien, a fin de eliminar/minimizar las perturbaciones y los valores iniciales no conocidos se puede agregar un controlador [24]. Para eso consideramos la diferencia de la variable de salida del sistema de control y la salida del modelo y se muestra a continuación:

$$\underline{e}_y = \underline{y} - \underline{\tilde{y}} \quad \text{con} \quad \underline{\tilde{y}} = \underline{C} \underline{\tilde{x}} \quad (3.20)$$

Entonces tenemos el Controlador proveniente de la diferencia de de las salidas multiplicado por la matriz de retroalimentación (Luenberger) \mathbf{L} de la siguiente manera [24]:

$$\underline{u}_y = \underline{L} \cdot \underline{e}_y \quad (3.21)$$

3.2.3 Convergencia del Observador

Ahora bien, a fin de mejorar el error entre el modelo y el sistema se considera un Estimador de Error de Estimación de la diferencia de las variables de estado del sistema controlado \underline{x} y la variable de estado del modelo $\underline{\tilde{x}}$, que se espera que sea cada vez menor de modo que llegue a casi cero:

$$\underline{e} = \underline{x} - \underline{\tilde{x}} \quad (3.22)$$

Posteriormente , el error de estimación se calcula utilizando las ecuaciones de estado del sistema controlado y el observador de esta manera [28] :

$$\text{Sistema Controlado :} \quad \dot{\underline{x}} = \underline{A} \underline{x} + \underline{B} \underline{u} \quad (3.23)$$

$$\begin{aligned} \text{Observador :} \quad \dot{\underline{\tilde{x}}} &= \underline{A} \underline{\tilde{x}} + \underline{B} \underline{u} + \underline{L} \underbrace{(\underline{y} - \underline{\tilde{y}})}_{\underline{C} \underline{x} - \underline{C} \underline{\tilde{x}}} \\ &= \underline{A} \underline{\tilde{x}} + \underline{B} \underline{u} + \underline{L} \underline{C} (\underline{x} - \underline{\tilde{x}}) \\ \dot{\underline{\tilde{x}}} &= (\underline{A} - \underline{L} \underline{C}) \underline{\tilde{x}} + \underline{B} \underline{u} + \underline{L} \underline{y} \end{aligned} \quad (3.24)$$

Es entonces valido que la diferencia del error de estimación:

$$\begin{aligned}\underline{\dot{e}} &= \underline{\dot{x}} - \underline{\dot{\hat{x}}} \\ &= \underline{A} (\underline{x} - \underline{\hat{x}}) + \underline{L} \underline{C} (\underline{x} - \underline{\hat{x}})\end{aligned}\quad (3.25)$$

De modo finalmente se tiene el error de estimación del observador, con $\underline{e} = \underline{x} - \underline{\hat{x}}$, sea:

$$\underline{\dot{e}} = (\underline{A} - \underline{L} \underline{C}) \cdot \underline{e} \quad (3.26)$$

Una parte muy importante a tomar en cuenta es el error de estimación del observador sea nulo. Esto se logra cuando $t \rightarrow \infty$, por eso es importante que los valores propios de la matriz $(\mathbf{A} - \mathbf{L}\mathbf{C})$ tienen una parte real negativa [28].

$$\lim_{t \rightarrow \infty} \|e\| = 0 \quad (3.27)$$

Por eso se espera que el error no solo disminuya o sea casi nulo, sino también que sea rápido. Estas características se logran cuando los valores propios del observador se encuentran en el lado izquierdo de los valores propios dominantes, es decir de los valores propios del sistema a lazo cerrado. Por lo tanto los valores propios no son elegidos aleatoriamente sino se considera que sean 2 a 10 veces los valores propios de sistema a lazo cerrado. Sin embargo, cuando mas lejos tengo los valores propios del observador si bien disminuye rápidamente el error del observador entre el sistema y el observador; hay una sensibilidad grande al ruido [28]. Este punto se estudiará en la parte de simulación probando diferentes valores propios.

3.2.4 Elección de la matriz de retroalimentación L

El calculo del vector del observador \underline{L} se hace por medio de *Ackerman* para el diseño del controlador mediante la inserción de valores propios o posicionamiento convenientemente de los mismos en el plano real-imaginario para cualquier sistema de matrices (parte real negativa) [24].

$$\underline{L} = (\tilde{a}_0 \underline{I} + \tilde{a}_1 \underline{A} + \dots + \tilde{a}_{n-1} \underline{A}^{n-1} + \underline{A}^n) \underline{m}_B \quad (3.28)$$

$$\text{con } \underline{m}_B = \underline{M}_B^{-1} \cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \text{ y } \tilde{a}_i = a_i + l_{i+1} \quad (3.29)$$

En donde los coeficientes \tilde{a}_i son los coeficientes del polinomio característico del Observador.

3.2.5 Diseño del Observador de Estado con Controlador de Estado

Una vez que se conocen todas las variables de estado o se estiman por medio del observador, se puede diseñar un Controlador de Estado. La figura 3.9 muestra al Observador y así también al Controlador de estado \mathbf{K} , el cual se multiplica con el vector de estado estimado $\underline{\tilde{x}}$ proveniente del Observador. Este producto se retroalimenta e ingresa nuevamente al por medio de la diferencia con la entrada w de modo que este sea la nueva entrada u . Esta entrada vuelve a alimentar al sistema controlado y al modelo respectivamente; cuyo objetivo es dirigir el sistema a su punto de equilibrio. Por lo tanto, hasta ahora tenemos (tomado del [24]):

$$\text{Sistema Controlado} \begin{cases} \dot{\underline{x}} = \underline{A} \underline{x} + \underline{B} \underline{u} \\ \underline{y} = \underline{C} \underline{x} \end{cases} \quad (3.30)$$

$$\text{Estimación del Error} \begin{cases} \underline{e} = \underline{x} - \underline{\tilde{x}} \\ \dot{\underline{e}} = (\underline{A} - \underline{L} \underline{C}) \underline{e} \end{cases} \text{ e del Observador} \quad (3.31)$$

$$\text{Controlador de Estado} \begin{cases} \underline{u} = \underline{w} - \underline{K} \underline{\tilde{x}} \end{cases} \quad (3.32)$$

Y la combinación Observador y controlador de estado proveniente de la ecuación 3.24 y 3.32 (tomado del [24]):

$$\text{Observador con} \begin{cases} \dot{\underline{\tilde{x}}} = (\underline{A} - \underline{B} \underline{K} - \underline{L} \underline{C}) \underline{\tilde{x}} + \underline{B} \underline{w} + \underline{L} \underline{y} \end{cases} \text{ Controlador de Estado} \quad (3.33)$$

3.2.6 Teorema de separación

Cabe hacer una pregunta si los valores propios del control a lazo cerrado influenciarían a los valores propios del observado. Según el enunciado el teorema de separación, si el sistema a lazo abierto (A,B,C) completamente observable y controlable es, los valores propios del sistema a lazo cerrado (sin observador) y los valores propios del observador serán independientes uno del otro [33].

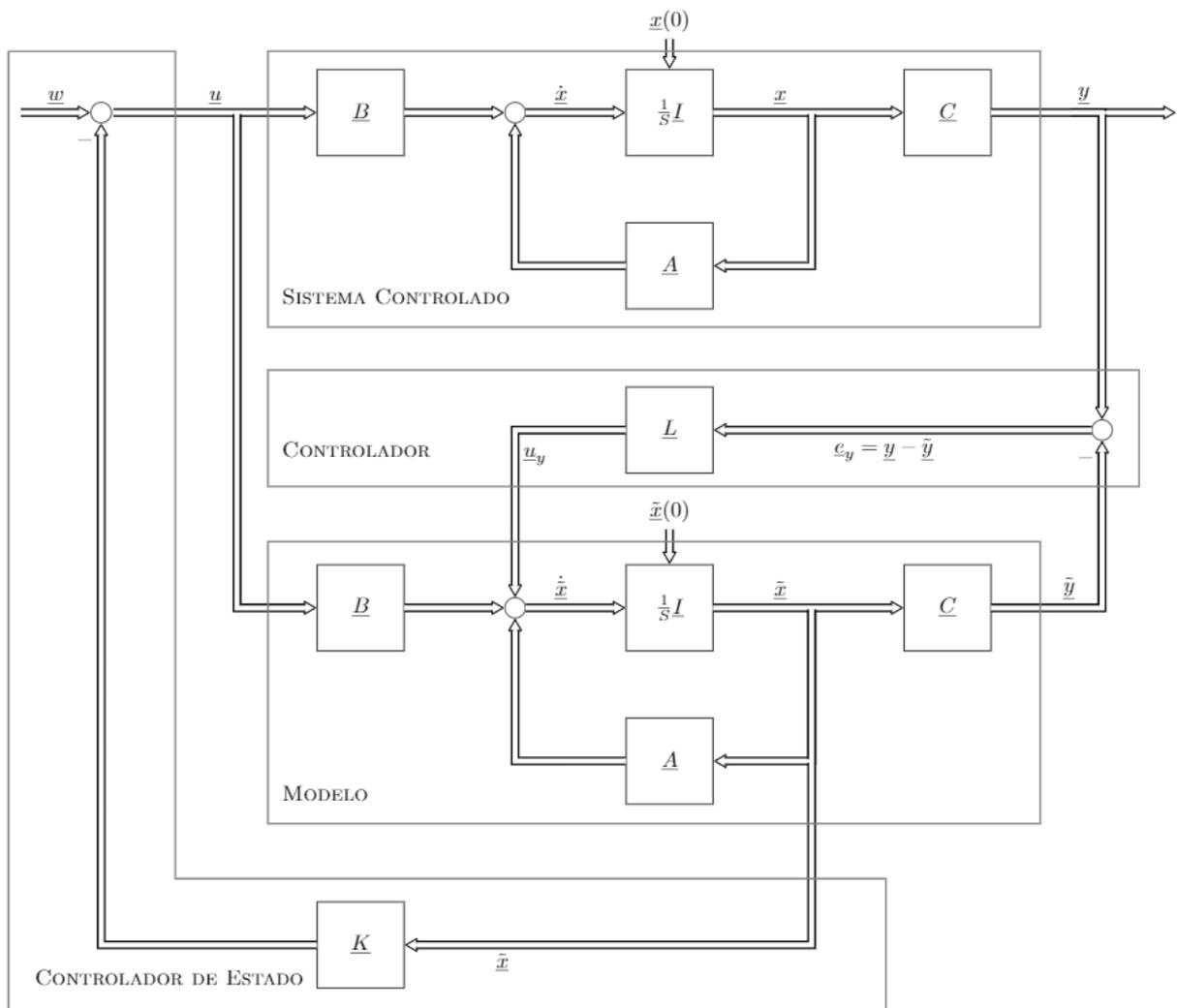


Figura 3.9: Observador y Controlador de Estado
Fuente: Systemdynamik und Regelungstechnik II, J. Adamy

De las ecuaciones 3.30, 3.31 y 3.32 pueden sintetizar en sistema considerando los componentes de los vectores de estado \underline{x} y \underline{e} para todo el sistema, estas ecuaciones dan como resultado la siguiente descripción del sistema:

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{e}} \end{bmatrix} = \begin{bmatrix} \underline{A} - \underline{B}\underline{K} & \underline{B}\underline{K} \\ \underline{0} & \underline{A} - \underline{L}\underline{C} \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{e} \end{bmatrix} + \begin{bmatrix} \underline{B} \\ \underline{0} \end{bmatrix} \underline{w}$$

$$\underline{y} = \begin{bmatrix} \underline{C} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{e} \end{bmatrix} \quad (3.34)$$

Valores propios del Observador

Como se puede observar en el sistema de matrices 3.34, es una matriz triangular donde elementos por encima o por debajo de su diagonal principal son cero. Los valores propios para el observador y del sistema de control, por lo tanto no influyen una a la otra. Así pues, el sistema total es estable, siempre que el observador y el sistema de control cerrado sin observador son respectivamente estables [29].

Debido al teorema de separación se puede diseñar el controlador de estado con observador. Para eso, se deberán seguir los siguientes pasos (citado del [24]):

1. Se diseña en primera instancia el controlador de estado sin el observador. Se entiende que como no medimos todas las variables estas deberían venir de la parte de la estructura del Observador Luenberger. Cuya estructura sera parte de la retroalimentación del controlador de estado.
2. Luego, se añade entonces al lazo de control, el Observador como parte de su retroalimentación. Cabe señalar que los valores propios del observador deberán tener un valor (negativo) y más pequeño en referencia a la parte real de los valores propios del sistema de control a lazo cerrado. Es decir, los polos del observador deberán ser mas rápidos que el de los polos a lazo cerrado.

3.2.7 Diseño del Observador de Estado con controlador de estado en MATLAB y SIMULINK

Después de haber visto la parte teórica, se arma entonces el modelo del Observador en SIMULINK en donde se estimará las variables del sistema en base al modelo del sistema como se muestra en la figura 3.10.

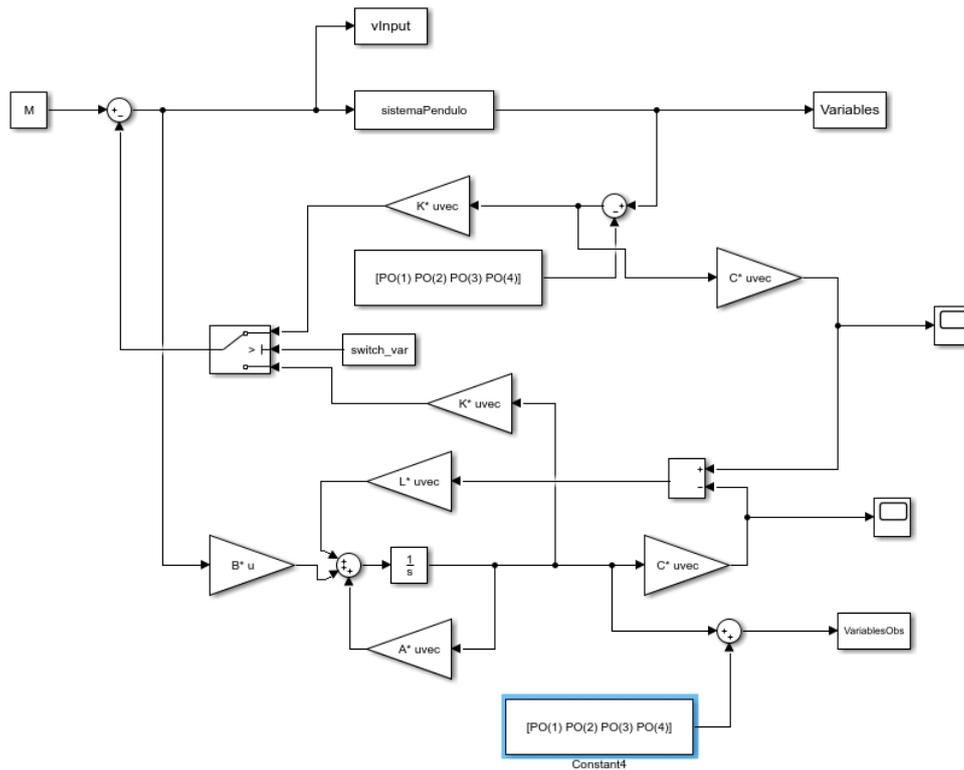


Figura 3.10: Modelo y controlador del Observador

Fuente : Elaboración propia

Se puede observar que el SistemaPendulo es una S-Function Level 2, donde como salida se tienen las cuatro variables $\varphi_1, \dot{\varphi}_1, \varphi_2, \dot{\varphi}_2$ explicado en el capítulo 2. Según la figura 3.9 se necesita la salida Y del modelo para ser comparado con la salida del Observador del que se forma a través de las matrices A, B y C . De la misma forma se tiene al controlador Luenberger L en la parte media y el controlador de estado. Este controlador de estado por el teorema de separación se hallan los valores de cada una independientemente al del Observador. En el medio de ambos modelos hay un switch donde se puede seleccionar simular con o si el observador. Para la simulación, se probará la influencia de los valores propios del Observador con los valores propios del sistema a lazo cerrado.

Los valores del Observador con controlador, los valores iniciales y el punto de operación están preestablecidos.

$$\begin{aligned}
 \mathbf{Q} &= \text{diag}([1 \ 1 \ 1 \ 1]) \\
 \mathbf{R} &= 1 \\
 \mathbf{X}_0 &= [3 \ 0 \ \pi \ 0] \\
 \mathbf{X}_{Obs} &= [\pi \ 0 \ \pi \ 0] \\
 \mathbf{X}_{PO} &= [\pi \ 0 \ \pi \ 0]
 \end{aligned} \tag{3.35}$$

Para determinar la influencia de los valores propios del observador y del sistema controlado en los variables estimados, se considera los valores propios del sistema de control a lazo cerrado en su conjunto. En los tres gráficos por ejemplo, los valores propios del observador se multiplicaran 1 vez, 2.5 veces y 10 veces los polos del sistema. De allí se definirá cual de ellos presenta el mejor resultado.

En cuanto al modelo en SIMULINK se simulará por un intervalo de 5 segundos de donde se tomarán las cuatro variables estimadas del Observador (VariablesObs) y las variables provenientes Sistema del Péndulo doble (Variables).

A continuación se muestra el algoritmo para el calculo del observador en el algoritmo 3.8 y Los plots de de las variables φ_1 y φ_2 del Sistema y del Observador.

Algoritmo 3.8: Algoritmo del Observador

```

1  %Calculo del Observador 2.5 veces los polos del sistema
2  polObs1 = 2.5*[poleRK(1) poleRK(2) poleRK(3) poleRK(4)];
3  [L] = calculaObs(A,C,polObs1);
4  sim('sim_system_pendell')
5  vT1 = Variables.Time;
6  mX1 = [Variables.Data(:,1) Variables.Data(:,3)];
7  mXobs1 = [VariablesObs.Data(:,1) VariablesObs.Data(:,3)];
8
9  %Calculo del Observador 10 veces los polos del sistema
10 polObs2 = 10*[poleRK(1) poleRK(2) poleRK(3) poleRK(4)];
11 [L] = calculaObs(A,C,polObs2);
12 sim('sim_system_pendell')
13 vT2 = Variables.Time;
14 mX2 = [Variables.Data(:,1) Variables.Data(:,3)];
15 mXobs2 = [VariablesObs.Data(:,1) VariablesObs.Data(:,3)];

```

```

16
17 %Calculo del Observador 1 vez los polos del sistema
18 polObs3 = [poleRK(1) poleRK(2) poleRK(3) poleRK(4)];
19 [L] = calculaObs(A,C,polObs3);
20 sim('sim_system_pendell')
21 vT3 = Variables.Time;
22 mX3 = [Variables.Data(:,1) Variables.Data(:,3)];
23 mXobs3 = [VariablesObs.Data(:,1) VariablesObs.Data(:,3)];

```

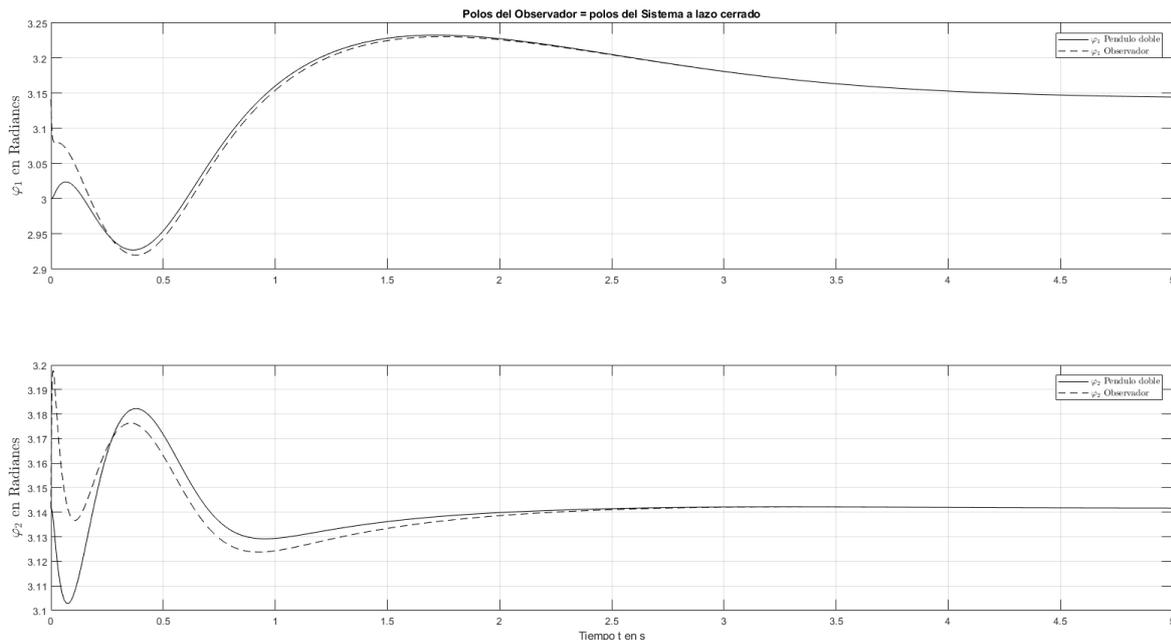


Figura 3.11: Observador con 1 vez los polos del Sistema a lazo cerrado

Fuente : Elaboración propia

Los tres ejemplos que se mostraron antes, nos dan a entender que cuanto mayor es la magnitud de los valores propios del Observador en relación a los valores propios del Sistema controlado a lazo cerrado, se mejora el tiempo de establecimiento, el "*Overshoot*", decrece por mencionar algunas cosas mejoras. Veamos los resultados más a detalle según los valores del Observador y el Sistema controlado, en la figura 3.11 se muestra que los valores propios tanto de sistema controlado a lazo cerrado son iguales a los valores propios del Observador. Sin embargo, existe un desfase entre ambas respuestas tanto para el ángulo φ_1 y φ_2 . También se puede observar que a pesar en las condiciones iniciales en comparación con los valores esperados para el Observador, convergen cerca a los 4.7 segundos y 3.8 segundos respectivamente. En la figura 3.12 muestra una mejora sustancial en su respuesta con una convergencia de casi 3.6 segundos y 1.7 segundos para φ_1 y φ_2 respectivamente, considerando

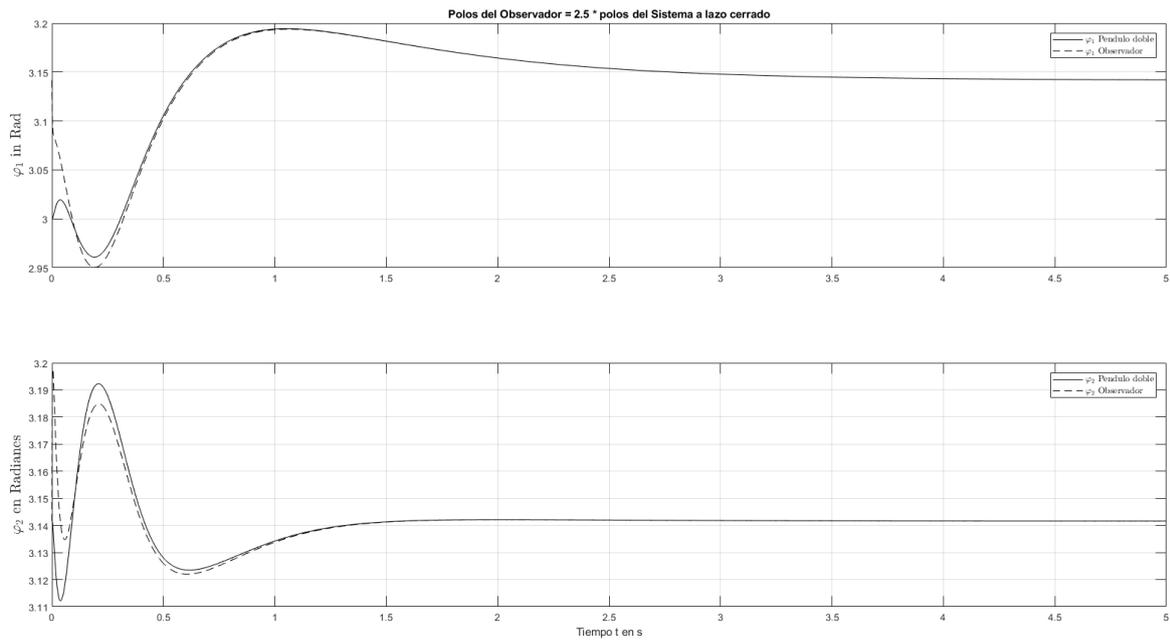


Figura 3.12: Observador con 2.5 veces los polos del Sistema a lazo cerrado
Fuente : Elaboración propia

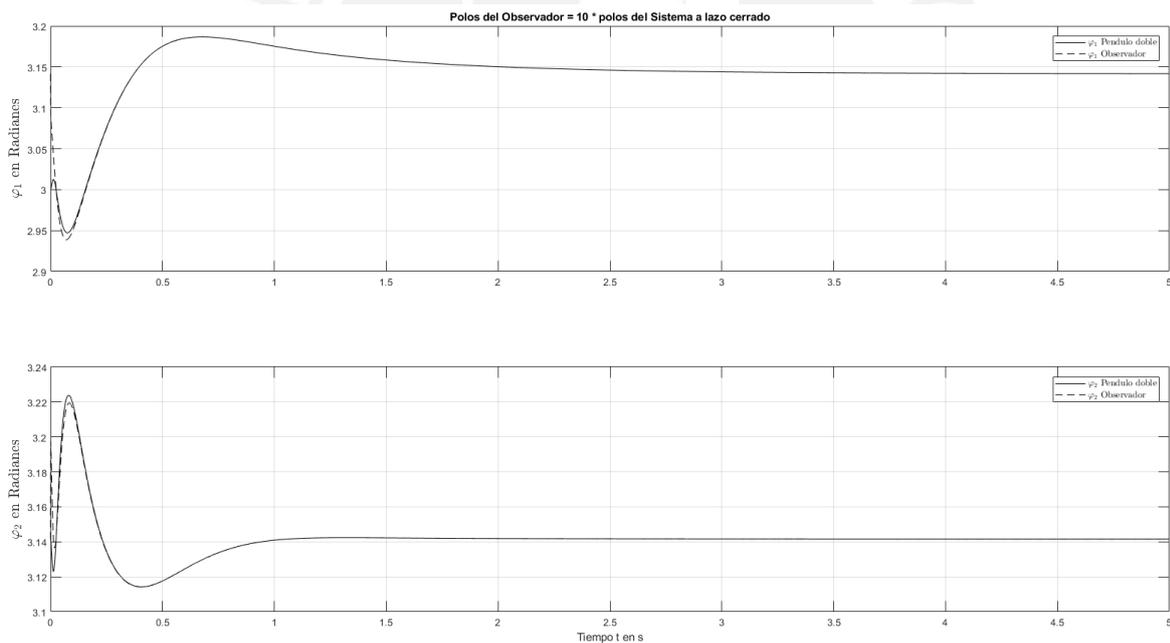


Figura 3.13: Observador con 10 veces los polos del Sistema a lazo cerrado
Fuente : Elaboración propia

que los valores propios del Observador son de 2.5 veces que el del Sistema a lazo cerrado. Por último, la figura 3.13 muestra una mejora aún mejor que de la figura anterior ya que el tiempo de respuesta llega a menor de los 3.2 segundos y 1.2 segundos para φ_1 y φ_2 respectivamente. Ahora bien como se

mencionó antes, en general, el valor que asignamos a los valores propios de los Observadores deberán ser aproximadamente de dos a seis veces o incluso hasta 10 veces los valores propios del sistema a lazo cerrado según la parte teórica y en los resultados se muestran que este anunciado es útil a la hora de elegir los valores de los valores propios. Sin embargo, tenemos que tener cuidado ya que si elegimos los valores muy a la izquierda, si pensamos en el plano real-imaginario, es decir valores reales negativos cercanos al $-\infty$, el Observador adquiere entonces un carácter cada vez más diferenciador y por lo tanto inestable. El comportamiento del sistema podría aún mejorarse introduciendo un observador reducido que no necesita estimar todas las variables sino las que no se miden.

3.3. Diseño de un Observador Reducido y un Controlador de estado

El Observador reducido surge de la idea que no es necesario estimar todas las variables de estado si ya de por sí algunas se pueden medir o en su caso más complejo, calcularlas ya que se pueden derivar de alguna variable de estado o en su defecto obtenerlas desde la salida del sistema [29]. Por lo tanto no es necesario implementar un Observador para cada variable no medida como se vió en el apartado anterior, de allí el nombre de observador reducido. Con esto, se puede hacer que los cálculos y costo computacional sea menor. Veamos a continuación cómo se puede implementar dicho Observador reducido; comenzamos a identificar las variables que sí son medibles. Desde esta premisa, esta variable medible se considera como salida \underline{y} y de allí se procede a medir solo la salida y calcular las demás variables de estado. Pongamos el siguiente ejemplo [24], si tenemos la única variable medible a el vector de estado \underline{x}_1 , entonces se procede a asignar esta misma como salida y_i como se muestra a continuación:

$$\underline{y} = \underline{C}\underline{x} \quad \longrightarrow \quad \underline{y} = \underline{x}_1$$

Veamos entonces cómo se procede; tenemos a continuación la siguiente sistema de ecuación de estado y se separa los términos de la matriz \underline{A} así como el la matriz \underline{B} de la siguiente manera (tomado del [24]):

$$\begin{bmatrix} \dot{\underline{x}}_1 \\ \dot{\underline{x}}_2 \end{bmatrix} = \begin{bmatrix} \underline{A}_{11} & \underline{A}_{12} \\ \underline{A}_{21} & \underline{A}_{22} \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \end{bmatrix} + \begin{bmatrix} \underline{B}_1 \\ \underline{B}_2 \end{bmatrix} \begin{bmatrix} \underline{u} \end{bmatrix} \quad (3.36)$$

Desarrollamos el sistema en primera instancia en ecuaciones diferenciales así:

$$\begin{aligned}\dot{\underline{x}}_1 &= \underline{A}_{11} \underline{x}_1 + \underline{A}_{12} \underline{x}_2 + \underline{B}_1 \underline{u} \\ \dot{\underline{x}}_2 &= \underline{A}_{21} \underline{x}_1 + \underline{A}_{22} \underline{x}_2 + \underline{B}_2 \underline{u}\end{aligned}\tag{3.37}$$

Recordemos que la variable de estado medible es el vector \underline{x}_1 , por lo tanto la variable de estado desconocida es el vector \underline{x}_2 . Entonces ordenamos el sistema separando los vectores de estado conocidos y desconocidos por separado. Haciendo cambio de vector de variables tanto para \underline{x}_2 y $\dot{\underline{x}}_2$ por \underline{y}_2 y \underline{v} respectivamente. Entonces tenemos las siguientes ecuaciones que formarían el Observador reducido (tomado del [24]):

$$\begin{aligned}\dot{\underline{x}}_1 &= \underline{A}_{11} \underline{x}_1 + \underline{y}_2 + \underline{B}_1 \underline{u} \\ \underline{v} &= \underline{A}_{21} \underline{x}_1 + \underline{B}_2 \underline{u}\end{aligned}\tag{3.38}$$

$$\begin{aligned}\dot{\underline{x}}_2 &= \underline{A}_{22} \underline{x}_2 + \underline{v} \\ \underline{y}_2 &= \underline{A}_{12} \underline{x}_2\end{aligned}\tag{3.39}$$

La ecuación 3.39 describe la parte desconocida de \underline{x}_2 del vector de estados y \underline{v} a estimador integrando para este fin a la matriz de retroalimentación (Luenberger) \underline{L} , de modo que tendríamos la siguiente ecuación para el Observador reducido Luenberger:

$$\begin{aligned}\dot{\tilde{\underline{x}}}_2 &= \underline{A}_{22} \tilde{\underline{x}}_2 + \underline{v} + \underline{L} \cdot (\underline{y}_2 - \tilde{\underline{y}}_2) \\ &= (\underline{A}_{22} - \underline{L} \underline{A}_{12}) \tilde{\underline{x}}_2 + \underline{v} + \underline{L} \underline{y}_2\end{aligned}\tag{3.40}$$

La estructura del Observador Luenberger reducido entonces tendría la siguiente forma:

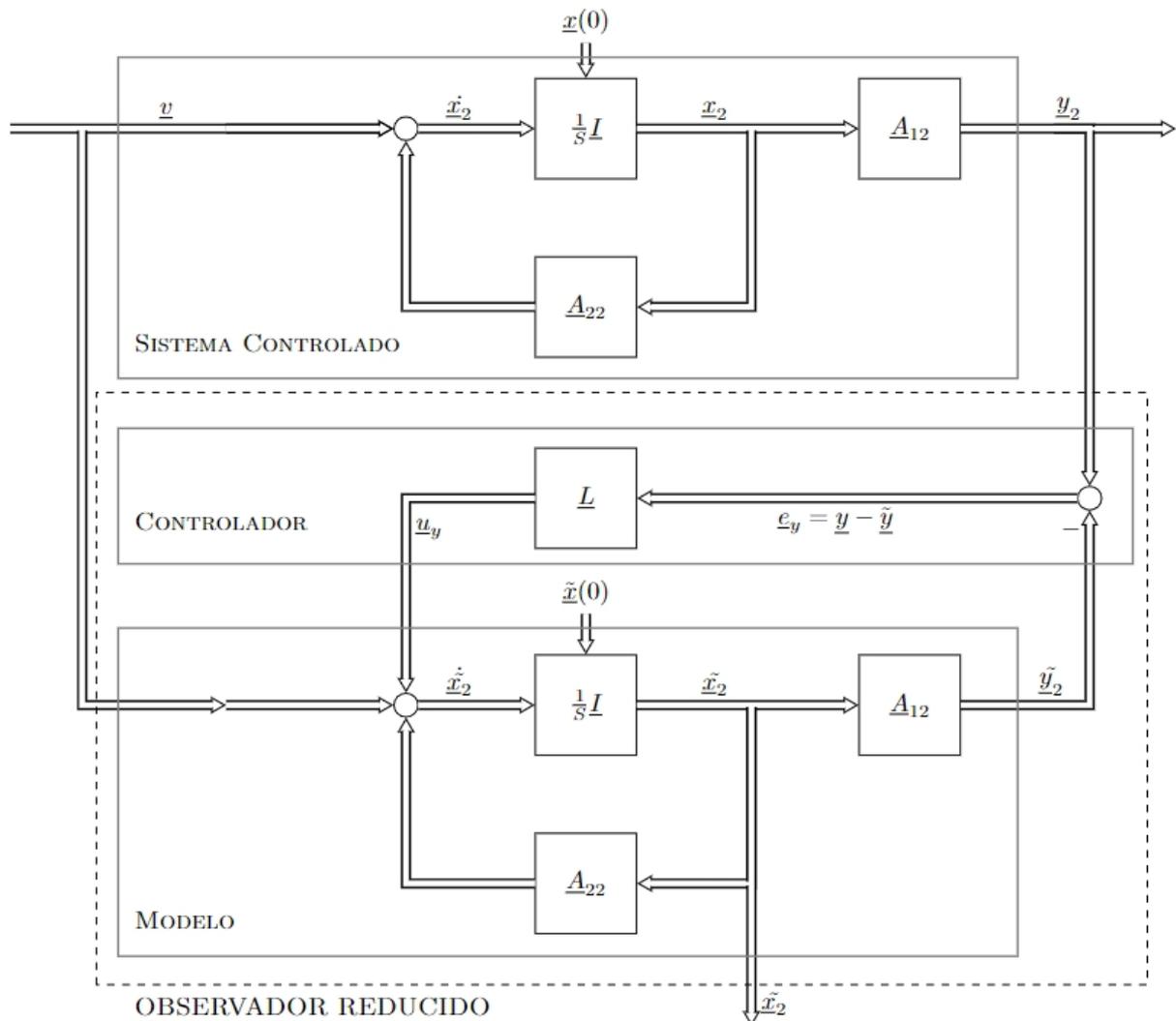


Figura 3.14: Observador Luenberger reducido
Fuente : Systemdynamik und Regelungstechnik II, J. Adamy

3.3.1 Diseño del Observador Reducido de Estado con controlador de estado en MATLAB y SIMULINK

Ahora se procede a implementar el observador reducido según la nueva estructura donde las matrices son reducidas. En esta oportunidad asumimos que solo 2 variables son medibles. Es decir las variables de posición φ_1 y φ_2 son las que se considerarían medibles y las variables de velocidad las variables no medibles. La idea del diseño y simulación del Observador Reducido será analizar el tiempo de computo y la performance numérica de las variables manipuladas es decir φ_1 y φ_2 , las variables observadas $\hat{\varphi}_1$ y $\hat{\varphi}_2$ si bien son las que forman parte del algoritmo se busca si existe mejora en las variables

manipulas por esa razón se muestra dichos en resultados en las siguientes figuras. A continuación mostramos el bloque en simulink y posteriormente explicaremos los resultados.

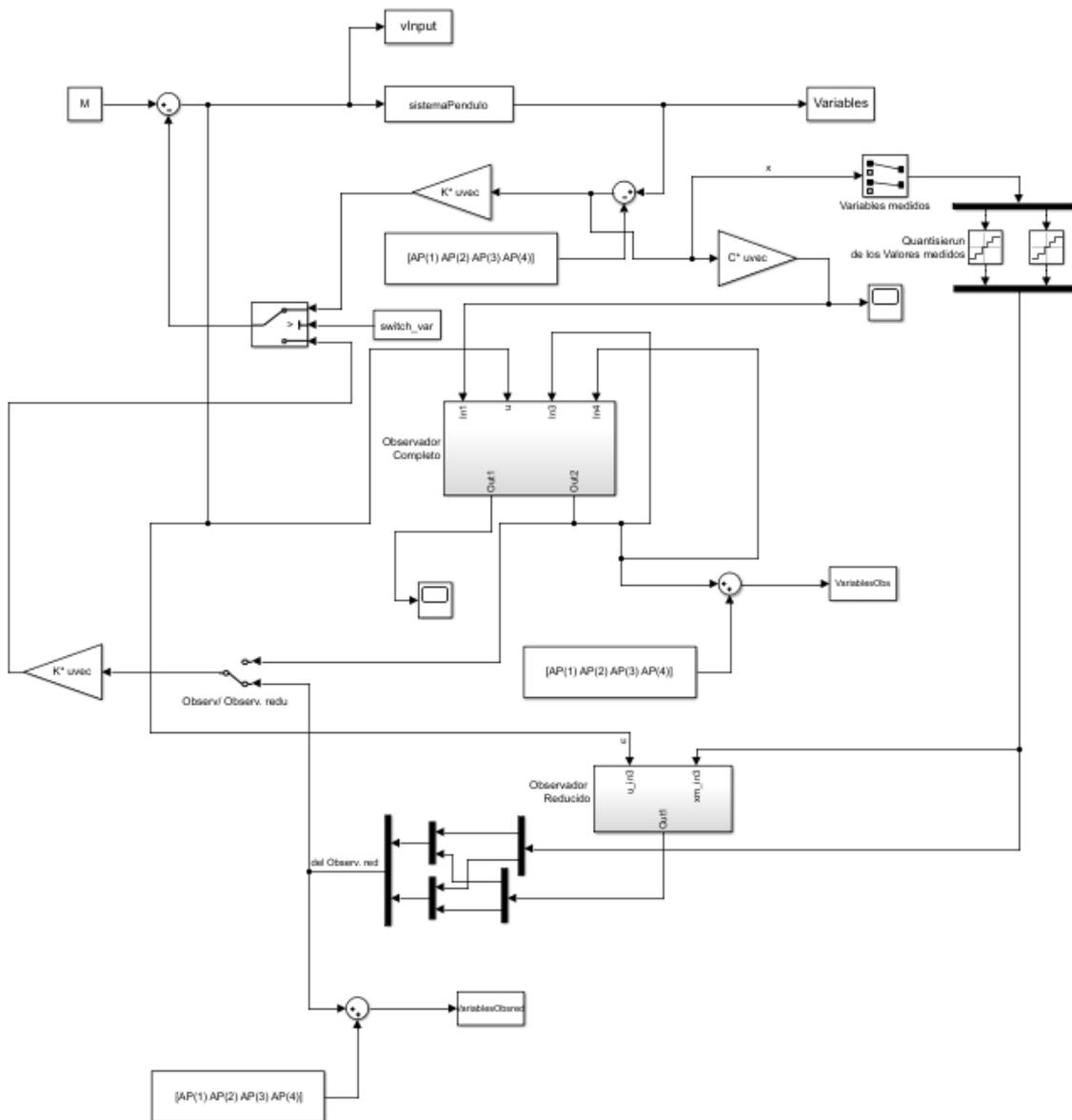


Figura 3.15: Controlador de Estado con Observador Luenberger reducido
Fuente : Elaboración propia

La estructura del Observador reducido en MATLAB se muestra a continuación con los valores respectivos se muestra en la figura 3.16

Los resultados se comparan entre el observador reducido y el observador completo. Cabe mencionar que se mantienen mostrados de los valores para

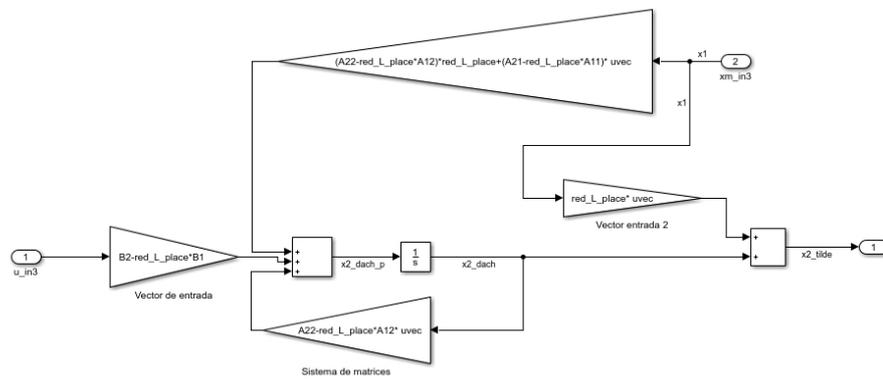


Figura 3.16: Estructura del Observador reducido en bloques en SIMULINK

Fuente : Elaboración propia

condiciones iniciales, punto de operación para el Sistema y el Observador de las ecuaciones 3.73. Los resultados se muestran a continuación:

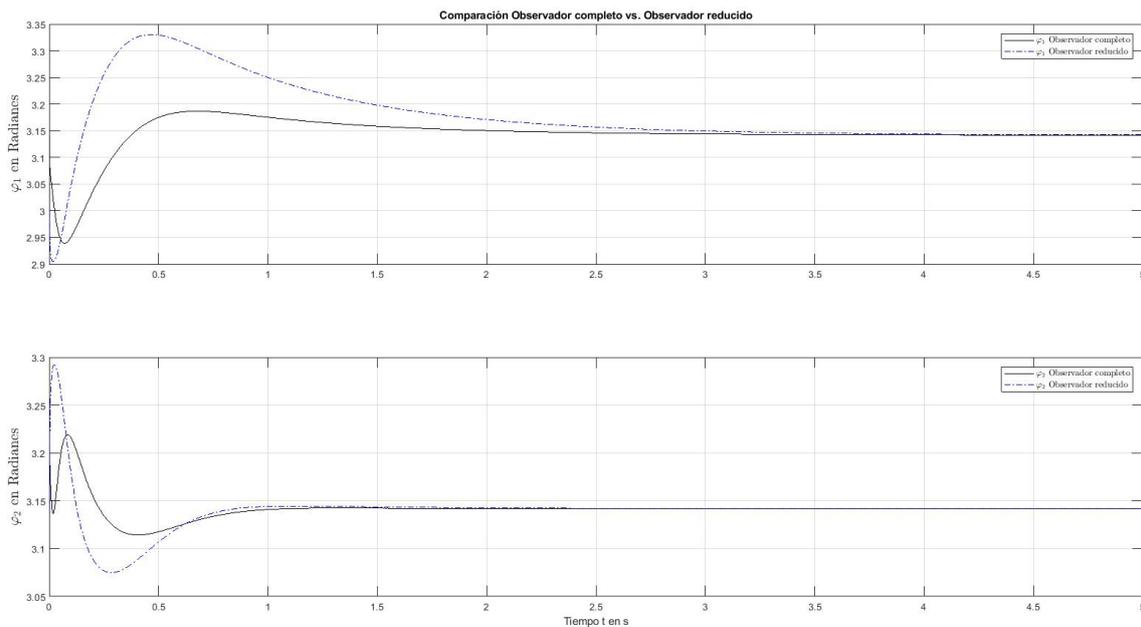


Figura 3.17: Observador completo vs. Observador reducido

Fuente : Elaboración propia

De los resultados vistos en la figura 3.17 que aunque convergen en los valores esperados tienen una diferencia marcada. En efecto, el tiempo de establecimiento y el "Overshoot" tienen diferente performance; sin embargo, debemos tener en cuenta que el Observador completo requiere más calculo y tiempo para dar una respuesta ya que estima todas las variables y en el Observador reducido no. Este característica se puede ver en la respuesta de cada

ángulo en su parte inicial. Veamos un poco más de cerca este aspecto a considerar en la figura 3.18, los valores iniciales de ambos observadores (reducido y completo) muestran una diferencia significativa como se ve a continuación:

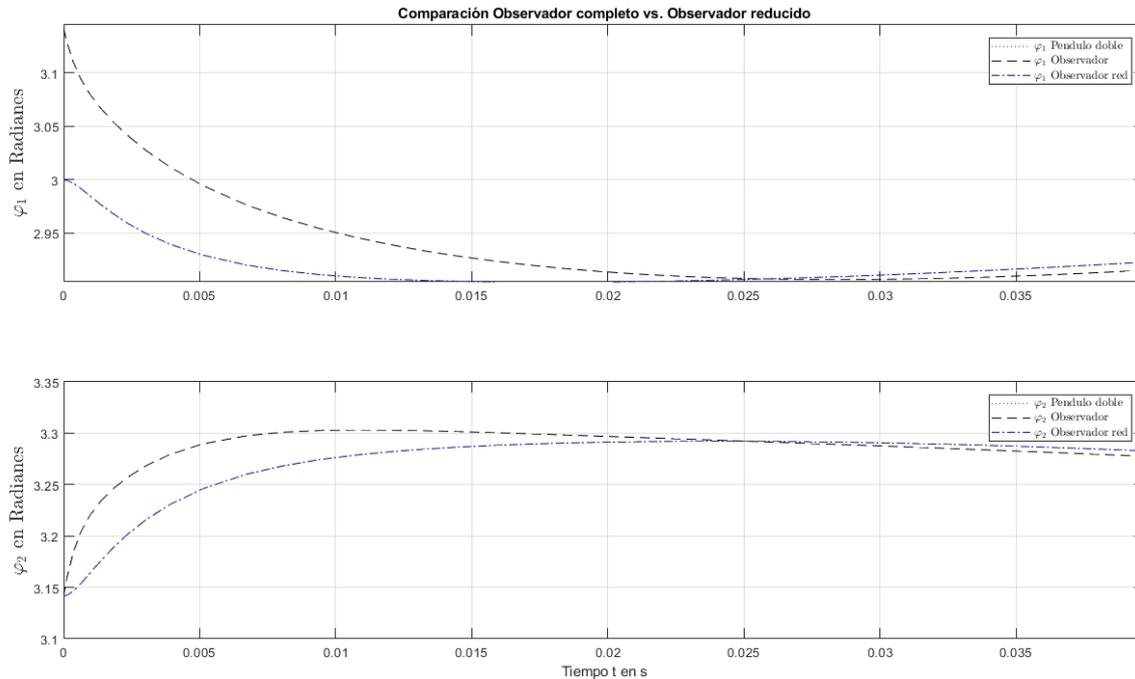


Figura 3.18: Diferencia entre el Observador completo y reducido

Fuente : Elaboración propia

Se ve claramente que el Observador reducido tiene una performance más rápida a la de la respuesta que se espera a la del Sistema, eso quiere decir que el Observador reducido reacciona desde el primer instante a estimar los valores deseados. En cambio, se nota que en cuanto a la respuesta del Observador completo; éste tiene una demora al comienzo pero con el paso del tiempo converge a los valores deseados.

A fin de entender y ver la mejora del Observador reducido también podemos observar una respuesta oscilante casi imperceptible pero marcada para el Observador completo por cierto espacio de tiempo como se puede apreciar en la figura 3.19. Esto se debe a un error numérico y el costo computacional que requiere el algoritmo para obtener los valores. Como se mencionó anteriormente al emplear un Observador reducido se emplea menos recursos y se vuelve más óptimo.

Finalmente, se muestra claramente que los valores de los ángulos convergen en el tiempo pero es casi mínima como se puede ver en la figura 3.20. De todas maneras se muestra una mejora sustancial del observador reducido. Y como

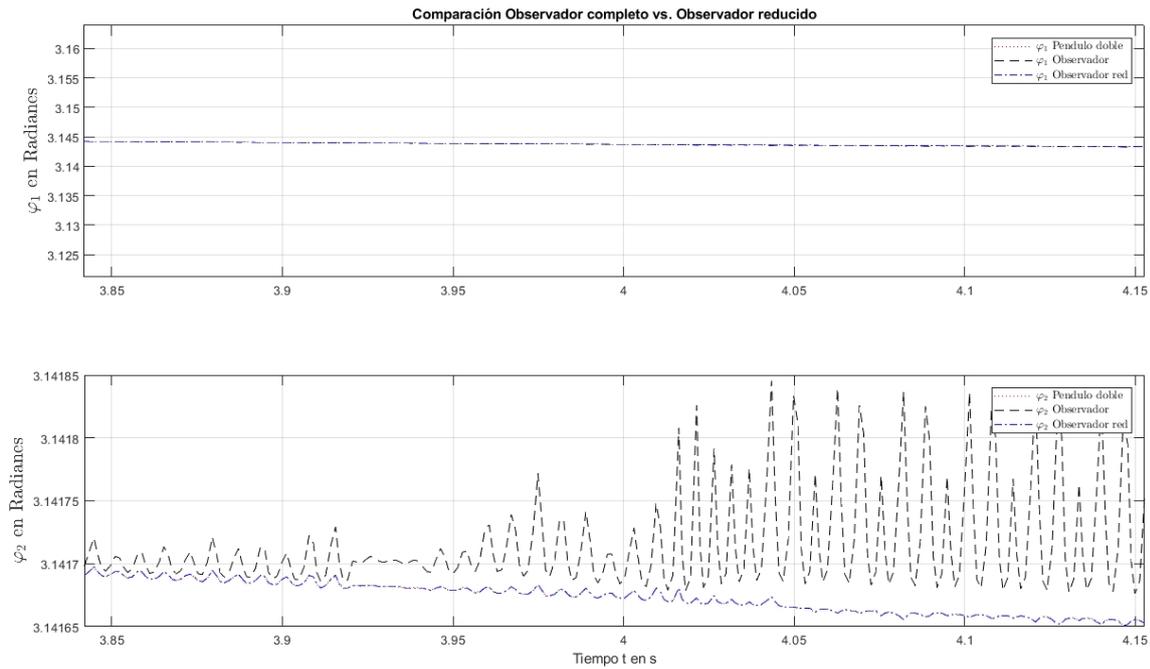


Figura 3.19: Oscilaciones en la respuesta del Observador completo

Fuente : Elaboración propia

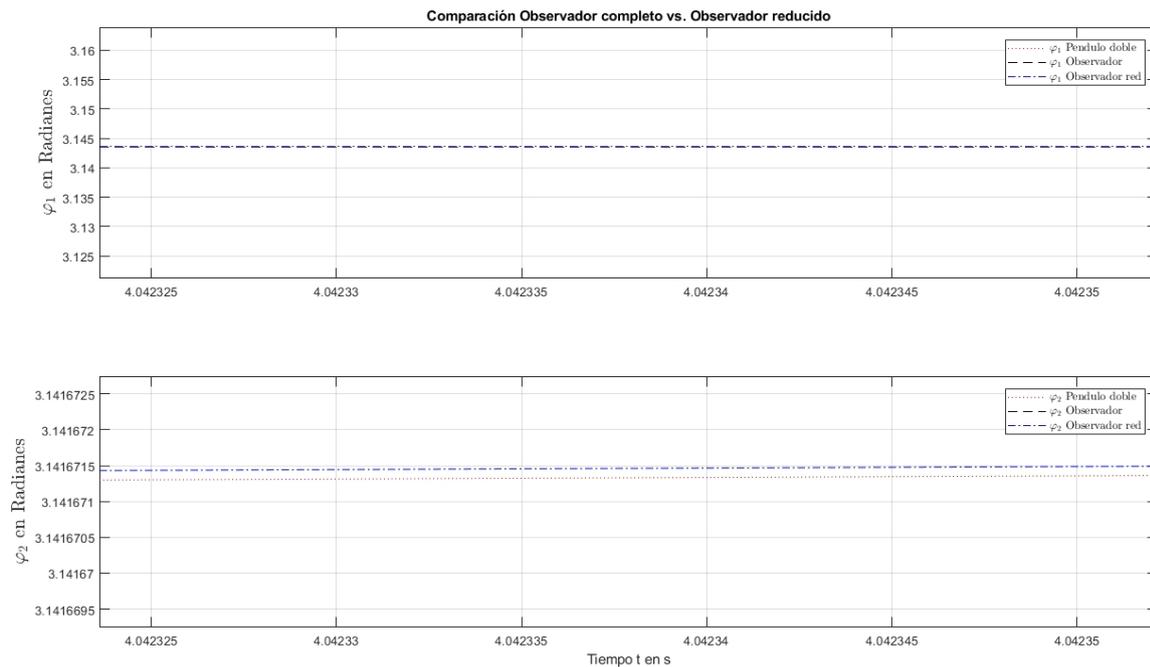


Figura 3.20: Observador completo y reducido convergencia

Fuente : Elaboración propia

se mencionó antes, se debe a que no se requiere observar todas las variables sino parte de ellos a partir de variables de estados medibles.

3.4. Control Lineal Predictivo Basado en Modelo (MPC)

El control predictivo basado en modelos es una técnica de control basada en la optimización que permite alcanzar un rendimiento óptimo de un sistema controlado [34, 35]. Al mismo tiempo, garantiza la satisfacción de restricciones que pueden ser críticas para la seguridad [36, 37]. Como se pudo visualizar en los capítulos anteriores, el controlador LQR junto con los Observadores tanto completo o reducidos, en caso que no es posible medir todas las variables, todos en conjunto aunque presentan buenos resultados, éstos son limitados. Por ejemplo, si se necesita estabilizar nuestro sistema cerca del origen a la vez que satisface restricciones en la entrada; estos controladores no garantizan que las restricciones en la entradas sean satisfechas y menos que se establezca alrededor del origen. O por ejemplo, si añadimos adicionalmente que el controlador tenga restricciones en las variables de estado; el controlado LQR, aunque sea un controlador óptimo, no podría garantizar satisfacer estas restricciones en las variable de estado [38].

Por las razones antes mencionadas, el controlador LQR no cumple con estos requerimientos. Sin embargo, el Control Predictivo Basado en Modelo sí puede garantizar estabilidad en el sistema para todas las variables de estado a la vez que satisfaga restricciones en la entrada. Por lo tanto, en este capítulo se presentará al Control Predictivo Basado en Modelo conocido por sus siglas en inglés MPC (Model Predictive Control). En sus dos variantes es decir lineal y no lineal.

Otros puntos a favor del MPC con respecto al LQR y otros controladores tradicionales es que el MPC puede manejar Sistemas Multivariantes y Sistema no lineales. Este último punto se va a desarrollar a cabalidad en esta tesis y será base para presentar el Control No lineal Predictivo basado en Modelo (NMPC) como la solución idónea para el *Sistema de péndulo doble*.

3.4.1 Principio básico del Control Lineal Predictivo Basado en Modelo

El control predictivo es una técnica de control que utiliza modelos matemáticos para predecir el comportamiento futuro de un sistema en base a sus señales de entrada y salida anteriores. Se trata de un control iterativo que busca optimizar el comportamiento del sistema mediante ajustes continuos de las variables de control en función de las predicciones realizadas. De esta

manera, se pueden tomar decisiones anticipadas y ajustar las variables de control en tiempo real para garantizar un mejor desempeño del sistema en el futuro [39].

Para entender de manera clara el concepto de control óptimo, podemos observar el ejemplo ilustrado en la 3.21. En este caso, un automóvil rojo debe recorrer un circuito sin inconvenientes, buscando llegar de manera eficiente y ahorrando energía en lo posible, lo que se conoce como horizonte de predicción. En el problema de control, las curvas y los bordes del circuito son las restricciones que el automóvil debe considerar para lograr un mejor rendimiento.

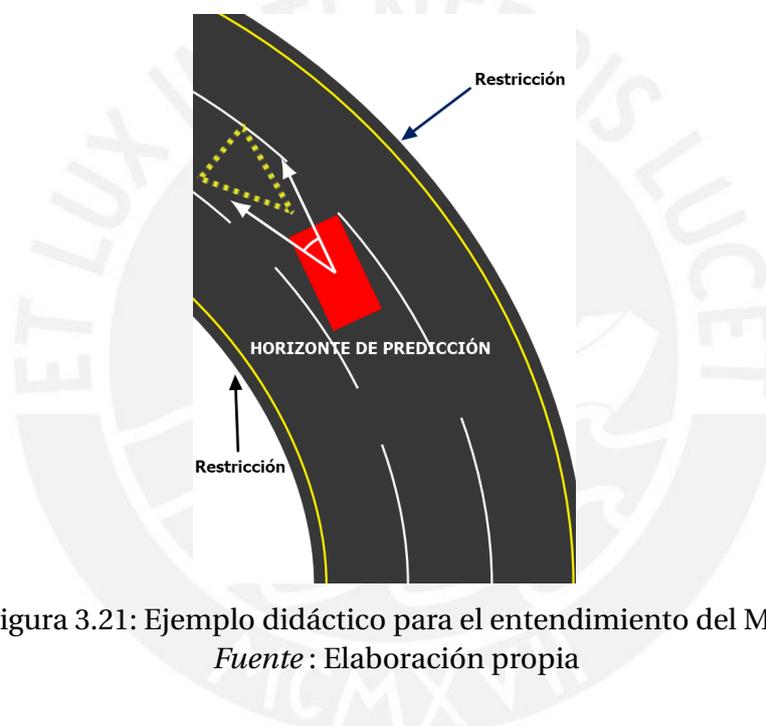


Figura 3.21: Ejemplo didáctico para el entendimiento del MPC
Fuente: Elaboración propia

El problema de control se resuelve de la siguiente manera para el ejemplo antes mencionado:

1. Identificar el modelo representado por el triángulo amarillo punteado en la figura.
2. Predecir el sistema en el futuro utilizando el horizonte de predicción mostrado por las líneas azules en la figura.
3. Resolver el problema óptimo de control para obtener la mejor entrada para ese tiempo, tomando en cuenta las restricciones previamente descritas marcadas mediante las líneas amarillas.

4. Aplicar la entrada óptima anteriormente obtenida en el tiempo actual.
5. Volver a resolver el problema óptimo de control para obtener la mejor entrada para el siguiente tiempo.
6. Regresar al primer punto donde se identifica un nuevo modelo con la información anterior y se vuelve a resolver el problema óptimo de control.
7. Repetir los pasos anteriores hasta completar el circuito, eligiendo siempre la mejor entrada óptima.

En resumen, se utiliza la información del modelo identificado y se predice el sistema en el futuro, se resuelve el problema óptimo de control para obtener la mejor entrada posible, se aplica la entrada óptima actual y se repiten estos pasos hasta completar el circuito, eligiendo siempre la mejor entrada óptima en cada momento.

3.4.2 Formulación matemática del Control Predictivo Óptimo

Si deseamos explorar soluciones al problema de control óptimo desde una perspectiva analítica, es crucial que el sistema de control esté en modo de lazo cerrado para permitir que la retroalimentación influya en el comportamiento óptimo del sistema [40], como se muestra en la Figura 3.22. Para lograrlo, es necesario medir la variable de estado actual en cada intervalo de tiempo t_i tan pronto como se disponga de nueva información. Luego, se resuelve el problema de control óptimo utilizando esta información y se aplica la señal de entrada óptima obtenida $u(t_i)$ al sistema de control. Este proceso se repite continuamente, avanzando en el tiempo t_{i+1} y utilizando la señal de entrada óptima previamente obtenida para resolver nuevamente el problema de control óptimo y aplicarlo al sistema hasta obtener una respuesta óptima.

La secuencia de acciones que se requiere llevar a cabo los pasos para la obtención de una respuesta óptima, se puede comprender de una manera más clara y detallada a través de un diagrama de flujo, que se muestra en la figura 3.23, y en el cual se utiliza el enfoque de control basado en la predicción y optimización repetida, tal y como se ha descrito en la literatura [41].

Ahora se detallará cómo funciona la secuencia para formular el MPC según la secuencia descrita en la 3.23 desde el tiempo t_i , t_{i+1} y t_{i+2} en adelante.

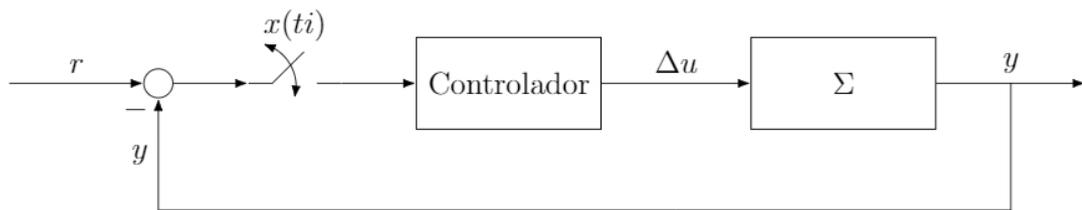


Figura 3.22: Tiempo de muestreo a lazo cerrado
Fuente : Handbook of Model Predictive Control, R. Goebel

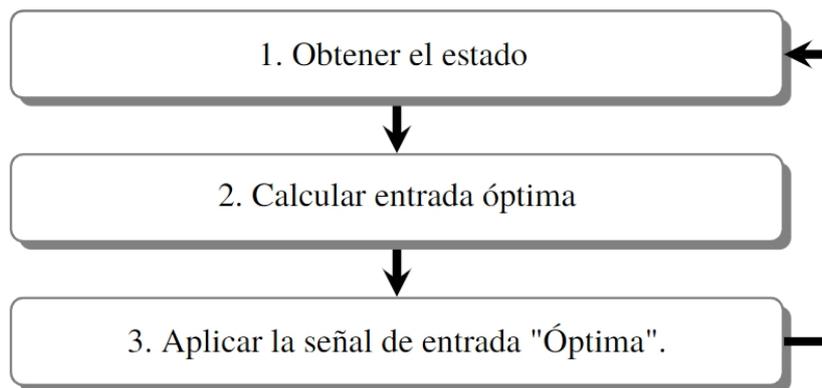


Figura 3.23: Secuencia de formulación del MPC
Fuente : Handbook of Model Predictive Control, R. Sanfelice

Obtener el estado para el tiempo t_i

Se determina el modelo a lo largo de todo el intervalo de tiempo [42], lo que permite abordar el problema de control y obtener la señal de control $u(t)$, como se muestra en la figura 3.24 [43].

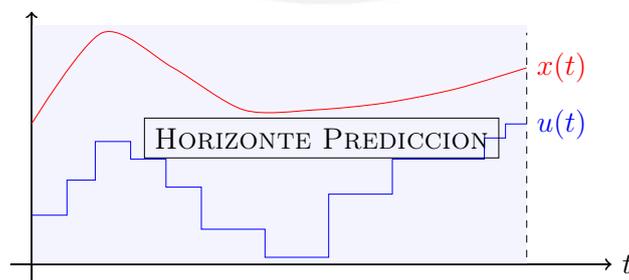


Figura 3.24: Obtención de la señal de control $u(t)$ y variable $x(t)$ en tiempo t .
Fuente : Optimal and Predictive Control , R. Findeisen

Es importante entender que la señal de control solo se aplica durante un breve lapso de tiempo y avanzamos hacia el futuro hasta el instante t_i (Shift Forward),

tal como se muestra en la figura 3.25 [43]. A continuación, se extrae y se obtiene el estado $x(t_i)$ de la trayectoria completa $x(t)$ en el tiempo específico t_i [44], y luego pasamos al siguiente paso.

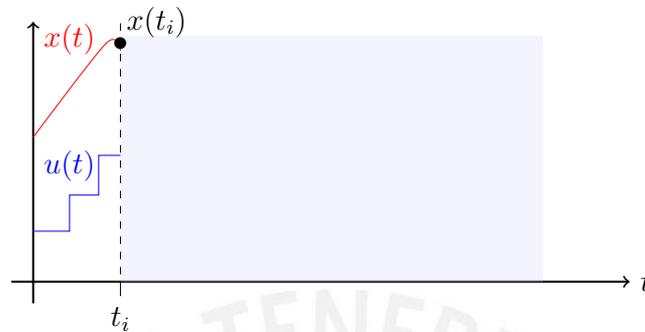


Figura 3.25: Obtención del estado $x(t_i)$ en t_i
Fuente : Optimal and Predictive Control , R. Findeisen

Calcular la entrada óptima en el intervalo de tiempo (t_i a $t_i + T_p$)

Una vez que se ha obtenido el estado $x(t_i)$, se procede a calcular nuevamente la señal de control $u(t)$ desde el instante t_i hasta el final del horizonte de tiempo predicho T_p , tal como se muestra en la figura 3.26.

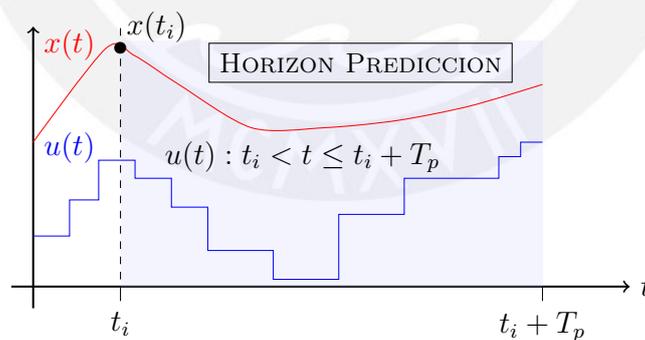


Figura 3.26: Cálculo de la señal de control $u(t)$ para el intervalo $t_i < t \leq t_i + T_p$
Fuente : Optimal and Predictive Control , R. Findeisen

Con el fin de obtener la señal de control óptima $u(t)$, se realiza un cálculo que considera el intervalo de tiempo desde t_i hasta el final del horizonte de tiempo predicho, es decir, $t_i < t \leq t_i + T_p$ [39]. Este cálculo se lleva a cabo mediante la aplicación de una ecuación específica, descrita en la literatura [43].

Calculamos la señal óptima $u(t) : t_i \leq t \leq t_i + T_p$ via

$$\begin{aligned} \min_{u(\cdot)} \int_{t_i}^{t_i+T_p} F(x(\tau), u(\tau)) d\tau + E(x(t_i + T_p)) \\ \text{s.t. } \dot{x}(t) = f(x(t), u(t)), \quad x(t_i) = x_i \\ x(t) \in \mathcal{X}, \quad u(t) \in \mathcal{U} \end{aligned} \quad (3.41)$$

Luego de obtener la nueva señal de control $u(t)$ mediante la ecuación 3.41, avanzamos al siguiente instante t_{i+1} (Shift Forward) y extraemos únicamente el valor de $u(t)$ para el intervalo de tiempo $t_i < t \leq t_{i+1}$, descartando el resto, como se muestra en la figura 3.27 [43].

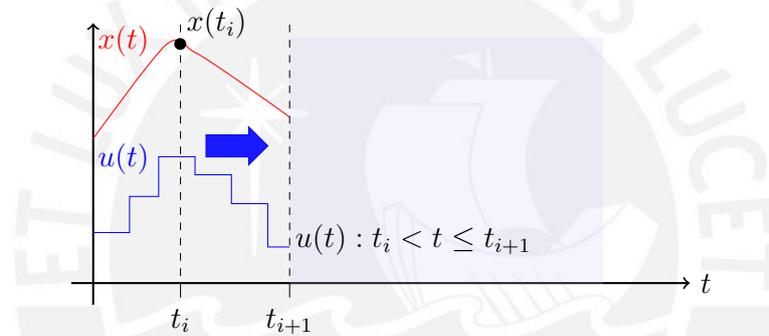


Figura 3.27: Extracción de la señal de control $u(t)$ para el intervalo acortado de $t_i < t \leq t_{i+1}$

Fuente : Optimal and Predictive Control , R. Findeisen

Aplicar la señal de entrada Óptima

Como se pudo observar de la figura anterior, ahora tenemos un nuevo tiempo t_{i+1} y eso hace que la señal de control antes hallada es decir $(u(t) : t_i \leq t \leq t_i + T_p)$ sea acortada entre el intervalo de t_i a t_{i+1} [39]. Y esta nueva señal acortada de control se aplica al sistema:

Al avanzar al siguiente instante de tiempo t_{i+1} , la señal de control anteriormente obtenida $(u(t) : t_i \leq t \leq t_i + T_p)$ se acorta en el intervalo de t_i a t_{i+1} y se aplica al sistema, como se muestra en la figura anterior [39].

$$u(t) : t_i \leq t \leq t_{i+1} \quad (3.42)$$

Por lo tanto, volvemos al punto de partida con la información previamente adquirida. La secuencia comienza desde el tiempo t_{i+1} :

Obtener el estado ahora para el nuevo tiempo t_{i+1}

En el nuevo tiempo t_{i+1} se obtiene el nuevamente el estado $x(t_{i+1})$ para el tiempo específico t_{i+1} , como se puede apreciar en la figura 3.28 [43].

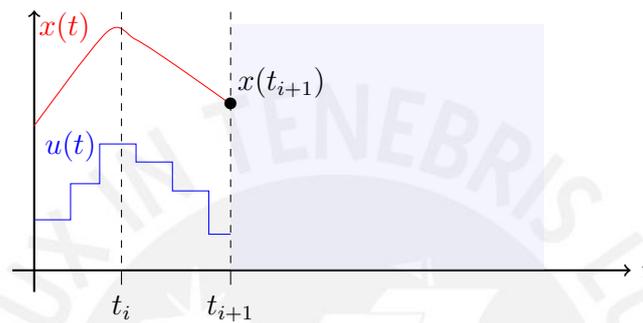


Figura 3.28: Obtención del estado $x(t_{i+1})$ en t_{i+1}
Fuente : Optimal and Predictive Control , R. Findeisen

Calcular la entrada óptima en el intervalo de tiempo $(t_{i+1}$ a $t_{i+1} + T_p$)

Se procede por lo tanto a calcular la entrada o señal de control para este nuevo intervalo como se aprecia en la figura 3.29 [43].

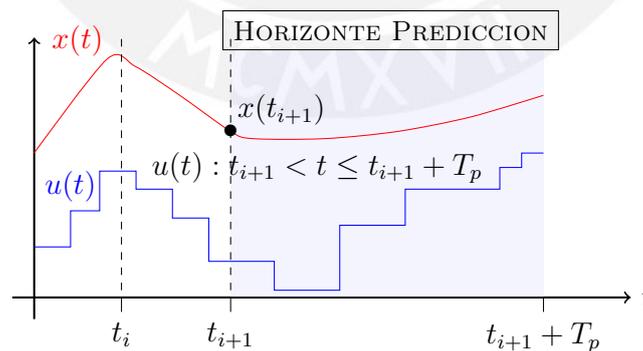


Figura 3.29: Señal de control $u(t)$ para $t_{i+1} < t \leq t_{i+1} + T_p$
Fuente : Optimal and Predictive Control , R. Findeisen

Y la ecuación correspondiente es de la siguiente manera [43]:

Calculamos la señal óptima $u(t) : t_{i+1} \leq t \leq t_{i+1} + T_p$ via

$$\begin{aligned} \min_{u(\cdot)} \int_{t_{i+1}}^{t_{i+1}+T_p} F(x(\tau), u(\tau)) d\tau + E(x(t_{i+1} + T_p)) \\ \text{s.t. } \dot{x}(t) = f(x(t), u(t)), \quad x(t_{i+1}) = x_{i+1} \\ x(t) \in \chi, \quad u(t) \in \mathcal{U} \end{aligned} \quad (3.43)$$

Luego nos movemos nuevamente hacia adelante t_{i+2} (Shift Forward) como se muestra en la figura 3.30 y pasamos al siguiente paso.

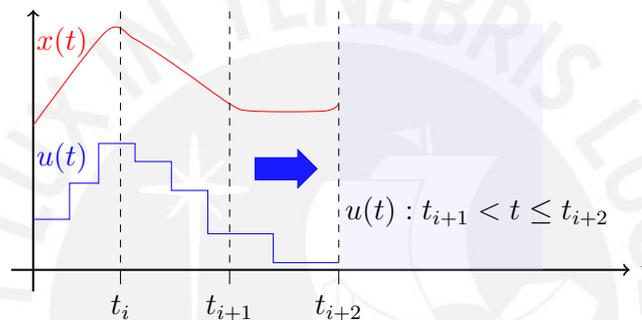


Figura 3.30: Extracción de la señal de control $u(t)$ para el intervalo acortado de $t_{i+1} < t \leq t_{i+2}$

Fuente: Optimal and Predictive Control, R. Findeisen

Aplicar la nueva señal de entrada Óptima

En el presente instante disponemos de un momento t_{i+2} adicional, lo que implica que la señal de control previamente adquirida ($u(t) : t_{i+1} \leq t \leq t_{i+1} + T_p$) se suspende una vez más en el lapso comprendido entre t_{i+1} y t_{i+2} . Se utiliza la señal de control acortada resultante para regular el sistema:

$$u(t) : t_{i+1} \leq t \leq t_{i+2} \quad (3.44)$$

Y así sucesivamente hasta completar el tiempo predicho T_p . Cómo se vio el control predictivo se basa en repetidas predicciones y optimizaciones.

3.4.3 Formulación lineal del MPC

El modelo del sistema dentro de la optimización es o deberá ser una perfecta representación del sistema real. Por lo tanto, las predicciones serían iguales a las mediciones del proceso real. Sin embargo, en la realidad, esto nunca es cierto, existen claramente errores en el modelo "*mismatch*" y/o ruidos en los datos obtenidos entre otras cosas. Teóricamente, se asume por lo tanto un modelo nominal para el MPC [45].

En cuanto a la formulación lineal del MPC, se resuelve el problema de óptimo de control a cada tiempo de muestreo i [39, 46] y representa así:

$$\min_{u(\cdot)} \frac{1}{2} \sum_{k=i}^{i+N-1} (x(k)^T Q x(k) + u(k)^T R u(k)) + \frac{1}{2} x(i+N)^T P x(i+N) \quad (3.45)$$

Sujeto a :

$$x(k+1) = Ax(k) + Bu(k) \quad , \quad x(i) = x_i \quad (3.46)$$

$$x(k) \in \chi, \quad u(k) \in \mu \quad (3.47)$$

$$x(i+N) \in \varepsilon \quad (3.48)$$

Es importante señalar que aunque la función de costo es cuadrático, lo que hace el lineal al MPC lineal es el modelo del sistema como se muestra en la ecuación 3.46 donde χ es la restricción en los estados, μ es la restricción en la entradas, ε es el conjunto terminal al final de la predicción y N es el horizonte de predicción. Ahora bien, podemos usar una simplificación a la notación porque el sistema, las restricciones y las funciones de costos son invariantes en el tiempo [42]. Por lo tanto tenemos la siguiente formulación del MPC simplificada:

$$\min_{u(\cdot)} \frac{1}{2} \sum_{k=0}^{N-1} (x(k)^T Q x(k) + u(k)^T R u(k)) + \frac{1}{2} x(N)^T P x(N) \quad (3.49)$$

Sujeto a :

$$x(k+1) = Ax(k) + Bu(k) \quad , \quad x(0) = x_i \quad (3.50)$$

$$x(k) \in \chi, \quad u(k) \in \mu \quad (3.51)$$

$$x(N) \in \varepsilon \quad (3.52)$$

Con χ, μ, ε como conjuntos convexos

Viabilidad recursiva

Como se puede observar en la ecuación 3.49, el objetivo de la función de costo es encontrar la señal óptima de control u . Sin embargo, se debe satisfacer las restricciones para las variables χ y entradas μ para todos los valores de $K = 0, \dots, N - 1$. Además al final de la predicción el conjunto terminal ε , las variables deberán permanecer alrededor de este conjunto de modo que se satisfaga también esta condición.

Por lo tanto, una vez que obtenemos la secuencia de entradas U_N y que satisfagan todas las restricciones, ésta deberá satisfacerlas iterativamente para todo los tiempos desde $i = 0$. A esto se le denomina viabilidad recursiva [40].

3.4.4 Diseño Control Lineal Predictivo Basado en Modelo (MPC)

En cuanto al MPC lineal como se dijo antes, la idea es predecir el comportamiento del sistema calculando en cada iteración del algoritmo, la entrada necesaria que resuelve el problema de control óptimo. Asimismo, el objetivo de este controlador es que las variables de estado convergen al valor de referencia esperado, modificando la entrada hasta lograr que la distancia entre las variables de estado del sistema con respecto a la referencia esperada sean mínimas o casi nula, considerando naturalmente las restricciones del caso.

La implementación de el controlador MPC esta hecho en forma discreta. Sin embargo, se debe tener en cuenta que nuestro modelo es lineal y continuo. Por lo tanto, se deberá linealizar el modelo Espacio-Estado continuo a discreto en el punto de operación $X_{PO} = [\pi \ 0 \ \pi \ 0]$. Consideramos además un sistemas lineal de tiempo discreto e invariable de la siguiente forma:

$$x_{k+1} = Ax_k + Bu_k \quad (3.53)$$

Donde x_k y u_k son las variables de estado y de entrada respectivamente en un tiempo dado, y x_{k+1} es la variable estado siguiente. Cabe resaltar y dejar en claro que este algoritmo para la implementación del controlador MPC es llamado *simultáneo* ya que se obtienen en cada tiempo de muestreo, la entrada de control y la variable de estado cuyos valores serán parte de las decisiones a tomar en cuenta para los siguientes cálculos del controlador. Matemáticamente, el objetivo es llevar a las variables de estado del sistema

desde un valor inicial (x_0) al valor de la referencia (w) [39] y se describe de la siguiente manera:

$$\min_z \frac{1}{2} \sum_{k=0}^{N_p-1} \underbrace{((x_k - w)^T Q (x_k - w) + u_k^T R u_k)}_{\text{Coste de fase}} + \frac{1}{2} \underbrace{(x_N - w)^T P (x_N - w)}_{\text{Coste Terminal}} \quad (3.54)$$

Sujeto a :

$$x_k = x_0 \quad (3.55)$$

$$x_{k+1} = Ax_k + Bu_k, \quad k = 0, \dots, N_p - 1 \quad (3.56)$$

$$u_{\min} \leq u_k \leq u_{\max}, \quad k = 0, \dots, N_p - 1 \quad (3.57)$$

$$x_{\min} \leq x_k \leq x_{\max}, \quad k = 0, \dots, N_p \quad (3.58)$$

Donde la ecuación 3.54 es la función objetivo que conduce a las variables de estado del sistema al punto de referencia deseado bajo las restricciones dinámicas del sistema lineal representadas por la ecuación 3.56. Las restricciones para las variables de estado y la entrada de control están representadas en las ecuaciones 3.57 y 3.58 respectivamente. La ecuación 3.55 fuerza al algoritmo de predicción empezar con las variables de estado iniciales. En cuanto al N_p representa al horizonte de predicción, en otras palabras significa hasta qué punto predecimos el futuro. Finalmente, las matrices Q , R y P son las matrices de algebraicas de Ricatti. En cuanto a la matriz Q (Matriz Coste de estado) deberá ser semidefinida positiva y simétrica cuya función es ajustar los valores de las variables de estado, la matriz R (Matriz Coste de entrada) deberá ser definida positiva y simétrica y su función es ajustar los valores de las entradas o entradas, y finalmente la matriz P (Matriz Coste final) deberá ser semidefinida positiva y simétrica cuya función es ajustar los valores en el tiempo final. La variable z que aparece en la ecuación 3.54 representa a las entradas y las variables de estados donde $z = [x_0, \dots, x_{N_p-1}, u_0, \dots, u_{N_p-1}]^T$

Se explicará a continuación el algoritmo y los resultados del mismo para este punto de operación por ser lineal.

Conversión del Sistema continuo a discreto

Como se mencionó anteriormente la implementación del MPC es realizada en forma discreta para ello, nuestro sistema discreto es obtenido a través del

método Zero-Order Hold (ZOH), método que se adecua al tipo de entrada que usamos para usamos para el péndulo doble es decir escalón. Por ejemplo de una entrada para el estado φ_1 (Por ejemplo, de $\frac{\pi}{2}$ a π) y de la misma forma para el otro estado φ_2 [47]. En nuestro caso, usaremos los valores iniciales y puntos de operación que fueron usamos para el control LQR mostrados en la ecuación 3.19 del capítulo 3.2 de modo que se pueda mostrar una mejor comparación de los diferentes controladores. A continuación se muestra el Sistema Continuo en el algoritmo 3.9 para el punto de operación $\mathbf{PO2} = [\pi \ 0 \ \pi \ 0]$. dado por el algoritmo 2.3 del capítulo 2 como parte de estudio para este y demás controladores.

Algoritmo 3.9: Definición del Sistema Continuo

```

1 %Definicion de Sistema continuo
2 Asys_c= [0 1 0 0;-126.128571428571 0 63.0642857142857 ...
           0;0 0 0 1;189.192857142857 0 -168.171428571429 0];
3 Bsys_c= [0;142.857142857143;0;-214.285714285714];
4 Csys_c= [1 0 0 0;0 0 1 0];
5 Dsys_c= [0;0];

```

Las matrices continuas que representan al péndulo doble para el punto de operación antes mencionados se deberá discretizar para un tiempo de muestreo $T_s = 0.01$. El valor que se toma para el tiempo de muestreo no es arbitrario debido a que tiene que cumplir requisitos de estabilidad, capacidad de computo y requisito de reconstrucción requerida por el teorema de Nyquist y Shannon [48]. En nuestro caso, es necesario elegir un tiempo de muestreo que pueda mantener el sistema estable. Para eso se toma en cuenta en su forma discreta que los polos y ceros de la correspondiente función de transferencia discreta se encuentren dentro del círculo unitario con el tiempo de muestreo.

Ahora bien, el comando en MATLAB convierte el Sistema-Espacio de estado continuo en un Sistema-Espacio de estado discreto usando el metodo ZOH ya antes mencionado. A continuación se presenta el comando donde las matrices A_{sys_c} , B_{sys_c} , C_{sys_c} y D_{sys_c} son los que se hallaron luego de la linealización es el siguiente:

Algoritmo 3.10: Discretización Sistema Péndulo

```

1 %% Metodo discretizacion ZOH para un Ts=0.01

```

```
2 [F,G,H,J]=c2dm (Asys_c,Bsys_c,Csys_c,Dsys_c,Ts,'zoh');
```

Se obtiene a partir de este comando las matrices Espacio-Estado discretas con las que el algoritmo trabajará guardadas en las matrices F, G, H, J que representan como es natural las correspondientes matrices discretas que describen el *Sistema de péndulo doble* para los puntos de operación $\mathbf{X}_{PO} = [\pi \ 0 \ \pi \ 0]$.

Parámetros del controlador

Para el controlador MPC consideramos un horizonte de predicción de $N_p = 30$, valor aceptable que resulta después de varias pruebas realizadas. Sobre el valor para el N_p depende de la estabilidad del sistema controlado que en forma ideal el valor deberá tender al ∞ que garantiza estabilidad [42]. Sin embargo, este valor no es realizable pero existe un teorema que se denomina "*Quasi to Infinite*" y éste aproxima el sistema a un valor cercano al infinito o dentro de la región de estabilidad [49]; de ese tema se hablará mas adelante en los siguientes capítulos de esta tesis.

Ahora bien, se define los parámetros del controlador MPC; en cuanto al valor inicial del controlador sería $\mathbf{x}_0 = [3 \ 0 \ \pi \ 0]^T$ y las restricciones para las variables de estado y las entradas son las siguientes:

$$\begin{aligned} x_{\min} &= [-\pi, -\pi, -\pi, -\pi]^T, & x_{\max} &= [\pi, \pi, \pi, \pi]^T \\ u_{\min} &= -10, & u_{\max} &= 10 \end{aligned} \quad (3.59)$$

Y los parámetros para la función de costo son los siguientes en primera instancia que luego dependiendo de los resultados se pueden variar:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R = 1 \quad (3.60)$$

Implementación del MPC linear en MATLAB

La idea para resolver problemas de control óptimo dinámico con restricciones es reformular un problema de optimización dinámica en un programación

cuadrática (QP) [50, 51]. Es decir, la ecuación 3.54 se reformula a la ecuación 3.61 que corresponde a una QP. Para eso, para la implementación de esta nueva ecuación, se usará el *solver cuadrático* cuyo comando es `quadprog` en MATLAB [52]. Este *solver* resolverá el siguiente problema de programación cuadrática discreta de la forma:

$$\min_z \frac{1}{2} z^T H z + f^T z \quad (3.61)$$

Sujeto a :

$$A z \leq b \quad (3.62)$$

$$A_{eq} z = b_{eq} \quad (3.63)$$

$$lb \leq z \leq ub \quad (3.64)$$

El problema de control óptimo que resulta de la formulación MPC es también un tipo de programación cuadrática como se puede ver al mirar la ecuación 3.54, pero no está en una forma estándar como en la de ecuación 3.61. De modo que para utilizar el *solver cuadrático* `quadprog` tenemos que construir todas las matrices y vectores, y así encontrar la solución de nuestro problema utilizando el comando en MATLAB `quadprog` [53]. El comando sería de la siguiente forma:

Algoritmo 3.11: Comando `quadprog`

```
1 %% Solucion del problema cuadratico en cada tiempo de ...
   muestreo
2 Z = quadprog(H, f, A, b, Aeq, beq, lb, ub, z0);
```

Construcción las matrices para la predicción del sistema

A partir de la ecuación 3.56 se puede apreciar que la dinámica del sistema lineal, forma parte de las restricciones lineales en la formulación del problema óptimo. A fin de incluir esta información en el algoritmo se usarán las matrices

A_{eq} y b_{eq} . Donde pueden ser escritas de la siguiente forma:

$$\begin{aligned}
 Ax_0 - x_1 + Bu_0 &= 0 \\
 Ax_1 - x_2 + Bu_1 &= 0 \\
 Ax_2 - x_3 + Bu_2 &= 0 \\
 &\vdots \\
 Ax_{N_p-1} - x_{N_p} + Bu_{N_p-1} &= 0
 \end{aligned} \tag{3.65}$$

El comando quadprog necesita que las ecuaciones antes mostradas se escriban en forma estándar de la forma $A_{eq}z = b_{eq}$ y puede ser escrito de la siguiente forma:

$$\begin{bmatrix}
 A & -I & 0 & 0 & \cdots & B & 0 & 0 & \cdots & 0 \\
 0 & A & -I & 0 & \cdots & 0 & B & 0 & \cdots & 0 \\
 0 & 0 & A & -I & \cdots & 0 & 0 & B & \cdots & 0 \\
 & & & \vdots & & & & & & \\
 & & & \vdots & & & & & & \\
 & & & \vdots & & & & & & \\
 0 & 0 & 0 & A & -I & 0 & 0 & 0 & \cdots & B
 \end{bmatrix}
 \begin{bmatrix}
 x_0 \\
 x_1 \\
 \vdots \\
 x_{N_p} \\
 u_0 \\
 \vdots \\
 u_{N_p-1}
 \end{bmatrix}
 =
 \begin{bmatrix}
 0 \\
 0 \\
 0 \\
 0 \\
 \vdots \\
 0 \\
 0 \\
 0
 \end{bmatrix}$$

$\underbrace{\hspace{15em}}_{n_x + n_x \cdot N_p} \quad \underbrace{\hspace{15em}}_{n_u \cdot N_p} \quad \underbrace{\hspace{5em}}_z$

Donde n_x es el número de variables de estado (dimensión del vector x), n_u es el número de entrada de control (dimensión del vector u) y z es el vector extendido de variables extendidas donde contiene las entradas y variables $z = (x_1^T \ \cdots \ x_{N_p}^T \ u_0^T \ \cdots \ u_{N_p-1}^T)^T$.

Algoritmo 3.12: Construcción matrices A_{eq} y B_{eq} con comando kron

```

1  % *****
2  % Construcción de las matrices de predicción
3  % *****
4
5  % Construcción Aeq %Asys=discreto de A%
6  aux1 = eye(Np,Np+1);
7  Abar1 = kron(aux1, Asys);
8  % Parte de matriz identidad
9  for i = 1 : Np
10     aux2(i,i+1) = -1;

```

```

11 end
12 Abar2 = kron(aux2, eye(4));
13 % Part of the control matrix %Bsys=discreto de B%
14 aux3 = eye(Np, Np);
15 Abar3 = kron(aux3, Bsys);
16
17 % Construccion completa matriz Aeq
18 Aeq = [Abar1+Abar2, Abar3];
19 % Construccion Beq
20 beq = zeros(nx*Np, 1);

```

En el entorno de MATLAB a fin de crear las matrices utilizamos el comando `kron` que junta dos matrices en una sola. En nuestro caso, empezamos creando con las matrices A del sistema discreto, luego añadimos las matrices de identidad y creamos por separado la parte con las matrices B proveniente del sistema discreto. Finalmente ponlas todas juntas en una única matriz A_{eq} como se puede apreciar en el código de línea 18 en el algoritmo. Por último, creamos la matriz B_{eq} con ceros.

Construcción de las matrices para la función de costo

De la ecuación 3.54 se puede apreciar que la función de costo del controlador MPC es la suma de costos cuadráticos sobre el horizonte de predicción. Esto puede ser transformado en matrices requeridas para el comando `quadprog` requerido en la forma estándar como se puede apreciar en la ecuación 3.61 y se vería de la siguiente forma:

$$\frac{1}{2}z^T \underbrace{\begin{bmatrix} Q & & & & \\ & Q & & & \\ & & \ddots & & \\ & & & Q & \\ & & & & R \\ & & & & & \ddots \\ & & & & & & R \end{bmatrix}}_H z + \underbrace{\begin{bmatrix} 0 & 0 & \dots & 0 & 0 \end{bmatrix}}_{f^T} z$$

En este caso volvemos a hacer uso del comando (`kron`). Además, creamos por separado la parte de la matriz H que contiene las matrices Q en la diagonal

y la parte que contiene las matrices R. Después puedes utilizar el comando (blkdiag) de MATLAB que creará una matriz con los bloques dados en la diagonal.

Construcción de las matrices para la predicción inicial, y los límites inferiores y superiores de la variables de optimización

El valor de los valores iniciales de las variables de optimización $x_0 = x_{initial}$ mostrado en la ecuación 3.55 se debe ingresar como argumento en el comando, para ello se debe forzar al optimizador iniciar con la predicción inicial y así sucesivamente actualizar este valor en cada tiempo de muestreo previamente computado. Este vector se halla de igual manera con el comando kron. En cuanto a los límites superiores e inferiores de las ecuaciones 3.57 y 3.58 respectivamente se usan este método de concatenación. A continuación se muestra el algoritmo para la construcción de los vectores:

Algoritmo 3.13: Construcción vectores x_0 - lb - ub con comando kron

```

1 % *****
2 % Construcción valor inicial, los límites inferiores y ...
   superiores
3 % Parte de variables de estado
4 z0_states = kron(ones(Np+1,1), x_guess);
5 lb_states = kron(ones(Np+1,1), x_min);
6 ub_states = kron(ones(Np+1,1), x_max);
7 % Parte de Control
8 z0_control = kron(ones(Np,1), u_guess);
9 lb_control = kron(ones(Np,1), u_min);
10 ub_control = kron(ones(Np,1), u_max);
11 % Construcción final de los vectores
12 z0 = [z0_states; z0_control];
13 lb = [lb_states; lb_control];
14 ub = [ub_states; ub_control];

```

Solución del MPC en cada tiempo de muestreo

Como se mencionó antes, se usará el comando quadprog y los argumentos (matrices y vectores) ya fueron hallados en los pasos previos a este apartado.

Asimismo, al no tener restricciones de desigualdad estas matrices A y b se definirán nulas.

En el algoritmo 3.11 la salida del comando será el vector "Z" en mayúscula y no confundir con el vector extendido de entradas y variables "z", el cual contiene el control U_{mpc} y la trayectoria de las variables. Otro punto importante a recordar es que con el algoritmo obtenemos la entrada óptima de control en cada tiempo de muestro y este lo usaremos directamente en la dinámica del sistema $x_{next} = Ax_{initial} + BU_{mpc}$. Pero no usaremos todo el vector de salida Z, sino que extraemos solo la primera señal de control procedente del algoritmo 3.11 de la siguiente manera:

Algoritmo 3.14: Extraccion de la entrada de control

```
1   % Extraccion de la primera entrada de control de la ...
    solucion optima
2   u_mpc = Z(dim_states+1,1);
```

Esto responde al cálculo de la señal de control repetitiva explicado desde el capítulo 3.5 en adelante sobre el termino denominado como Shift Forward hasta el final de la simulación (t_{end}) de tal manera que en cada paso de tiempo se resuelve el problema de optimización (N_p) pasos de predicción en el futuro considerando sólo una señal de control óptima en cada paso. Por lo tanto, calculamos el sistema real usando la senal de control calculado en cada tiempo de muestreo de lasiguiente manera:

Algoritmo 3.15: Simulacion del sistema real

```
1   % Simulacion del sistema real
2   x_next = Asys*x_initial + Bsys*u_mpc;
```

Finalmente se ejecuta el comando quadprog y se analiza los resultados a continuación.

Los resultados del MPC lineal

Una vez que se que se ejecuta el comando, se obtiene los resultados para los ángulos φ_1 y φ_2 , las velocidades $\dot{\varphi}_1$ y $\dot{\varphi}_2$, y la entrada de control u como se muestra en la figura 3.31.

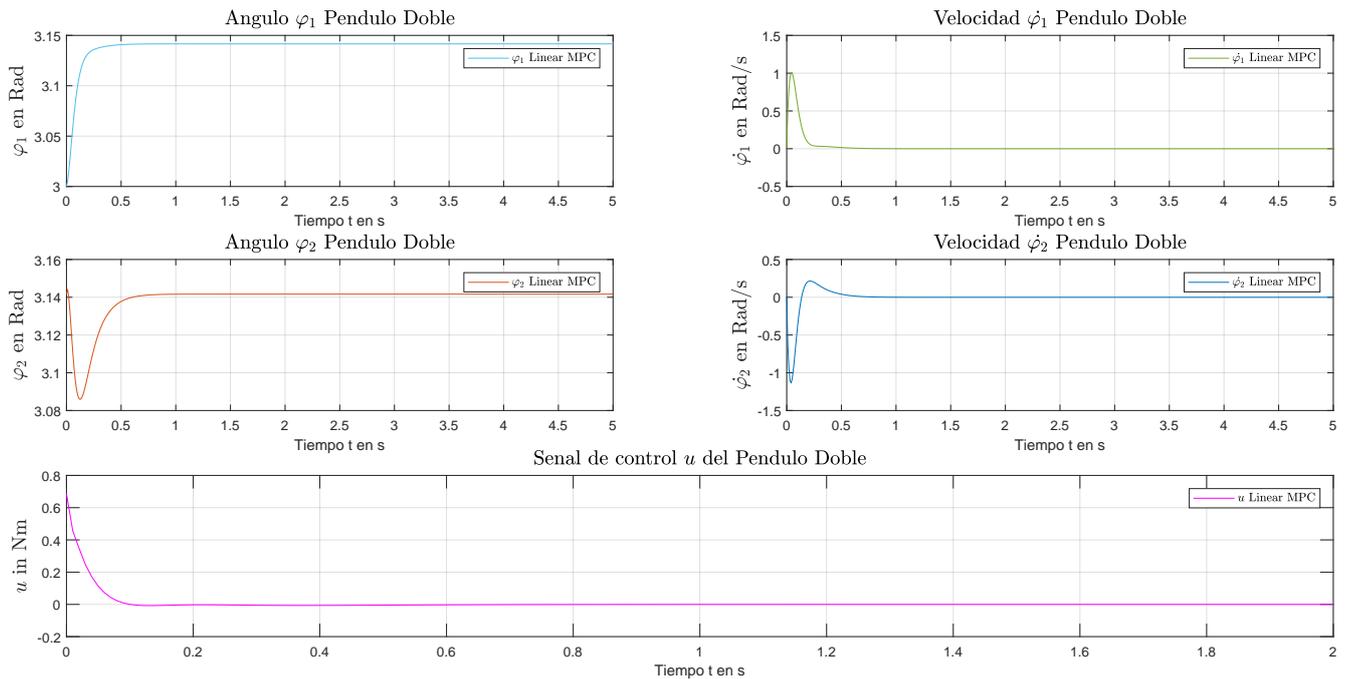


Figura 3.31: Resultado de simulación del MPC lineal usando el comando

QUADPROG

Fuente: Elaboración propia

Como se puede apreciar en las figura 3.31, para el caso del primer ángulo φ_1 empieza como punto inicial 3 radianes y como punto final π radianes. Dentro de nuestro intervalo de simulación, el ángulo φ_1 llega a su punto de equilibrio o esperado en un tiempo de 1.57 segundos aproximadamente. En cuanto al ángulo φ_2 perteneciente al segundo péndulo empieza como punto inicial en π radianes y llega a su punto de equilibrio en 1.86 segundos aproximadamente. Lo mismo se puede apreciar la señal de entrada de control u , en la que después de 1.6 segundos llega a su punto de equilibrio.

Para entender lo que representa los resultados anteriores, debemos recordar que el problema óptimo de control (OCP) se solucionó por medio de una programación cuadrática (QP). En donde la función de costo se tuvo que reformular para que pueda ser usado por el *solver* quadprog. La primera explicación de los resultados se debe a que el modelo dinámico del sistema se tuvo que llevar al plano discreto. Es decir, se usó una aproximación para llevarlo a cabo, en ese ínterin es posible que se halla perdido información de la dinámica ya se trabaja en un punto específico de operación como se señalo en la ecuación 3.19 del capítulo 3.2. Debe quedar claro que este punto

de operación no representa el modelo completo de la dinámica del sistema sino una parte. Además, otra explicación de los resultados va por el hecho de la existencia de retardos propios que se deriva del sistema real y que afectan al tiempo de establecimiento final del sistema. Cabe bien señalar que este aspecto no se tomó en cuenta al momento de realizar la solución del sistema del péndulo doble por medio del algoritmo de MATLAB utilizado. Si bien tenemos buenos resultados para el uso del MPC lineal por medio del *solver* QUADPROG, como se puede apreciar en los resultados. Sin embargo, existe una diferencia importante con los resultados antes propuestos como el del controlador de estado y sus variantes, a decir, observador reducido y completo. Si observamos detalladamente en la figura 3.31 existe una oscilación evidente al inicio de la simulación. Por eso, es muy recomendable usar un sistema no lineal donde la casi la totalidad de la dinámica del sistema es tomado en cuenta, esto sera visto en el capítulo siguiente.

Otro punto interesante a notar se podría mejorar el algoritmo usando un controlador MPC para el seguimiento de referencia y no un comando de MATLAB, ya que si cambia los puntos de referencia, cambia totalmente el controlador que debe asegurar la viabilidad de la optimización [54]. Esa técnica ya de por si seria un estudio exhaustivo que escapa del espectro de esta tesis pero se deja al lector como un posible tema de investigación a futuro.

En cuanto a la entrada de señal de control, era de esperar que su estabilización tenga el valor en equilibrio de 0 Nm como se mostró también en la tabla 2.5. Esto se explica que cuando los dos péndulos alcanzan el valor de π , es decir se encuentren de forma perpendicular en referencia al plano en el que el sistema gira, se mantengan en esa posición hasta tiempo indefinido por el hecho que no existe algún tipo perturbación externa que haga que cambien como por ejemplo la fricción, la gravedad en sí misma, el desgaste de los engranajes o el viento para mencionar algunas de ellas. En cuanto a la fricción en particular no se tomó en cuenta para el modelado del *Sistema de péndulo doble* en su variante lineal. En cuanto al valor de la entrada óptima de control calculada, en forma numérica, se puede apreciar, detalladamente en la parte inferior central de la figura 3.31, que el valor en Nm cambia de 0.23 Nm a 0 Nm en 1.86 segundos aproximadamente y se mantiene en 0 unidades hasta tiempo indefinido debido a que el péndulo doble sólo necesita de cierto Momento o Fuerza para llegar a su posición perpendicular y luego de eso ya no necesita de alguna fuerza adicional para mantener los péndulos en la posición perpendicular por

eso el valor del momento queda en 0 Nm considerando adicionalmente que no hay perturbación que lo haga cambiar.

Por otro lado el número de pasos del horizonte de predicción es de $N=30$ y es en cierta forma suficiente a fin de encontrar el mínimo de la función de coste. Además, junto con la buena elección del horizonte predicción y como bien se vió en la parte teórica, el uso del termino final en la función de costo puede mejorar la respuesta ya que concentra a la misma dentro de un conjunto que garantice su estabilidad y performance cercana al su punto de equilibrio. Sin embargo, no se usó este termino final ya que el *solver* quadprog no lo desarrolla dentro de su algoritmo.

Por último, valores de las matrices R y Q harían un cambio sustancial pero debido a los errores antes mencionados no tendrían mucho efecto. Ya que incrementando la matriz Q vamos a forzar al sistema a alcanzar la referencia lo mas rápido posible pero a costa del incremento del control para hacerlo. Y al incrementar la matriz R significa que se fuerza al sistema a llegar al reposo rápidamente y esto puede llevar a que los tardará más tiempo en alcanzar el estado estacionario.

Esta claro que un MPC lineal puede ser reformulado en una programación estática cuadrática (QP), no necesita aproximaciones iterativas ya que el modelo ya es lineal y el problema es convexo. Por lo tanto, se puede encontrar una solución óptima global. Estas bondades del MPC lineal es, sin embargo, limitado ya que en la práctica y los sistemas a controlar no son lineales o se pueden aproximar de manera que cubra todas el comportamiento propio del sistema. Lamentablemente un MPC lineal no cubre todas las necesidades de la industria, por eso, se estudiará en el siguiente capítulo el MPC no lineal que es ampliamente efectivo y sí cumple las necesidades antes mencionadas.

Prueba MPC lineal bajo otros puntos de operación

A continuación se prueba el MPC lineal en diferente puntos de operación. Debemos recordar que el punto de operación es aquel punto donde se linealiza el sistema y ademas sirve como punto de referencia donde deberán llegar ambos péndulos. Ahora bien se van a variar las condiciones iniciales alejandonos desde el punto de operación a fin de probar el controlador. El primer punto es el siguiente:

$$\mathbf{X}_0 = [3 \ 0 \ 3 \ 0]$$

$$\mathbf{X}_{PO} = [\pi \ 0 \ \pi \ 0]$$

$$\begin{aligned} x_{\min} &= [-\pi, -\pi, -\pi, -\pi]^T, & x_{\max} &= [\pi, \pi, \pi, \pi]^T \\ u_{\min} &= -10, & u_{\max} &= 10 \end{aligned} \quad (3.66)$$

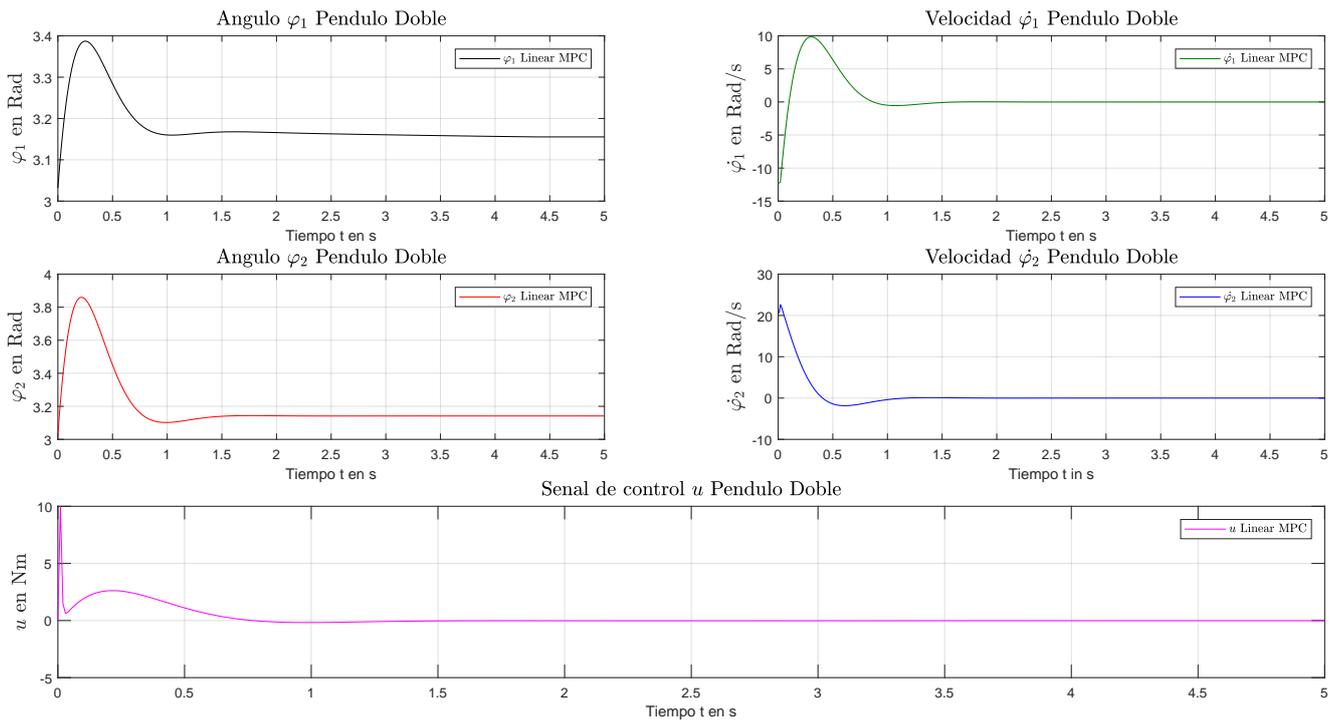


Figura 3.32: Resultado de simulación del MPC lineal (MPC) con $x_0 = [3 \ 0 \ 3 \ 0]$
y $x_{PO} = [\pi \ 0 \ \pi \ 0]$

Fuente: Elaboración propia

De los resultados de la figura 3.32 se puede apreciar que el primer ángulo φ_1 no converge a π pero si llega a un valor cercano. En cuanto al segundo ángulo φ_2 converge al punto esperado luego de los 1.7 segundos aproximadamente. Veamos un segundo punto mas alejado al punto de operación:

$$\mathbf{X}_0 = [\frac{\pi}{2} \ 0 \ \frac{\pi}{2} \ 0]$$

$$\mathbf{X}_{PO} = [\pi \ 0 \ \pi \ 0]$$

$$\begin{aligned} x_{\min} &= [-\pi, -\pi, -\pi, -\pi]^T, & x_{\max} &= [\pi, \pi, \pi, \pi]^T \\ u_{\min} &= -10, & u_{\max} &= 10 \end{aligned} \quad (3.67)$$

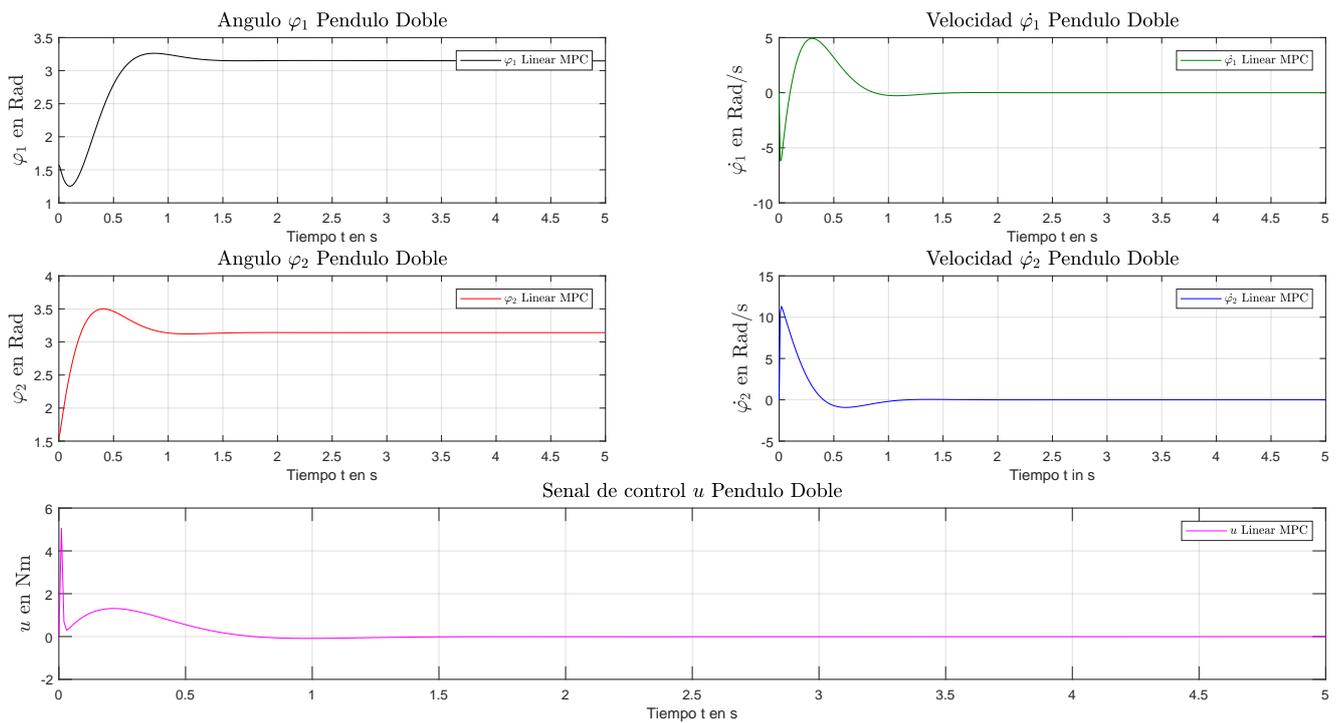


Figura 3.33: Resultado de simulación del MPC lineal (MPC) con

$$x_O = \left[\frac{\pi}{2} \ 0 \ \frac{\pi}{2} \ 0 \right] \text{ y } x_{PO} = \left[\pi \ 0 \ \pi \ 0 \right]$$

Fuente : Elaboración propia

Del mismo modo los resultados mostrados en la figura 3.33 hacemos el rango de acción más amplio. De la figura podemos notar claramente que no convergen a los puntos de equilibrio queridos. Para ser exactos, la no convergencia para ambos péndulos φ_1 y φ_2 por pocos puntos. Veamos un tercer punto más amplio y veamos que sucede:

$$\mathbf{X}_0 = [0 \ 0 \ 0 \ 0]$$

$$\mathbf{X}_{PO} = [\pi \ 0 \ \pi \ 0]$$

$$x_{\min} = [-\pi, -\pi, -\pi, -\pi]^T \quad , \quad x_{\max} = [\pi, \pi, \pi, \pi]^T$$

$$u_{\min} = -10 \quad , \quad u_{\max} = 10 \quad (3.68)$$

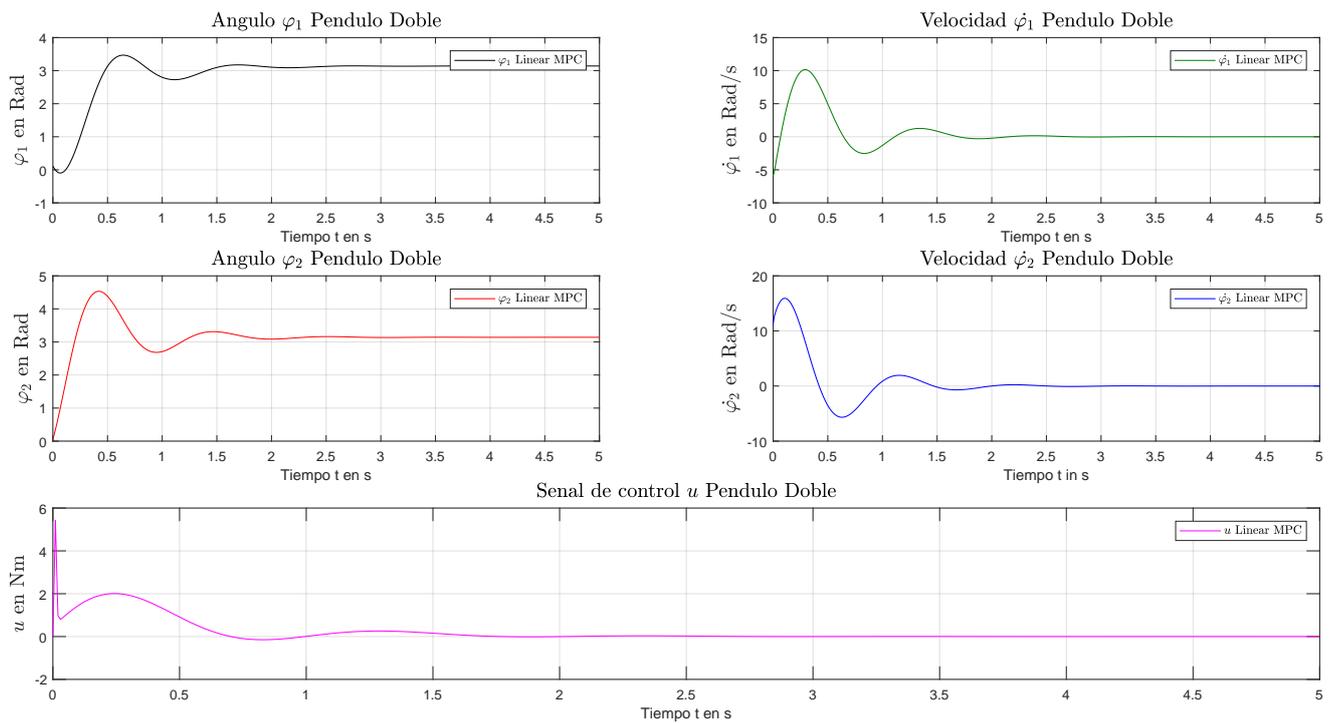


Figura 3.34: Resultado de simulación del MPC lineal (MPC) con $x_O = [0 \ 0 \ 0 \ 0]$

$$y \ x_{PO} = [\pi \ 0 \ \pi \ 0]$$

Fuente : Elaboración propia

Tal como se puede apreciar también en la figura 3.34, aunque los ángulos convergen no precisamente en los puntos de referencia deseados, convergen en valores cercanos a ellos pero como se puede apreciar de la misma forma que en los casos anteriores, más lentos.

Para entender mejor sobre los puntos previamente probados para el MPC lineal, se debe entender que hizo una linealización bajo en un punto de operación específico. Para nuestro caso fue $x_{PO} = [\pi \ 0 \ \pi \ 0]$ y los puntos de pruebas fueron inferiores a ese hasta llegar $x_O = [0 \ 0 \ 0 \ 0]$. De los resultados se ven claramente que la convergencia varía a cuán lejano se encuentre al punto de operación. Precisamente esto se cumple de acuerdo a la teoría de linealización sobre la zona de validación de la linealización [55]. En cuanto más se aleja desde los puntos de linealización se distorsiona y por ende la dinámica ya se pierde. Sin embargo, si queremos utilizar aún así el MPC lineal, se deberá ser uso de un controlador Gain Scheduling la cual por cada rango de operación se utiliza una ganancia para cada rango del sistema linealizado cubriendo la trayectoria esperada [56]. Ahora bien, se presenta otra solución que deja de

lado la linealización del sistema y toma el modelo no lineal. En la siguiente sección se detalla dicha solución.

3.5. Diseño Control No Lineal Predictivo Basado en Modelo (NMPC)

Como se mencionó en el capítulo anterior las limitaciones del MPC lineal sobre sistemas y restricciones no lineales hacen que se contemple otro tipo de estrategia de control que pueda trabajar y tener un buen performance con el *Sistema de péndulo doble* no lineal. Las ecuaciones de movimiento 2.35 y 2.36 describen el modelo del sistema real. Claramente se nota en ellas que poseen términos como seno y coseno que muestran el comportamiento no lineal propio del *Sistema de péndulo doble*. Este tipo de problema de control no lineal pueden resolverse con paquetes de *Solvers* que tienen algoritmos e integradores compatibles con MATLAB [57]. Estos proveen una solución óptima para problemas MPC y los hace adecuados para aplicaciones en tiempo real.

La opción que sugiero para resolver el problema óptimo de control de nuestro sistema no lineal es mediante el uso del software ACADO: <https://acado.github.io/>. Aunque existen otras opciones de solucionadores disponibles, ACADO tiene la capacidad de adaptarse eficientemente a sistemas no lineales, incluyendo controladores no lineales como el NMPC, así como a restricciones no lineales. Además, ACADO ofrece una ventaja en términos de velocidad y eficiencia en comparación con otras herramientas de optimización como CasADi, especialmente en sistemas con dimensiones pequeñas a medianas, es decir, aquellos con menos de 20 variables de estado, una función de costo no cuadrática y un horizonte de predicción corto [58] y [59]. Todas estas características se ajustan perfectamente a las necesidades del problema del *Sistema de péndulo doble*.

3.5.1 Principio básico del Control No-Lineal Predictivo Basado en Modelo

El MPC lineal como se vio en la capítulo anterior, tiene éxito en cierta medida. Sin embargo, es muy usado en la industria como en el sector petroquímicos/-refinería [60]. La limitación, es otros sectores o sistemas a controlar es que un sistema lineal tiene pobre capacidad de predicción. En otras palabras su performance es reducida. Y esta claro señalar que una función de costo cuadrática y restricciones lineales no reflejan las necesidades industriales. Entonces dada

las limitaciones, se debe entender que cambiaría un MPC lineal a diferencia de MPC No-Lineal. Veamos por lo tanto un cuadro comparativo de ambos controladores mostrado en la tabla 3.1 y 3.2 (Fuente: Optimal and Predictive Control [43]).

Nomenclatura Lineal MPC	
· Usa modelo lineal:	$\dot{x} = Ax + Bu$
· Función de costo Cuadrática:	$F(x, u) = x^T Qx + u^T r u$
· Restricciones afines:	$Hx + Gu \leq 0$
Programación cuadrática (Tiempo discreto)	

Tabla 3.1: Nomenclatura de un MPC lineal

Nomenclatura No-Lineal MPC (NMPC)	
· Usa modelo No-lineal:	$\dot{x} = f(x, u)$
· Función costo puede ser No-Cuadrática:	$F(x, u)$
· Restricciones No-lineales:	$h(x, u) \leq 0$
Programación No-Lineal (NLP)	

Tabla 3.2: Nomenclatura del un MPC no lineal

En las tablas 3.2 y 3.1 se pueden observar las diferencias en el modelo, la función de costo y las restricciones utilizadas entre los dos controladores. Sin embargo, ambas estrategias de control siguen una metodología similar para resolver problemas óptimos de control, lo que significa que la secuencia de pasos en el algoritmo presentado en la figura 3.23 es la misma en ambos casos.

En lo que respecta a la resolución del sistema no lineal de péndulo doble, se utiliza el software ACADO, un software de código abierto para control automático y optimización dinámica. Este software se encarga de encontrar la solución óptima al problema de control mediante la formulación de una función de costo que tiene en cuenta las restricciones del sistema y las preferencias del usuario. La solución óptima se obtiene a través de la implementación de algoritmos de optimización numérica, lo que garantiza la eficacia y precisión del proceso de control.

En resumen, el uso de ACADO permite una resolución eficaz del sistema no lineal de péndulo doble mediante la formulación y solución de un problema óptimo de control que tiene en cuenta las restricciones y preferencias del usuario.

Implementación del MPC No-lineal basado en ACADO en MATLAB

A fin de implementar nuestro problema se debe considerar definir nuestra ODE (*Ordinary Differential Equation*), establecer las variables de estado ($\varphi_1, \dot{\varphi}_1, \varphi_2, \dot{\varphi}_2$), la entrada de control u y la perturbación d .

Definimos entonces los estados y la entrada de control de la siguiente manera:

```

1 % Sistema Pendulo doble Master PUCP
2 %=====Author: Carlos Raymundo =====%
3 %=====Year :2023===== %
4 clc;
5 clear all;
6 close all;
7
8 % Condition for compiling
9 EXPORT = 1;
10 DifferentialState phi1 dphi1 phi2 dphi2; % define states
11 Control u % define controls
12 OnlineData d; % Disturbance d *0.001*d

```

Luego el modelo no lineal de la ODE definido en las ecuaciones 2.35 y 2.36 así:

```

1 % Differential Equation
2 f = dot([phi1; dphi1; phi2; dphi2]) == [dphi1;
3 (375*((9*cos(phi1 - phi2)*sin(phi1 - ...
4 phi2)*dphi1^2)/1250 ...
5 + (3*sin(phi1 - phi2)*dphi2^2)/625 - (4*u)/5 + ...
6 (8829*sin(phi1))/12500 ...
7 - (8829*cos(phi1 - ...
8 phi2)*sin(phi2))/25000))/((27*cos(phi1 - ...
9 phi2)^2)/10 - 24/5);
10 dphi2;
11 -(1250*((18*sin(phi1 - phi2)*dphi1^2)/3125 ...
12 + (27*cos(phi1 - phi2)*sin(phi1 - ...
13 phi2)*dphi2^2)/12500 ...
14 - (8829*sin(phi2))/31250 + (79461*cos(phi1 - ...
15 phi2)*sin(phi1))/250000 ...
16 - (9*u*cos(phi1 - phi2))/25))/((27*cos(phi1 - ...
17 phi2)^2)/10 - 24/5)];

```

Posteriormente definimos el horizonte de predicción, el tiempo de muestreo,

la función de costo así como el número de variables de estado y entrada de control.

```

1 %define the step length
2 % =====
3 simTs = 0.01;           % time interval
4 Num = 30;
5 numSteps = 5;
6 %define the objective function ( stage and end costs)
7 h = [diffStates; controls];
8 hN = [diffStates];
9 % Define number of states and control signal
10 n_XD = length(diffStates);
11 n_U = length(controls);

```

Y añadimos los parámetros como el punto inicial, la referencia y los pesos para la matriz Q que se halló de forma experimental obteniendo el mejor resultado que mostrará posteriormente.

```

1 % Simulation Parameters
2 % =====
3 N=30;
4 dt =0.01;           % Amount of punts
5 input.N =30;       % prediction Horizon
6 X0 = [3 0 pi 0 ];  % Initial Point
7 Xref = [pi 0 pi 0]; % Reference
8 input.x = repmat(Xref,N+1,1);
9 Xref = repmat(Xref,N,1);
10 Uref = zeros(N,n_U);
11 input.u = Uref;
12 input.y = [Xref(1:N,:) Uref];
13 input.yN = Xref(N,:);
14 input.W = diag([10, 0.01, 10 0.01 1]);
15 input.WN = diag([10, 0.01, 10 ,0.01]); %

```

Posteriormente, como se menciona sobre los métodos numéricos para la solución de la OCP, se elige una discretización de tipo Multiple Shooting y la aproximación Gauss Newton para la obtención de la solución óptima local como se muestra a continuación:

```

1 % Configuration MPC
2 % =====
3 ocp = setModel(f);
4 mpc = acado.OCPexport(ocp);
5 mpc.set('HESSIAN_APPROXIMATION', 'GAUSS_NEWTON');
6 mpc.set('DISCRETIZATION_TYPE', 'MULTIPLE_SHOOTING');

```

Ahora se construye el algoritmo para el NMPC. Como se puede notar en los punto 20 se obtiene del vector de control óptimo y de ella se toma la primera salida de control tal como se menciona en la formulación matemática del MPC. Por consiguiente, esta es reutilizada en la siguiente iteración tal como se vio en la sección 3.4 y se calcula en base a la entradas y salidas de las variables anteriores, la nueva salida de control repetitivamente.

```

1 while time(end) < Tf
2 %Solve NMPC OCP
3     input.x0 = state_sim(end,:);
4     output = acado_doble(input); % Solve our OCP
5
6     for k = 1:length(N)
7         output1 = [output1; output.x(1,:)];
8         k=k+1;
9     end
10 % Optimal input
11     input.x = [output.x(2:end,:); output.x(end,:)];
12     input.u = [output.u(2:end,:); output.u(end,:) ];
13     input_sim.x = state_sim(end,:)';
14 % NMPC control
15     input_sim.u = output.u(1,:)';
16     input_sim.od = 10;
17
18     for k = 1:length(N)
19         input1 = [input1; input.x(1,:)];
20         k=k+1;
21     end
22
23 % Simulate our system
24     states = integrate_doble(input_sim);
25     state_sim = [state_sim; states.value'];
26 % Save the MPC step
27     INFO_MPC = [INFO_MPC; output.info];
28     KKT_MPC = [KKT_MPC; output.info.kktValue];

```

```

29     controls_MPC = [controls_MPC; output.u(1,:)];
30     controls = [controls; output.u(1,:)];
31     % Update iteration
32     iter = iter+1;
33     nextTime = iter*dt;
34     time = [time nextTime];
35     result.CalTime = output.info.cpuTime;
36     itera = [itera iter];
37     cpuTime = [cpuTime output.info.cpuTime];
38     % Time display
39     disp(['current time: ' num2str(nextTime) ' ' ...
          char(9) ' (RTI step: ' ...
          num2str(output.info.cpuTime*1e6) ' s )'])
40 end

```

Los resultados del MPC No-lineal basado en ACADO

Para seguir la línea de los mismos puntos de operación tanto para los controladores lineales. Para este NMPC basado en ACADO probaremos los mismo puntos de operación y restricciones como se pueden apreciar en el siguiente cuadro:

$$\begin{aligned}
 \mathbf{X}_0 &= [3 \ 0 \ \pi \ 0] \\
 \mathbf{X}_{PO} &= [\pi \ 0 \ \pi \ 0] \\
 x_{\min} &= [-\pi, -\pi, -\pi, -\pi]^T, \quad x_{\max} = [\pi, \pi, \pi, \pi]^T \\
 u_{\min} &= -10, \quad u_{\max} = 10
 \end{aligned} \tag{3.69}$$

Una vez que se ejecuta el algoritmo basado en ACADO, se obtiene los resultados para los ángulos φ_1 y φ_2 , las velocidades $\dot{\varphi}_1$ y $\dot{\varphi}_2$, y la entrada de control u como se muestra en la figura 3.36.

Ahora si a comparación del MPC lineal, el MPC no lineal basado en ACADO tiene una mejor performance como se puede apreciar en las figuras. Los resultados son sin duda muy alentadores. Veamos el caso del ángulo φ_1 que empieza con un valor de 3 radianes, llega al valor esperado de π radianes en un tiempo muy óptimo ya que lo realiza en 0.83 segundos aproximadamente. Una diferencia bastante con el MPC lineal que se demora casi el doble en llegar a su punto de equilibrio. Cabe también resaltar que la respuesta no presenta

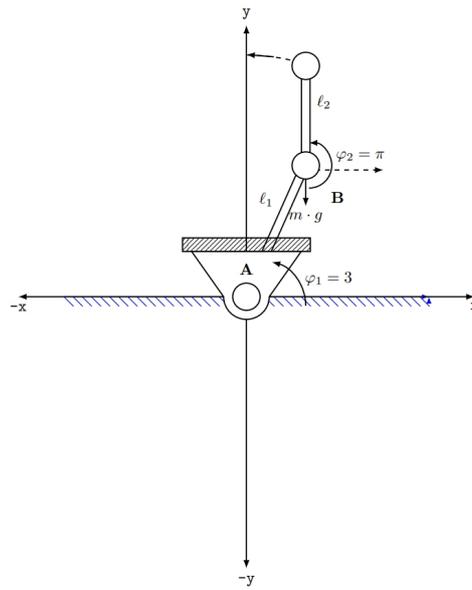


Figura 3.35: Los resultados de la simulación con $x_O = [3 \ 0 \ \pi \ 0]$
 y $x_{PO} = [\pi \ 0 \ \pi \ 0]$
Fuente : Elaboración propia

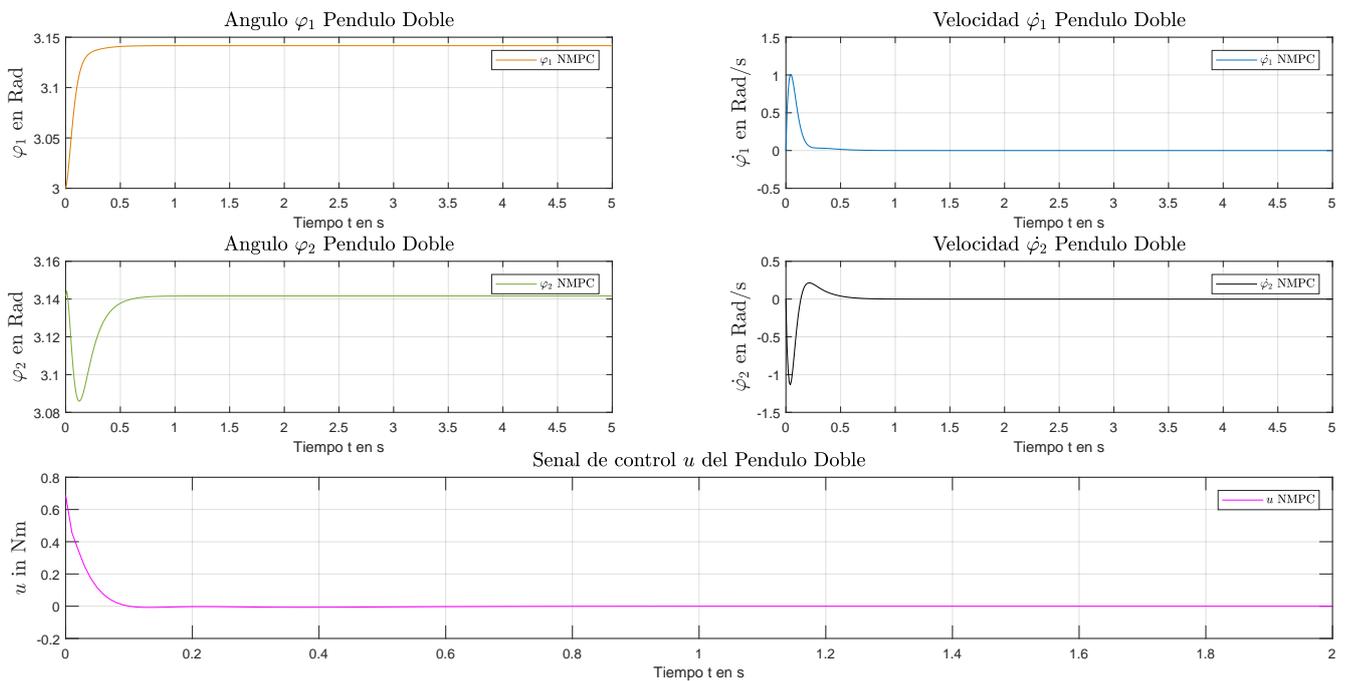


Figura 3.36: Resultado de simulación del NMPC basado en ACADO
Fuente : Elaboración propia

oscilaciones como en el caso del MPC lineal. En cuanto al ángulo φ_2 , el punto inicial π que alcanza el valor esperado para este controlador no lineal en 0.84

segundos aproximadamente. Y sin una oscilación tan marcada como lo es con el MPC lineal. Ahora, en cuanto a la señal de entrada de control u se estabiliza en las 0.13 segundos y sin oscilación. Si hablamos de las velocidades $\dot{\phi}_1$ y $\dot{\phi}_2$ claramente se observan que tienen una respuesta mucho más rápida y sin oscilaciones como sí las tiene su par MPC lineal.

Los buenos resultados que observamos en la respuesta se debe a que ACADO no utiliza un modelo linealizado como en los otros casos, sino una ecuación diferencial algebraica que describe el comportamiento matemático del *Sistema de péndulo doble* con sus características propias. ACADO utiliza el método de Runge-Kutta para la resolución numérica de ecuaciones diferenciales por eso razón se provee al algoritmo la ODE como modelo. En nuestro caso usamos Runge-Kutta implícito por ser un sistema denominado rígido que al ser resuelto por métodos numéricos estándar son estables solo por un corto intervalo de tiempo. De modo este método puede mejorar la estabilidad y la convergencia del sistema [61]. Para ser específicos usamos el método Radau IIA5 para solución de la ODE ya que en cierta forma la solución de las ODEs que tienen una parte no lineal, después de ser resueltas por métodos numéricos, pueden presentar inestabilidad (oscilaciones en su aproximación). Ya que como se comentó antes el sistema no lineal descrito por las ecuaciones diferenciales 2.35 y 2.36 donde los senos y cosenos representan la no linealidad de la dinámica del sistema. Por lo tanto, la solución de las ODEs por medio de Runge Kutta implícito es la mejor alternativa para ODEs que presentan rigidez (Stiff Differential Equation). Cabe señalar que el número del orden "5", se eligió ya que desde ese orden se tiene una convergencia al valor esperado y su implementación dentro de ACADO se puede fácilmente seleccionar [62] y [61].

Al igual que el MPC lineal el horizonte de predicción es de 30 pasos o intervalos y un tiempo de muestreo de 0.01 segundos. Y como se mencionó antes se usó el método de discretización "*Multiple Shooting*" usando los datos anteriores que son efectivos para el tipo de sistema no lineal que a diferencia del MPC lineal, sólo abarca un punto específico de operación por lo que el NMPC basado en ACADO según los resultados antes expuestos resulta ser más eficaz.

Como se puede apreciar en los puntos de operación iniciales empieza desde una posición inestable con $[3 \ 0 \ \pi \ 0]$, mientras que la referencia también es inestable $[\pi \ 0 \ \pi \ 0]$ pero el controlador NMPC presenta buen resultado a pesar que hay restricciones en la entrada acotada y las restricciones en las variables

de estado como se ven en la ecuación 3.59. Otro punto interesante en notar que las matrices Q y R no tuvieron un cambio extremo desde su estado inicial 3.60 ya que de por sí, la posición de los ángulos φ_1 y φ_2 tuvo su punto de convergencia bastante rápido y la matriz Q le afectaría directamente a las variables de estado; del mismo modo, para la entrada de control u , la matriz R no tendría un cambio importante en la respuesta de la misma.

En cuanto la entrada de control u , se nota un cambio importante ya que con el MPC lineal se necesitaba de un máximo de 0.23 Nm para que los ángulos convergen. Con el *solver* ACADO se necesita solamente alrededor de 0.75 Nm. Por lo tanto se encontró la señal de control óptima que minimiza la función de costo y a su vez satisface las restricciones en las variables de estado y de control y por ende requiere más potencia al potencia al motor. Además, debido a que la dinámica del sistema es no lineal es problema que nos enfrentamos en No-Convexo, hemos resuelto con un estándar SQP usando la aproximación Gauss-Newton que secuencialmente aproxima la NLP secuencialmente por sub QP convexos [63]. Aunque existe otro que se denomina Interior Point (IP) [64] también para problemas no convexos pero que no cubre este *solver*. Quizás sea también otro tipo de estudio a futuro el comparar esta técnica de aproximación.

Algo que sí se llevo a cabo para este tema de estudio, es la perturbación en el sistema de control. Este punto se lleva a cabo en el apartado sobre las incertidumbres. Sin embargo, es importante notar que en estos resultados la perturbación recibe la denominación d que para probar el *solver* en primera instancia lo mantenemos en 0. Posteriormente se prueba con diferentes perturbaciones y probamos a su vez la robustez del NMPC. Las simulaciones se llevaron a cabo con una Laptop HP con 1.80 GHz, procesador Intel Core i5-8250U y 8GB de RAM, de donde podemos ver que la media del tiempo de computo es de 210 μs que se demora el *solver* ACADO para resolver el NMPC. En la figura 3.37 se puede observar el CPU Time por cada iteración a lazo cerrado con un horizonte de predicción $N=30$ sin perturbación como se mencionó antes.

3.5.2 Robustez inherente del NMPC vs MPC

La idea principal que se presenta es que, en los capítulos anteriores, se ha demostrado la necesidad de que los controladores adopten una estrategia

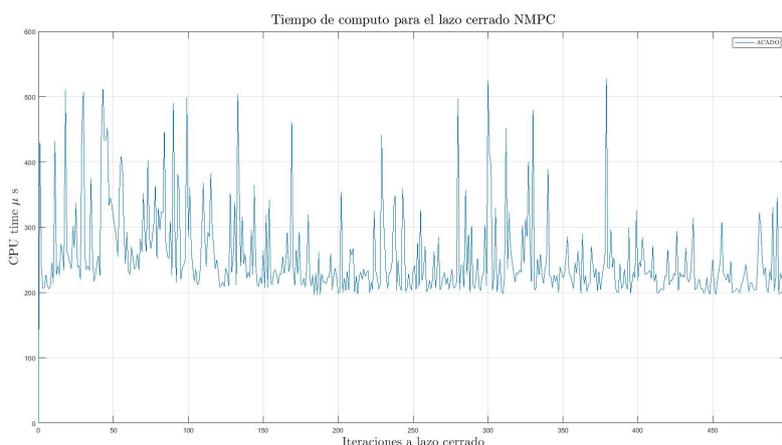


Figura 3.37: Tiempo de cómputo $210 \mu\text{s}$ para el NMPC a lazo cerrado implementado con ACADO
Fuente : Elaboración propia

adicional para mantener los estados de equilibrio en una zona segura y estable. No obstante, el NMPC cuenta con una robustez que le permite hacer frente a las diferentes incertidumbres W o errores que puedan afectar el comportamiento normal del sistema. Además, se utiliza un término final en las funciones de costo que garantiza que el sistema permanezca dentro del área de estabilidad y convergencia.

Error en el modelo del NMPC - MPC

El modelo del sistema es crucial para el controlador, ya que este debe reaccionar de manera similar en tiempo real a como lo hace con el modelo experimental. Los controladores MPC o NMPC crean un modelo interno del sistema que es una representación casi exacta del sistema real, por lo que las predicciones deberían coincidir con las mediciones en tiempo real. Sin embargo, en la práctica, existen errores de modelización y ruido en los datos que deben ser considerados [65]. La figura mostrada en [43] ilustra la diferencia entre la predicción ideal y la real de un controlador MPC/NMPC.

La figura 3.38 muestra que la predicción de un MPC real puede desviarse de la del MPC nominal debido a errores en el modelo. Sin embargo, la medición actual se utiliza como condición inicial para la siguiente predicción, lo que permite que el MPC real se ajuste y corrija cualquier desviación anterior. Esto se conoce como la robustez inherente del MPC, y si el error de modelado es mínimo, el MPC real puede funcionar igual de bien que el nominal [66]. En la

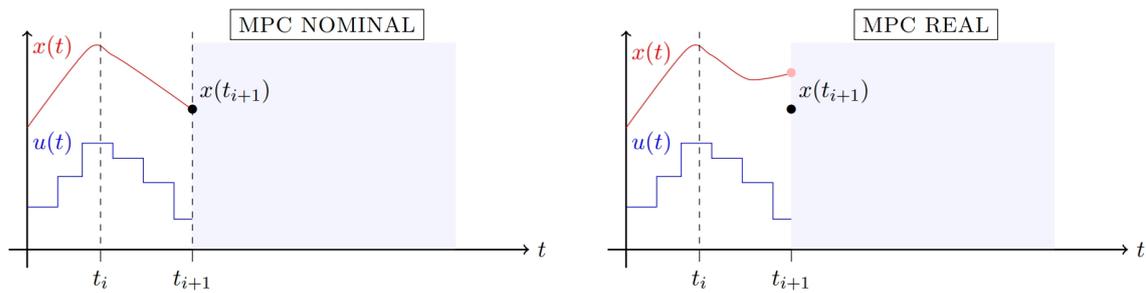


Figura 3.38: Estándar MPC nominal versus MPC real
Fuente : Optimal and Predictive Control, R. Findeisen

práctica, a menudo se utiliza un MPC nominal con buenos resultados. En el algoritmo, se modelan matemáticamente las incertidumbres, como errores de modelado o ruido.

Resultado de prueba de robustez del controlador NMPC

La instalación y configuración de ACADO para Matlab se muestra en el Anexo. En esta sección se demostrará cómo el NMPC reaccionaría ante diferentes tipos de perturbaciones en la señal de control, y se explicará que los errores e incertidumbres son inherentes al mundo real. Para fines experimentales, no se considerarán restricciones en la entrada de control o variables de estado. En la primera prueba, se introducirá una perturbación aditiva con valores aleatorios en el sistema de control, y los resultados se mostrarán en la figura 3.39.

En los resultados mostrados en la figura podemos apreciar que una perturbación aditiva ingresa al sistema de control que se suma a la entrada de control que afectaría directamente a la performance del NMPC y es así como se llevará a cabo para propósitos experimentales de esta tesis. Como resultado de esta perturbación aditiva, el controlador NMPC reacciona con retardo. En efecto, se observa claramente que el ángulo φ_1 llega al valor esperado de π radianes en un tiempo considerablemente tardío ya que lo realiza en 2.7 segundos aproximadamente. Una diferencia bastante con el MPC lineal que se demora casi el doble en llegar a su punto de equilibrio o con el NMPC sin perturbación. En cuanto al ángulo φ_2 el valor esperado para este controlador no lineal con perturbaciones es de 2.8 segundos aproximadamente. Para el caso de ambos ángulos antes de conseguir su estado de equilibrio tuvieron valores bastantes oscilaciones debido a que la perturbación afecta a directamente a las las variables de estado y sus derivadas. Por ejemplo, el ángulo φ_1 que

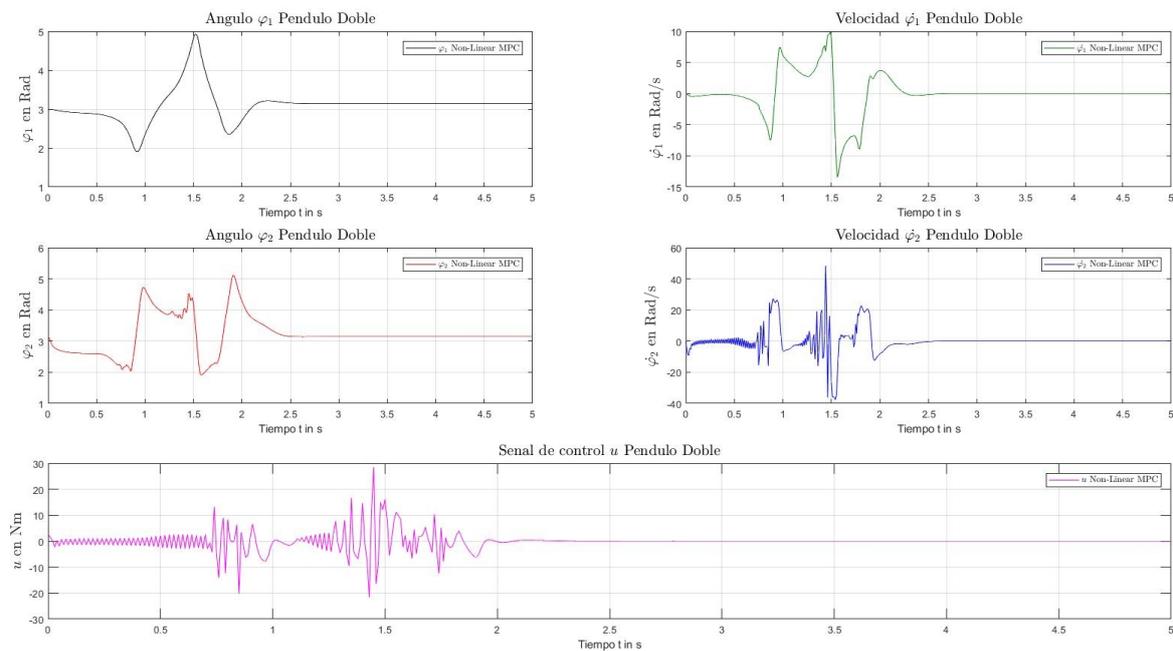


Figura 3.39: Resultado de simulación del MPC No-lineal (NMPC) con una perturbación con valores aleatorios basado en ACADO

Fuente : Elaboración propia

representa la posición del primer péndulo, tiene un movimiento leve en el sentido horario hasta el valor aproximadamente de 2 radianes, para luego tomar el momento a pesar de la perturbación y llevar nuevamente al péndulo a un valor de ángulo de 5 radianes. Este movimiento de extremo a extremo hace que el ángulo pueda finalmente pueda llegar al valor deseado. Lo mismo sucede con el ángulo φ_2 que da un movimiento oscilatorio más marcado ya que este ángulo representa al segundo péndulo el que recibe el mayor momento de inercia en base al movimiento que recibe el sistema. A esta característica añadimos la perturbación d en la entrada de control empeorando en cierto modo al sistema ya que como se puede apreciar, la entrada de control varia peligrosamente de -27 Nm a 27 Nm pero finalmente al termino del tiempo antes mencionado, termina en el punto de equilibrio deseado como también se puede apreciar.

El controlador reacciona ante una perturbación en el sistema, es necesario recordar que el método utilizado para resolver el problema óptimo de control (OCP) requiere que el valor inicial de la variable de estado siempre se conozca. De esta manera, el valor de la señal de control se calcula y predice antes de

que la perturbación ingrese al sistema. El objetivo es mantener el valor de la señal de control en 0 Nm, lo cual es evidente en los resultados obtenidos. Se puede concluir, en base al buen comportamiento del controlador, que una vez que el controlador aprende y predice la señal de control, puede enfrentar perturbaciones. Las simulaciones se realizaron utilizando una Laptop HP con procesador Intel Core i5-8250U y 8GB de RAM, y se tardó en promedio $228 \mu\text{s}$ en resolver el NMPC mediante el solver ACADO. En la figura 3.40 se muestra el tiempo de computación por cada iteración a lazo cerrado con un horizonte de predicción $N=30$ y una perturbación aleatoria, y se puede observar que el solver ACADO tarda en promedio $228 \mu\text{s}$.

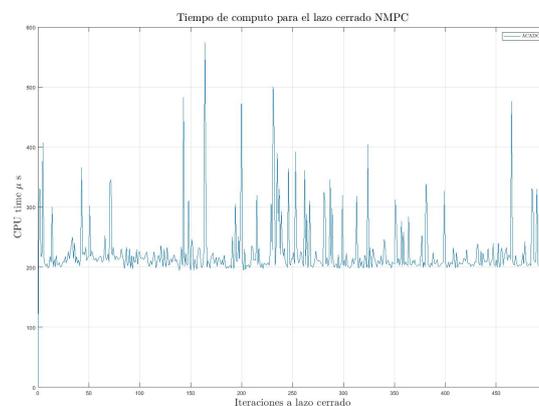


Figura 3.40: CPUTime $228 \mu\text{s}$ para el NMPC con perturbación aleatoria
Fuente : Elaboración propia

Ahora añadamos un ruido blanco Gaussiano de -6dB complejo cuyo periodograma se muestra en la figura 3.41.

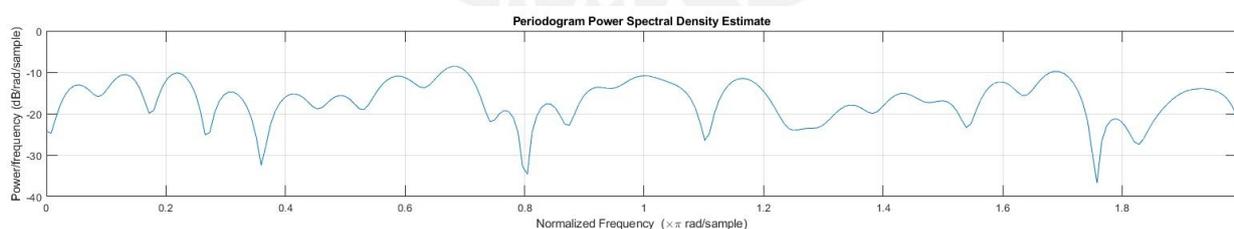


Figura 3.41: Ruido blanco Gaussiano con -6 dB
Fuente : Elaboración propia

La respuesta a esta perturbación es de la siguiente manera mostrado en la figura 3.42.

Vemos por lo tanto que el NMPC reacciona también a esta perturbación solo que a diferencia de la perturbación aleatoria lo hace en menos tiempo como se

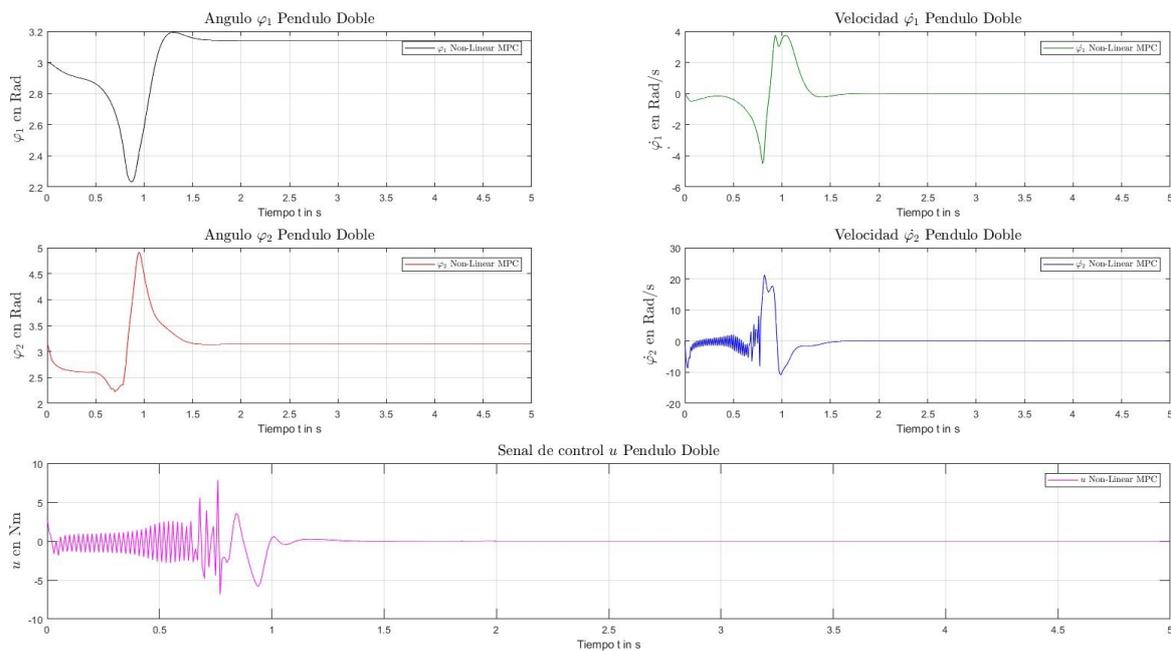


Figura 3.42: Resultado de simulación del MPC No-lineal (NMPC) con un ruido blanco Gaussiano de -6dB basado en ACADO

Fuente : Elaboración propia

puede apreciar. En efecto, se aprecia que el ángulo φ_1 y en el ángulo φ_2 a los 1.8 segundos y 2.1 segundos respectivamente. Esto se debe a que el ruido blanco Gaussiano tiene valores aleatorios que no guardan correlación estadística entre sus valores y su valor de potencia en los distintos anchos de banda. Por lo tanto este disturbio al ser recurrente ya es predicho por el NMPC y su reacción a este ruido blanco Gaussiano de -6dB no presenta oscilaciones en las variables de estados como en los casos anteriores pero la perturbación que ingresan al sistema de control hacen que el rango de la señal varíe casi de forma constante con valores que van de los -7 Nm a 7 Nm como máximo en comparación con el caso anterior. Si pensamos que pasaría con el actuador al recibir esta señal podría afectar al sistema. Las simulaciones se llevaron a cabo de la misma forma como la simulación anterior con una Laptop HP con 1.80 GHz, procesador Intel Core i5-8250U y 8Gb de RAM, de donde podemos ver que la media del tiempo de computo es de $217 \mu\text{s}$ que se demora el Solver ACADO para resolver el NMPC. En la figura 3.43 se puede observar el CPU Time por cada iteración a lazo cerrado con un horizonte de predicción $N=30$ con un ruido blanco Gaussiano.

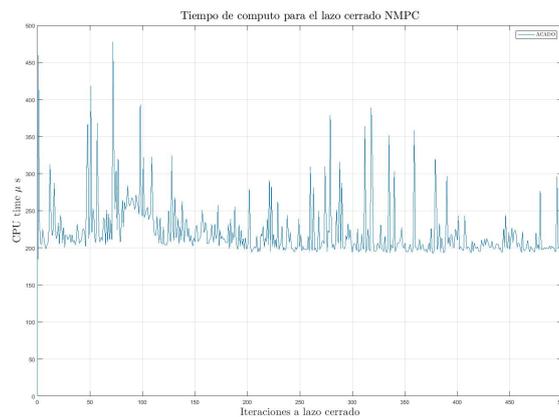


Figura 3.43: CPUTime 217 μ s el NMPC con ruido blanco Gaussiano a - 6dB

Fuente : Elaboración propia

3.6. Comparación de los controladores y resultados finales

Finalmente, realizamos una comparación a los 5 tipos de controladores que vimos en esta tesis, es decir LQR, Observador con controlador de estado, Observador reducido con controlador de estado, MPC y NMPC .

Si bien es cierto que los modelos de los 4 primeros controladores son lineales y el último no lineal; para fines académicos y de muestra sobre la efectividad de un controlador sobre el otro, se hicieron las pruebas en el mismo punto de operación, los mismos puntos iniciales y de referencia. De modo que todos los modelos en este punto de operación se aproximan bastante bien como se vio en la capítulo 2. Los resultados los podemos apreciar en la figura 3.44.

De hecho podemos concluir lo antes mencionado ya que el NMPC utiliza un modelo completamente no lineal que se aproxima mucho a la dinámica real del sistema de péndulo doble y fácilmente podría probarse en la práctica. Aunque en cierta forma el MPC que utiliza un modelo lineal al tener pequeños error de modelado, pueda usarse sin problemas en la práctica considerando las restricciones en las variables de estado y entrada de control.

A fin de detallar más los resultados, se puede apreciar en la figura 3.44. Veamos lo que sucede en la primera variable de estado, es decir el ángulo de posición del primer péndulo φ_1 , que para el caso del NMPC, la convergencia es más rápida en comparación al MPC lineal, seguido por el LQR, Observador y finalmente el Observador reducido. Si analizamos más a fondo los resultados, notamos que el MPC lineal y el LQR tienen las mismas características. Esto

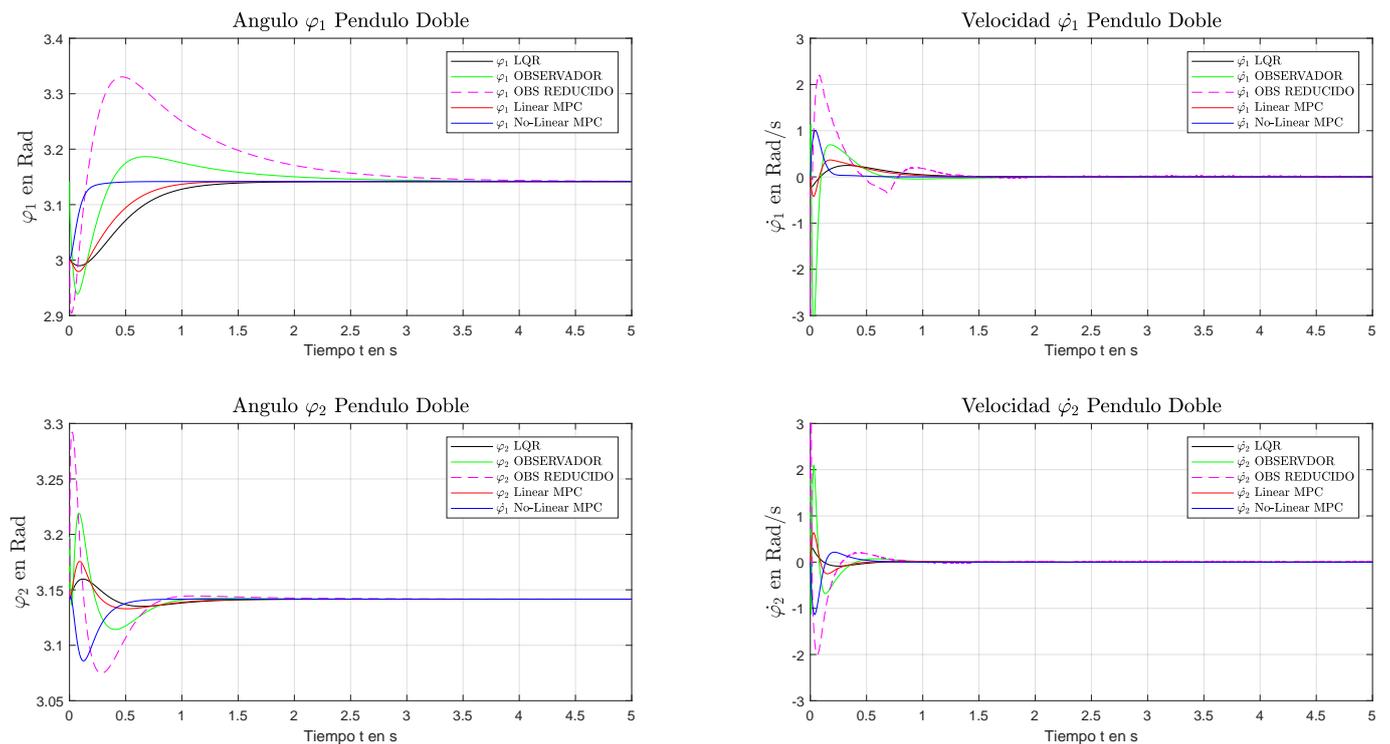


Figura 3.44: Comparación de los controladores LQR, Observador, Observador reducido, MPC y NMPC
Fuente : Elaboración propia

es claro ya que utilizan el mismo modelo continuo; uno varía del otro por poco, por ejemplo en el tiempo de establecimiento. Claramente, esto podría mejorarse con las matrices Q y R para ambos casos ya que si recordamos el algoritmo del MPC lineal, este utiliza el comando QUADPROG que penaliza tanto las variables de estado y/o la señal de entrada tal como lo hace el LQR. En cuanto a los Observadores controlador completo y reducido, tienen un respuesta mucho más *Overshooting* y más pronunciado que los hace menos efectivo que los otros controladores. Para entender lo que sucede con ellos, debemos tener en cuenta que los Observadores tienen por objetivo obtener variables que no son medibles, por esa razón el Observador completo toma más tiempo ya que mide todas las variables en base al modelo y el Observador reducido solo algunas de ellas. En este punto por lo tanto radica la velocidad de respuesta. De la misma forma, en cuanto al ángulo del segundo péndulo φ_2 , el controlador no lineal NMPC tiene una mejor performance frente a los otros controladores. En este caso, el Observador Reducido es aquel que tiene el resultado menos efectivo considerando que este segundo ángulo es que

recibe la mayor momento de inercia. Para el caso del LQR y el Observador completo tienen mejor resultado que este último pero no se compara al buen performance del NMPC.

Se nota claramente que el controlado del NMPC basado en ACADO tiene una mejor performance sobre los otros controladores. Es mucho más rápido y converge al punto de equilibrio de la misma manera. Por lo tanto, este controlador resulta ser el más adecuado para este tipo de *Sistema de péndulo doble*. En la siguiente sección se verá más a detalle los indicadores que señalan la performance de los controladores estudiados de forma cuantitativa.

3.6.1 Indicadores de performance de los controladores estudiados

A continuación vamos a comprobar la performance de los 5 controladores. Para eso se deben establecer indicadores que los miden.

RMSE para medir la precisión del modelo

El indicador RMSE (Root Mean Square Error) o Error medio cuadrático promedio. Como su nombre menciona se mide el error del valor predicho sobre el valor real o conocido. Este indicador es eficaz para medir la precisión del modelo usado en el control optimal presente en los controladores antes mencionados ya que precisamente que el control optimal utiliza un modelo para predecir el comportamiento del sistema y por esa razón este indicador es de suma importancia [67].

Tabla 3.3: Indicador RMSE para el modelo optimal

	LQR	OBS	OBSRED	MPC	NMPC
φ_1 (rad)	0.0320	0.0266	0.0527	0.0106	0.0267
$\dot{\varphi}_1$ (rad/s)	0.0531	0.2173	0.2866	0.0866	0.0674
φ_2 (rad)	0.0027	0.0087	0.0157	0.0069	0.0039
$\dot{\varphi}_2$ (rad/s)	0.0293	0.1406	0.8822	0.0905	0.0494
RMS(prom)	0.0294	0.0983	0.3093	0.0486	0.0369

Este indicador viene de la raíz cuadrada del cuadrado de la suma de las diferencias entre los valores reales y predichos, la cual es elevada al cuadrado dividido por la cantidad de valores observados. Este valor fue evaluado para cada controlador optimal y de la tabla 3.3 se puede apreciar los resultados. A fin de interpretar este valor, es importante notar que cuando más pequeño

es el RMSE, más preciso es el modelo. Por lo tanto, se nota claramente que el LQR y el NMPC poseen los valores más pequeños en comparación de los demás controladores. Ambos modelos se diferencian en que el LQR se base en un modelo lineal y el NMPC se basa en un modelo no lineal. El resto de los controladores poseen valores aunque también pequeños, no tan óptimos como los últimos mencionados.

Indicadores para medir la performance de los controladores

Introducimos ahora otros indicadores que miden la performance los controladores. En este caso usaremos el ISE (Integral Square Error) y IAE (Integral Absolute Error). Ambos indicadores de performance indican el mejor performance cuando poseen valores pequeños [68].

Por otro lado, incluimos también otros indicadores como el Overshooting (OS %), Peak time (PT) y Settling Time (ST). Los resultados se ve en la tabla 3.4 donde se puede apreciar que nuevamente el NMPC y el LQR poseen los valores mas pequeños para el ISE y IAE. Sin embargo, debemos analizar los otros indicadores también descritos. Precisamente, según la tabla se puede apreciar que ambos controladores para el primer ángulo del sistema de péndulo doble φ_1 no poseen ningún tipo de Overshooting pero para el segundo ángulo φ_2 el controlado LQR posee menos OS en comparación con el NMPC. Ahora bien, en cuando al tiempo de establecimiento es claro que el NMPC como respuesta es el que tiene mejor resultado.

Ante lo antes indicado, el controlador que posee una mejor performance ya que su ISE, IAE, OS, PT y ST es definitivamente el No lineal Model Predictive Controller (NMPC). Por lo tanto, el controlador NMPC basado en ACADO nuevamente comprueba que que tiene una mejor performance sobre los otros controladores, punto que también se concluyo en el analices cualitativo descrito en la sección anterior.

Tabla 3.4: Indicadores de la performance de los controladores

	Variabes	OS%	PT(s)	ST(s)	ISE	ISE prom	IAE	IAE prom
LQR	φ_1 (rad)	0	0	2.82	0.0085	0.0103	0.0827	0.0755
	$\dot{\varphi}_1$ (rad/s)	N.A	N.A	2.19	0.0271		0.1619	
	φ_2 (rad)	0.58	0.12	2.36	0.0001		0.0082	
	$\dot{\varphi}_2$ (rad/s)	N.A	N.A	1.75	0.0056		0.0492	
OBS	φ_1 (rad)	1.43	0.66	3.19	0.0071	0.1662	0.0873	0.1841
	$\dot{\varphi}_1$ (rad/s)	N.A	N.A	2.09	0.4717		0.3985	
	φ_2 (rad)	2.47	0.08	2.92	0.0007		0.0206	
	$\dot{\varphi}_2$ (rad/s)	N.A	N.A	1.77	0.1854		0.2300	
OBSRED	φ_1 (rad)	5.97	0.50	3.75	0.0275	0.9601	0.2268	0.3643
	$\dot{\varphi}_1$ (rad/s)	N.A	N.A	1.52	0.6588		0.6482	
	φ_2 (rad)	4.79	0.03	3.67	0.0025		0.0378	
	$\dot{\varphi}_2$ (rad/s)	N.A	N.A	1.92	3.1518		0.5445	
MPC	φ_1 (rad)	0	0	1.57	0.0010	0.0396	0.0657	0.0862
	$\dot{\varphi}_1$ (rad/s)	N.A	N.A	1.59	0.0750		0.1835	
	φ_2 (rad)	1.09	0.11	1.86	0.0005		0.0099	
	$\dot{\varphi}_2$ (rad/s)	N.A	N.A	1.61	0.0820		0.0858	
NMPC	φ_1 (rad)	0	0	0.83	0.0071	0.0193	0.0126	0.0713
	$\dot{\varphi}_1$ (rad/s)	N.A	N.A	0.88	0.0455		0.1151	
	φ_2 (rad)	1.03	0.09	0.84	0.0002		0.0134	
	$\dot{\varphi}_2$ (rad/s)	N.A	N.A	0.94	0.0245		0.1436	

3.7. Análisis del NMPC basado en ACADO con restricciones y diferentes puntos de operación

Hemos determinado que el NMPC basado en ACADO se presenta como el mejor controlador no lineal para un sistema de péndulo doble. Sin embargo, solo se probó este mismo junto con los demás controladores dentro de un rango relativamente pequeño para fines de comparación y performance como se vieron anteriormente. Ahora bien, a continuación probaremos diferentes de operación para el *Sistema de péndulo doble* no lineal. Además se incluyen las restricciones que en este caso se harán en la señal de entrada de control. Por lo tanto, empezaremos con los siguientes puntos:

$$\begin{aligned}
 \mathbf{X}_0 &= \left[\frac{\pi}{2} \ 0 \ \frac{\pi}{2} \ 0 \right] & \mathbf{X}_{PO} &= [\pi \ 0 \ \pi \ 0] \\
 x_{\min} &= [-\pi, -\pi, -\pi, -\pi]^T, & x_{\max} &= [\pi, \pi, \pi, \pi]^T \\
 u_{\min} &= -9, & u_{\max} &= 9
 \end{aligned} \tag{3.70}$$

La representación gráfica del *Sistema de péndulo doble* con estos puntos de operación se aprecian en la figura 3.45 se puede ver la representación gráfica

del Sistema de péndulo doble.

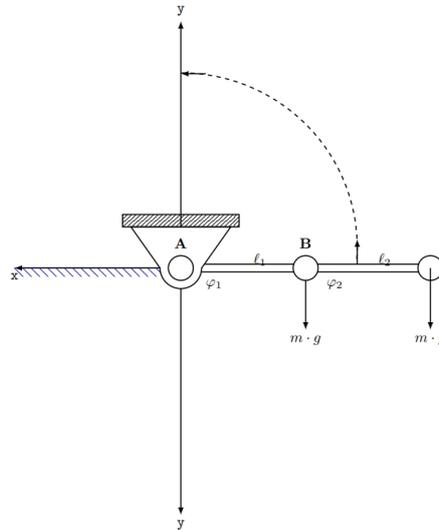


Figura 3.45: Representación de un Sistema de péndulo doble con $x_O = [\frac{\pi}{2} \ 0 \ \frac{\pi}{2} \ 0]$ y $x_{PO} = [\pi \ 0 \ \pi \ 0]$
Fuente : Elaboración propia

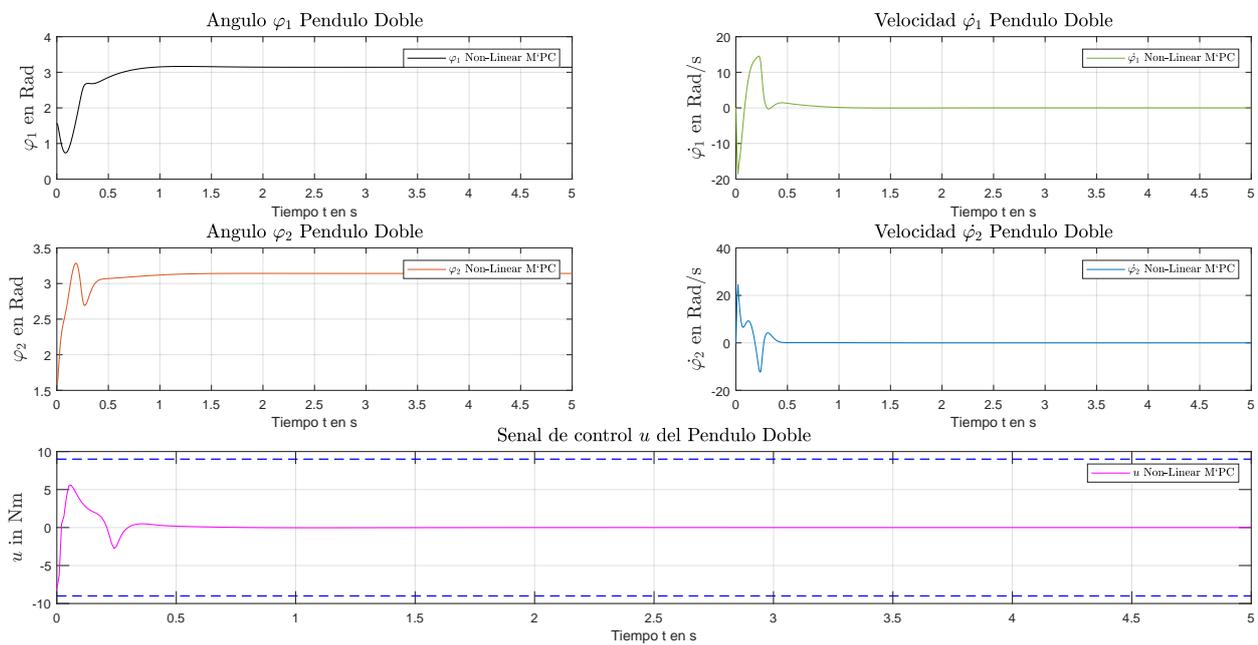


Figura 3.46: Resultado de simulación del MPC No-lineal (NMPC) con $x_O = [\frac{\pi}{2} \ 0 \ \frac{\pi}{2} \ 0]$ y $x_{PO} = [\pi \ 0 \ \pi \ 0]$
Fuente : Elaboración propia

Una vez configurado los nuevos parámetros, se dispone a probar el algoritmo y

el resultado se muestra en la 3.46 que el primer ángulo φ_1 converge a su punto de operación a los 1.62 segundos y el segundo ángulo φ_2 a los 1.25 segundos. Cabe señalar que las restricciones las cuales dan un buen resultado van de los -18Nm a 18Nm y el tiempo de computo CPUTime promedio fue de 221.4 μ s.

A continuación, configuramos los nuevos parámetros, el resultado se muestra en la 3.48 en donde el primer ángulo φ_1 converge a su punto de operación a los 2.53 segundos y el segundo ángulo φ_2 a los 2.8 segundos. Las restricciones fueron de -31 Nm a 31 Nm y el tiempo de computo CPUTime promedio fue de 405.8 μ s.

$$\begin{aligned} \mathbf{X}_0 &= [0 \ 0 \ 0 \ 0] & \mathbf{X}_{PO} &= [\pi \ 0 \ \pi \ 0] \\ x_{\min} &= [-\pi, -\pi, -\pi, -\pi]^T, & x_{\max} &= [\pi, \pi, \pi, \pi]^T \\ u_{\min} &= -6, & u_{\max} &= 6 \end{aligned} \quad (3.71)$$

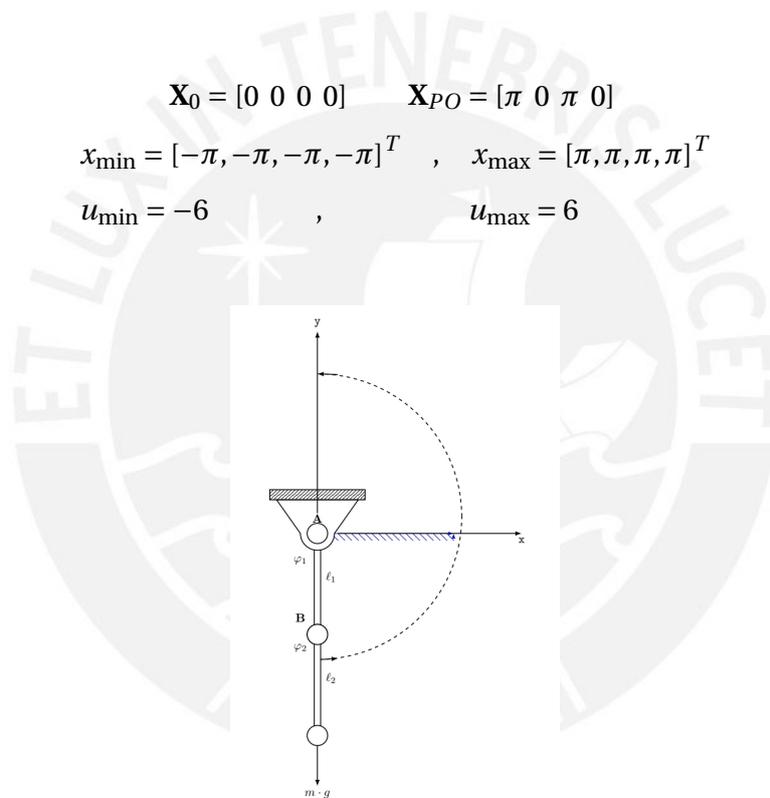


Figura 3.47: Representación de un Sistema de péndulo doble con

$$x_0 = [0 \ 0 \ 0 \ 0] \text{ y } x_{PO} = [\pi \ 0 \ \pi \ 0]$$

Fuente : Elaboración propia

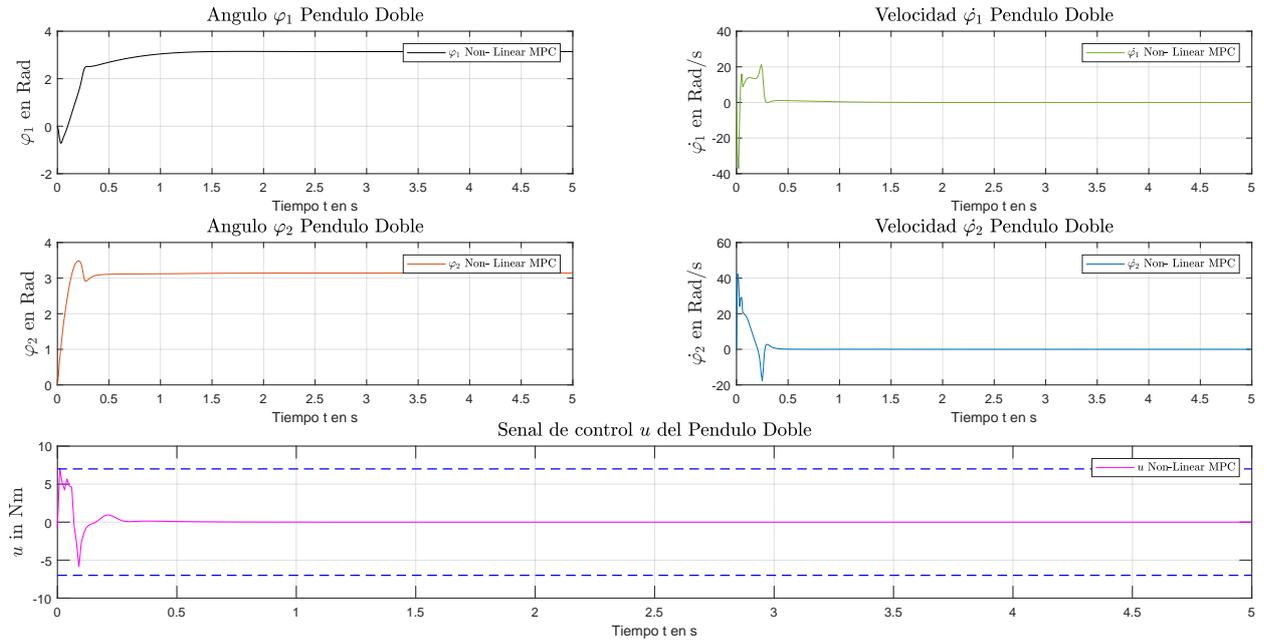


Figura 3.48: Resultado de simulación del MPC No-lineal (NMPC) con

$$x_O = [0 \ 0 \ 0 \ 0] \text{ y } x_{PO} = [\pi \ 0 \ \pi \ 0]$$

Fuente : Elaboración propia

Luego cambiamos a puntos de operación interesantes ya que las condiciones iniciales se diferencian una de la otra como se pueden apreciar en la figura 3.49:

$$\begin{aligned}
 \mathbf{X}_0 &= \left[\frac{\pi}{2} \ 0 \ \pi \ 0 \right] & \mathbf{X}_{PO} &= [\pi \ 0 \ \pi \ 0] \\
 x_{\min} &= [-\pi, -\pi, -\pi, -\pi]^T, & x_{\max} &= [\pi, \pi, \pi, \pi]^T \\
 u_{\min} &= -8, & u_{\max} &= 8
 \end{aligned} \tag{3.72}$$

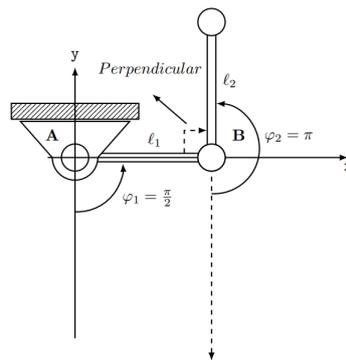


Figura 3.49: Representación de un *Sistema de péndulo doble* con $x_O = \left[\frac{\pi}{2} \ 0 \ \pi \ 0 \right]$ y $x_{PO} = \left[\pi \ 0 \ \pi \ 0 \right]$

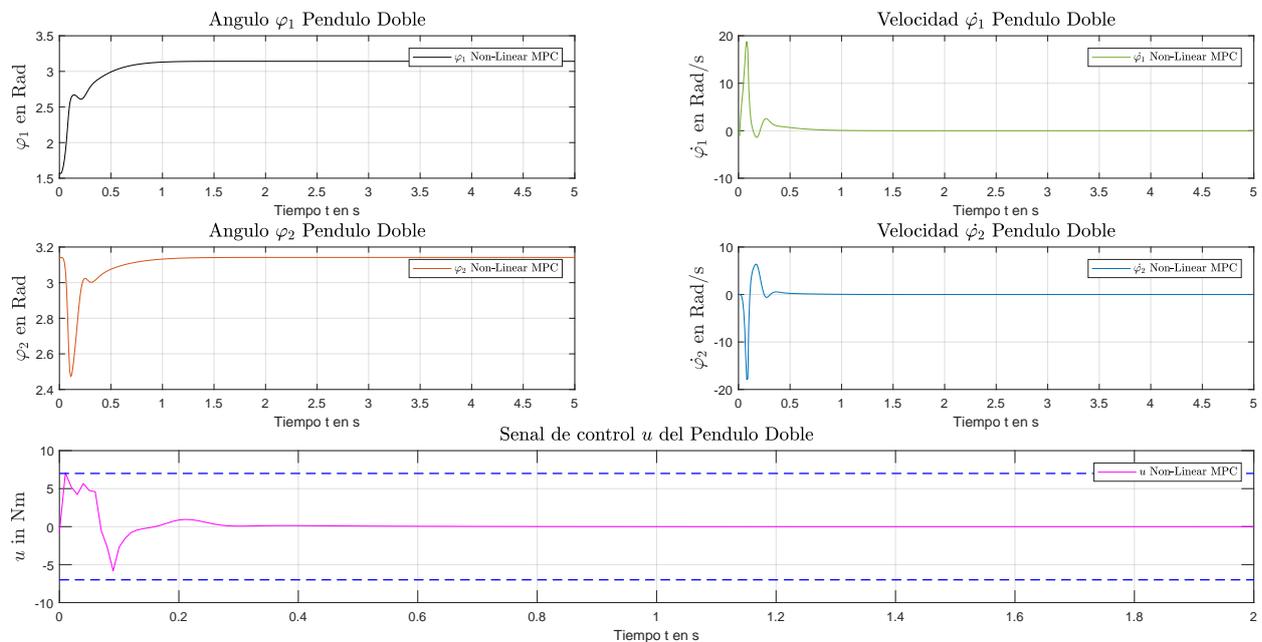


Figura 3.50: Resultado de simulación del MPC No-lineal (NMPC) con $x_O = \left[\frac{\pi}{2} \ 0 \ \pi \ 0 \right]$ y $x_{PO} = \left[\pi \ 0 \ \pi \ 0 \right]$
Fuente : Elaboración propia

Los resultados muestran en la 3.50 que el primero ángulo φ_1 converge a su punto de operación a los 1.67 segundos y el segundo ángulo φ_2 a los 1.73 segundos. Las restricciones fueron de -8 Nm a 8 Nm y el tiempo de computo CPUTime promedio fue de 205.9 μ s.

Por último, cabe una ultima prueba al NMPC basado en ACADO por eso

elegimos puntos diferentes iniciales y puntos finales diferentes a π :

$$\begin{aligned}
 \mathbf{X}_0 &= \left[\frac{\pi}{4} \ 0 \ \pi \ 0 \right] & \mathbf{X}_{PO} &= \left[\frac{\pi}{1.25} \ 0 \ \pi \ 0 \right] \\
 x_{\min} &= [-\pi, -\pi, -\pi, -\pi]^T, & x_{\max} &= [\pi, \pi, \pi, \pi]^T \\
 u_{\min} &= -9, & u_{\max} &= 9
 \end{aligned} \tag{3.73}$$

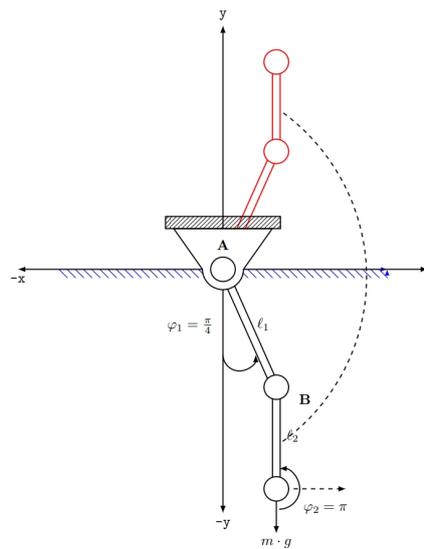


Figura 3.51: Representación de un *Sistema de péndulo doble* con $x_O = \left[\frac{\pi}{4} \ 0 \ \pi \ 0 \right]$ y $x_{PO} = \left[\frac{\pi}{1.25} \ 0 \ \pi \ 0 \right]$
Fuente : Elaboración propia

Finalmente los resultados muestran en la 3.52 que el primero ángulo φ_1 converge a su punto de operación a los 1.86 segundos y el segundo ángulo φ_2 a los 2 segundos. Las restricciones fueron de -6 Nm a 6 Nm y el tiempo de computo CPUTime promedio fue de 232.7 μ s.

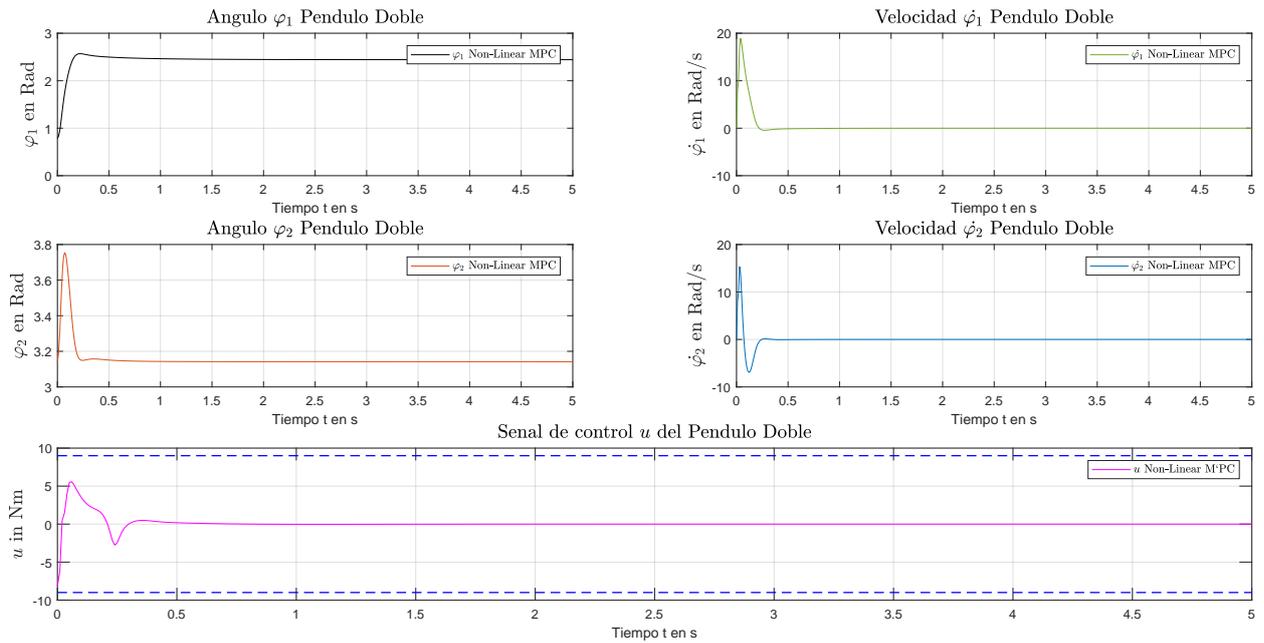


Figura 3.52: Resultado de simulación del MPC No-lineal (NMPC) con

$$x_O = \left[\frac{\pi}{4} \ 0 \ \pi \ 0 \right] \text{ y } x_{PO} = \left[\frac{\pi}{1.25} \ 0 \ \pi \ 0 \right]$$

Fuente : Elaboración propia

Como se pudo observar el NMPC basado en ACADO para diferentes puntos de operación y condiciones iniciales ha respondido bastante bien. Un hecho importante de este controlador es que maneja un modelo netamente no lineal e ideal para el *Sistema de péndulo doble*.

Capítulo 4

CONCLUSIONES Y OBSERVACIONES

Esta tesis se enfoca en diseñar un control para el *Sistema de péndulo doble*. Para eso, se modela matemáticamente *Sistema de péndulo doble* a fin de evaluar cuál controlador ejerce una mejor estrategia de control. De todos los controladores usados, vimos que el NMPC es aquel que presenta una mejor performance con resultados prometedores a pesar incluso de incertidumbres en la entrada control. Si bien es cierto que los otros controladores mostraron de igual forma buenos resultados; sin embargo, su rango de performance se reduce al punto de operación cosa que el NMPC no adolece.

Consiguientemente se concluyó que un modelo no lineal si bien es complicado de resolver, el *solver* ACADO tiene una herramienta de cálculo para las ODEs basado en el método de Runge-Kutta implícito a partir de quinto orden, que según los resultados obtenidos, presentan una convergencia aceptable y no existe el problema de rigidez (Stiff problem) que suele aparecer para los métodos de integración sobre sistemas no lineales. Queda como un trabajo futuro también de ser el caso probar diferentes ordenes y otro tipo de método de discretización dentro del espectro del programación cuadrática secuencial (SQP).

Por otro lado se concluye que si bien, los modelos lineales en la práctica pueden ser usados en la industria manera restringida, en el mundo real se debe considerar un modelo no linealizado, de modo que así, se tiene un espectro más amplio del comportamiento del *Sistema de péndulo doble*. Por lo tanto, se deben definir las ecuaciones diferenciales no lineales para una descripción completa de la dinámica real del sistema y no un *Sistema de péndulo doble* linealizado alrededor de un punto específico de operación. Por eso razón se eligió a lo largo de la solución del problema óptimo de control (OCP) por medio de ACADO, con un tiempo de muestro de valor de 0.01 segundos como el de mejor performance cumpliendo el teorema de Nyquist y Shannon junto con el números de pasos o horizonte de predicción que fue de $N=30$. Precisamente, estos 2 parámetros antes mencionados se recomiendan para usarlos en la configuración del *solver* ACADO en el sistema no lineal. cabe señalar que esta conclusion se llevo a cabo mediante las pruebas del algoritmo hasta que se

lograron muy buenos resultados.

Además se probó la robustez del controlador NMPC y se concluyó que este controlador no lineal es efectivo en contra las posibles incertidumbres. Lo primero que obtuvimos es que el modelo no es perfecto y en realidad no se lograría tener un modelo que represente el 100 por ciento del comportamiento real del *Sistema de péndulo doble*. Pero aún, así el controlador NMPC podría ser usado en la práctica y en la industria. Ahora bien, se tiene que dejar bastante claro que la incertidumbre que se agregó fue aditiva a modo de ejemplo. Existen incertidumbres paramétricas que no fueron parte de esta tesis y debería ser materia de investigación si fuera necesario llevarlas a cabo en base a esta presente tesis.

Se concluye claramente que ACADO como *solver*, se presenta como la mejor la elección para resolver el problema óptimo de control. La razón principal como se vio en la tesis es que con él se obtiene la mejor performance para resolver la OCP usando un modelo no lineal debido a que contempla el comportamiento real del *Sistema de péndulo doble*. Además, el *solver* es de obtención libre y los parámetros del algoritmo pueden modificarse a criterio como se vio en la tesis entre ellas la definición del sistema no lineal por medio de ODEs, la descripción de las trayectorias que queremos que el péndulo siga, el cambio de números de pasos, la definición de tiempo de muestro, las restricciones en las variables de estado y entradas de control, entre otras cosas. Existen otros *solvers* que podrían ser también usados pero según la literatura para este tipo de sistema que estudia esta tesis, ACADO es una herramienta que se adecua perfectamente al *Sistema de péndulo doble* ya que a diferencia de otros *solvers* no se emplean más de 4 variables de estado, una entrada y una salida los cuales forman parte de nuestro sistema, usa el método múltiple Shooting y aproximación Newton-Gauss, usa el término final en la función de costo que asegura la estabilidad y convergencia del sistema no lineal, y que es particular del *solver* ACADO.

Por último, si bien el tiempo de cómputo y los requerimientos de software/hardware sí son importantes. Sin embargo, el NMPC basado en el *solver* ACADO optimiza el tiempo de cómputo ya que encuentra el mínimo óptimo después de la llegada a la cantidad de pasos definidos. Y no requiere un uso de un CPU de alta gama. En efecto, el computador usado, Laptop HP con 1.80 GHz, procesador Intel Core i5-8250U y 8GB de RAM, fue más que suficiente para alcanzar buenos resultados. Por estas razones expuesta se concluye final-

mente que el NMPC basado en ACADO es la mejor opción para la solución del problema óptimo de control como estrategia de control para nuestro *Sistema de péndulo doble*.



Capítulo 5

RECOMENDACIONES Y PERSPECTIVAS

Como se vio en la presente tesis, se modeló el *Sistema de péndulo doble* siguiendo las ecuaciones de LAGRANGE que describen la energía cinética y potencial que gobiernan al *Sistema de péndulo doble*. Desde ese punto se obtienen las ecuaciones de movimiento con respecto a la posición φ_1 y φ_2 , y la velocidad de los péndulos $\dot{\varphi}_1$ y $\dot{\varphi}_2$. En base a estas ecuaciones se prueba el modelamiento matemático en MATLAB/SIMULINK con las funciones Fcn Blocks, Matlab Function block y S function. Los resultados de simulación sobre el modelo matemático obtenido en MATLAB se notan muy prometedores para diferentes entradas como la de una excitación nula y/o excitación sinusoidal. Como se mencionó en el capítulo 2, muchos sistemas no lineales pueden ser descritos de modo lineal con un rango limitado de acción, el llamado punto de operación. Era de forma más rápida y sencilla poder aproximar los términos no lineales como los senos y cosenos cuando los ángulos fueran pequeños pero a fin de evitar imprecisiones en el modelo, se procedió a linealizar el sistema no lineal en diferentes puntos de operación con el objetivo de obtener el sistema en espacio de estados mediante ecuaciones diferenciales de primer orden. Luego, mediante la expansión de la serie de Taylor en los diferentes puntos de operación se obtienen las matrices A , B , C y D en forma analítica. Ahora bien, se realizó posteriormente la linealización en MATLAB en diferentes puntos de operación de modo que obtengamos el sistema en espacio de estados. A partir de ese punto se pudieron calcular los valores propios de cada sistema que mostraron cual de esos sistemas linealizados según dado un punto de operación sea estable. A modo de prueba se consideró 3 puntos de operación con los cuales se procedió a probar la controlabilidad y observabilidad del sistema.

Estos términos a decir controlabilidad y observabilidad nos daría una vista extra sobre que puntos de operaciones estarían aptos a realizar un control y observabilidad en caso de crear un sensor artificial de modo que todas las variables aun si físicamente no sea posible. Los criterios que se usaron fueron el de Kalman, Gilbert y el de Hautus. Para lograrlo se usaron las funciones de MATLAB que ofrecen el estudio de los criterios. Como resultado se demostró

que los 2 primeros puntos de operaciones garantizan la controlabilidad y observabilidad del sistema compatible claramente con la física que gobierna los sistemas de péndulos doble. Una vez obtenida la información sobre los sistemas se procedió a implementar los controladores. El primer controlador usado fue el LQR (Controlador Lineal Cuadrático) cuyo objetivo es minimizar y optimizar la función de costo. La ley de control del LQR logra la solución del problema de optimización. Consiguientemente, se realizó el LQR en MATLAB Y SIMULINK donde se los valores de los ángulos convergieron en los valores deseados en un tiempo esperado. Sin embargo, estos resultados vimos que los resultados podrían mejorarse. Para eso se las matrices de ponderación Q y R podrían mejorar o empeorar derivando a errores numéricos dado que la ley de control K funciona alrededor del punto de operación. Dado los resultados, se concluyó que valores grandes en la Matriz Q produce errores numéricos ya que se presentaron valores grandes en la parte imaginaria en los valores propios. Se asume para el diseño de un controlador, que todas las variables puedan ser medidas por lo general por un controlador. Sin embargo, en la práctica esto no es siempre así ya que la medición pueda que no sea posible, ya sea por el costo o que físicamente sea imposible. Por esa razón, el diseño de Observador de estado con controlador sería nuestro segundo controlador para el *Sistema de péndulo doble* estimando las variables no medibles. Para ello se utiliza el denominado Observador Luenberger, cuya realimentación depende de la diferencia entre el valor real de salida y y el valor estimado \hat{y} . Adicionalmente, incluimos un controlador de estado que haga que el sistema llegue a los valores de referencia esperados. Se demostró posteriormente la Teoría de Separación donde muestran que los valores propios del sistema controlado y sistema observado no influyen una de la otra. De modo que su cálculo de los mismos puedan ser hallados independientemente. Según se vio en la parte del teorema, la mejor performance del observador y control de estado se da, cuando los valores propios del observador sean 10 veces los valores propios del controlador de estado. Es decir cuando los polos del observador sean mas rápidos. Para probar esta teoría se diseña el observador de estado con controlador en MATLAB Y SIMULINK. Para eso, se calculan los valores propios del sistema a lazo cerrado. Posteriormente se multiplican 1, 2.5 y 10 veces, cuyos resultados serán tomados como los valores propios del observador. Los resultados fueron interesantes ya que se comprobó que si los valores propios son mas pequeños que los de los polos a lazo cerrado, obteníamos una mejor performance. Y efectivamente ese fue el caso, sin embargo, el comportamiento del sistema

podría mejorarse ya que no es necesario observar todas las variables sino únicamente las que no se pueden medir. Para eso se identifica las variables medibles y se toman como salida. De allí se pueden calcular las variables faltantes sin un problema numérico considerable. Se debe tener en cuenta que el *Sistema de péndulo doble* lineal, solo tiene un punto de operación óptimo por lo que el valor de la ley de control tiene buen resultado alrededor del mismo. Al igual que el observador completo, se diseña según la estructura del observador Luenberger reducido y se muestra en la figuras los resultados, donde se puede apreciar claramente una mejor performance como antes se había predicho.

El controlador LQR y Observador con control de estado presentaron limitaciones ya que los sistemas en la practica son no lineales, multivariabes y con restricciones. Estas características antes mencionadas hacen que los controladores tradicionales no logren satisfacerlas. Por eso, se diseñó un controlador Lineal Predictivo Basado en Modelo (MPC Lineal). Como se vio en la parte teórica, el MPC hace que los péndulos lleguen en forma óptima y eficiente con el menor gasto de energía. Para eso, se resuelve el problema óptimo de control en donde a fin de predecir el estado siguiente, se toma el valor de la variable de estado anterior y con este se calcula la señal óptima de control para predecir el siguiente valor de estado. Con estos valores se procede a volver a obtener la nueva variable de estado y esta se convierte en nuestro nuevo valor inicial y se reinicia el proceso. Esto se realiza según la cantidad de pasos establecidas llamadas horizonte de predicción durante el tiempo de simulación. Al tener un forma de retroalimentación los valores finales convergen a los puntos deseados ya que se corrige a lo largo del tiempo. Posteriormente se lleva el diseño al MATLAB; usaremos por lo tanto el comando `quadprog`. Para ello se tuvo que llevar el sistema de espacio de estado continuo a discreto a fin de hacer uso del comando. Esta discretización se hizo también por medio de MATLAB y es muy probable que tenga errores de aproximación. El comando requiere que se construyan las matrices que forman parte del comando. Por lo tanto se resuelve el problema de control con el MPC.

Los resultados se muestran mas rápidos a diferencia de los controladores anteriores, además que el tiempo de convergencia es en su estado de equilibrio es bastante rápido también. Sin embargo, se muestra claramente una oscilación hasta el segundo 2.3 que es probablemente producto de las matrices de ponderación. Debemos acordarnos que el rango de punto de operación es muy pequeño y añadido a su error de discretización. Se podría mejorar la respuesta

en cuanto a la discretización del sistema considerando delays que muestren realmente el sistema en discreto. Sin embargo, los resultados son aceptables y sobre todo mejorable por esa razón se prueba el MPC no lineal para este problema óptimo de control. Como se mencionó en la capítulo 4 sobre los resultados del comando QUADPROG se podría mejorar usando un NMPC para el seguimiento de referencias que considera cambios en los mismos y que el comando no lo considera, es claro que esta técnica forma parte de un estudio en el futuro para el mejoramiento de un MPC lineal para un sistema lineal. Por lo tanto, es claro que la mayor concentración de esta tesis es en lo que respecta al diseño e implementación del MPC no lineal sino del NMPC.

Las limitaciones del MPC lineal debido al uso de un sistema discreto hace que no sea una estrategia de control adecuada para la solución del problema de control. Por esa razón el NMPC basado en ACADO llega a ser adecuado porque se resuelve las ODEs que previamente se hallaron en la modelización matemática del sistema. Las ecuaciones de movimiento ya no se deberán linealizar y debido a ACADO pueden resolverse mediante el método de Runge-Kutta implícito.

Para hacer el uso de ACADO se deberá instalar el paquete libre en MATLAB y asimismo el compilador de Visual C++. Los resultados del MPC no lineal basado en ACADO muestran resultados bastante buenos en comparación con los otros controladores. Y aún mejor incluso que el MPC lineal ya que no había necesidad de linear el sistema como con los otros controladores. Después se probó la robustez del NMPC sobre posible incertidumbres tanto en el modelo. Los resultados del NMPC frente a perturbaciones dan buenos resultados. Las perturbaciones que se uso fueron ruido blanco Gaussiano de -6 dB y un ruido aleatorio que se introduce adicionalmente a la entrada de control. Donde concluimos que el ruido blanco Gaussiano de -6dB estadísticamente es constante por lo que el NMPC aprende esta perturbación más rápidamente y reacciona de mejor manera que el del caso de la perturbación aleatoria ya que estos son aleatorios e impredecibles. Sin embargo, para ambos casos el controlador reacciona con buenos resultados solo que uno mas rápido que el otro. Para ambos casos se muestra además, el tiempo de cómputo (CPU Time) para todas las simulaciones considerando el procesador y velocidad RAM del equipo que se utilizó. Cabe resaltar de este punto que el CPU Time más rápido fue el del ruido blanco Gaussiano en comparación a la perturbación aleatoria. Queda en trabajos futuros probar perturbaciones en las variables de estado a fin de

probar la robustez del NMPC.

Finalmente, se prueban todos los controladores en un mismo punto de operación. De modo que todos tengan los mismo parámetros a ser evaluados. De los resultados se nota claramente que el NMPC tiene una mejor respuesta frente al *Sistema de péndulo doble*. Por esa razón, se probaron otros puntos de operación hasta incluso otros puntos de referencia o trayectorias, dando muy buenos resultados. Sin embargo, aunque no se tienen restricciones tan marcadas que harían al NMPC y el sistema más restrictivo son resultados bastantes importantes para concluir que el NMPC soluciona mejor la OCP; sin embargo, podría ser materia de un próximo trabajo restringir mucho más las variables de estado y entrada de control.

Se recomienda por lo tanto el uso del NMPC para sistemas no lineales tales como el *Sistema de péndulo doble*. Y se vaya aun mas allá ya que se recomienda encarecidamente aplicar esta solución a proyectos que utilicen este sistema no lineal tales como los que se vieron en la recopilación de trabajos que usan este tipo de sistema en el capítulo 1. Por otro lado, se recomienda a los autores de trabajos antes mencionados y que usaron un sistema linealizado del *Sistema de péndulo doble*, puedan probar esta solución no lineal.



BIBLIOGRAFÍA

- [1] Ulrich Konigorski. *Praktikum matlab®/simulink® ii. Skript und Vorbereitungsaufgaben*, Darmstadt, 2017.
- [2] G Rigatos. Nonlinear optimal control for the underactuated double-pendulum overhead crane. *Journal of Vibration Engineering & Technologies*, páginas 1–21, 2023.
- [3] Yijiang Zhao, Xianqing Wu, Fang Li, y Yibo Zhang. Positioning and swing elimination control of the overhead crane system with double-pendulum dynamics. *Journal of Vibration Engineering & Technologies*, páginas 1–8, 2023.
- [4] Qihang Guo, Lin Chai, y Huikang Liu. Anti-swing sliding mode control of three-dimensional double pendulum overhead cranes based on extended state observer. *Nonlinear Dynamics*, 111(1):391–410, 2023.
- [5] Bruno Strah. *Regelung eines Radbasierten Inversen doppelpendels Unter Betrachtung Hybrider Dynamik*. PhD thesis, Jul 2012. URL <https://tuprints.ulb.tu-darmstadt.de/3050/>.
- [6] Y. Bazargan Lari, A. Gholipour, M. Eghtesad, M. Nouri, y A. Sayadkooh. Dynamics and control of locomotion of one leg walking as self-impact double pendulum. In *The 2nd International Conference on Control, Instrumentation and Automation*, páginas 201–206, 2011. doi: 10.1109/ICCIAutom.2011.6356656.
- [7] Amanda Young, Chengyu Cao, Naira Hovakimyan, y Eugene Lavretsky. Control of a nonaffine double-pendulum system via dynamic inversion and time-scale separation. In *2006 American Control Conference*, páginas 6–pp. IEEE, 2006.
- [8] Naira Hovakimyan, Eugene Lavretsky, y Chengyu Cao. Dynamic inversion of multi-input nonaffine systems via time-scale separation. In *2006 American Control Conference*, páginas 6–pp. IEEE, 2006.
- [9] EK Lavrovskii y AM Formal'skii. The optimal control synthesis of the swinging and damping of a double pendulum. *Journal of Applied Mathematics and Mechanics*, 65(2):219–227, 2001.
- [10] AM Formal'skii. On stabilization of an inverted double pendulum with one control torque. *Journal of Computer and Systems Sciences International*, 45(3):337–344, 2006.
- [11] Oliver Urbann. *Laufen humanoider roboter auf regelungstechnischer basis MIT echtzeitmodifikation Der Fußpositionen*. PhD thesis,

2017. URL https://eldorado.tu-dortmund.de/bitstream/2003/35831/1/Dissertation_Urbann.pdf.
- [12] Katsuhiko Ogata et al. *Modern control engineering*, tomo 5. Prentice hall Upper Saddle River, NJ, 2010.
- [13] William L Luyben. Design and control degrees of freedom. *Industrial & engineering chemistry research*, 35(7):2204–2214, 1996.
- [14] Akira Ohsumi y Takeya Izumikawa. Nonlinear control of swing-up and stabilization of an inverted pendulum. In *Proceedings of 1995 34th IEEE Conference on Decision and Control*, tomo 4, páginas 3873–3880. IEEE, 1995.
- [15] Yingliang Bai, Zhigang Chen, y Xiaogang Ruan. Complex trajectory tracking of inverted pendulum using pid and lqr. In *International conference on manufacturing science and information engineering (ICMSIE)*, páginas 516–524, 2017.
- [16] Boran Yesilyurt. Equations of motion formulation of a pendulum containing n-point masses, Feb 2020. URL <https://arxiv.org/abs/1910.12610>.
- [17] Heston Neal. Using mechanics of a double pendulum to maximize sport performance, 2020. URL <https://digitalcommons.coastal.edu/bridges/vol13/iss13/5/>.
- [18] Alexander Bogdanov. Optimal control of a double inverted pendulum on a cart, Dec 2004. URL <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.78.6105&rep=rep1&type=pdf>.
- [19] Mahdieh Adeli, Seyedeh Hamideh Zarabadi, Hassan Zarabadipour, y Mahdi Aliyari Shoorehdeli. Design of a parallel distributed fuzzy lqr controller for double-pendulum-type overhead cranes. *2011 IEEE International Conference on Control System, Computing and Engineering*, 2011. doi: 10.1109/iccsce.2011.6190497.
- [20] Gildeberto S Cardoso y Leizer Schnitman. Analysis of exact linearization and approximate feedback linearization techniques. *Mathematical Problems in Engineering*, 2011, 2011.
- [21] Katsuhiko Ogata. *Discrete-time control systems*. Prentice-Hall, Inc., 1995.
- [22] Lars Grüne y Jürgen Pannek. Nonlinear model predictive control. In *Nonlinear model predictive control*, páginas 45–69. Springer, 2017.
- [23] Lars Grüne. Approximation properties of receding horizon optimal control. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 118(1):3–37, 2016.

- [24] Jürgen adamy. *Systemdynamik und Regelungstechnik II*. Shake Verlag, Aachen, 5 ed^{ón}, 2016. ISBN 9783844043464.
- [25] Carsten Timm. Theoretische mechanik. Skript, Institut für Theoretische Physik, TU Dresden, 2021.
- [26] Helmut Scherf. *Modellbildung und Simulation dynamischer Systeme*. Oldenbourg Verlag München, Rosenheimer Strasse 145, D-81671 München, 4 ed^{ón}, 2010. ISBN 9783486596557.
- [27] Karl Finckenstein, Lehn Jurgen, y Helmut Schellhaas. *Arbeitsbuch Mathematik für ingenieure: Band II: Differentialgleichungen, Funktionentheorie, Numerik und Statistik*. Vieweg+Teubner Verlag, 2013.
- [28] Jan Lunze. *Steuerbarkeit und Beobachtbarkeit*, páginas 51–114. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002. ISBN 978-3-662-09725-0. doi: 10.1007/978-3-662-09725-0_3. URL https://doi.org/10.1007/978-3-662-09725-0_3.
- [29] Ulrich Konigorski. Mehrgrößenreglerentwurf im zustandsraum. Skript, Darmstadt, 2018.
- [30] Jan Lunze. *Regelungstechnik 2*, tomo 6. Springer Berlin Heidelberg", 2016. ISBN 978-3-540-61404-3. doi: 10.1007/978-3-662-52676-7_7.
- [31] Huibert Kwakernaak y Raphael Sivan. *Linear optimal control systems*, tomo 1072. Wiley-interscience, 1969.
- [32] R. M. Murray. Cds 110b lecture 2 – lqr control. Lecture, Control and Dynamical Systems, CALTECH, 2006. URL <http://www.cds.caltech.edu/~murray/courses/cds110/wi06/lqr.pdf>.
- [33] Roland Büchi. *State space control, LQR and observer: step by step introduction with Matlab examples*. Norderstedt Books on Demand, 2010. ISBN 978-3-8370-2016-8.
- [34] R. Rivas-Perez, J. Sotomayor-Moriano, G. Perez-Zuñiga, y M.E. Soto-Angles. Real-time implementation of an expert model predictive controller in a pilot-scale reverse osmosis plant for brackish and seawater desalination. *Appl. Sci.*, 9(14):2932, 2019.
- [35] R. Rivas-Perez, J. Sotomayor-Moriano, y C.G. Perez-Zuñiga. Adaptive expert generalized predictive multivariable control of seawater ro desalination plant for a mineral processing facility. *IFAC- PapersOnLine*, 50(1): 10244–10249, 2017.
- [36] Eduardo F Camacho y Carlos Bordons Alba. *Model predictive control*. Springer science & business media, 2003. ISBN 1-85233-694-3.

- [37] R. Rivas-Perez, J. Sotomayor-Moriano, C.G. Perez-Zuñiga, y E.M. Calderon-Mendoza. Design of a multivariable gpc based on an industrial pc for control of a reverse osmosis unit of a pharmaceutical industry. *Rev. Mex. Ing. Quim.*, 15(1):295–273, 2016.
- [38] William S Levine. The essentials of model predictive control. In *Handbook of model predictive control*, páginas 3–27. Springer, 2019.
- [39] Francesco Borrelli, Alberto Bemporad, y Manfred Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017. ISBN 9781107016880.
- [40] Rafal Goebel y Saša V Raković. Set-valued and lyapunov methods for mpc. In *Handbook of Model Predictive Control*, páginas 53–73. Springer, 2019.
- [41] Ricardo G Sanfelice. Hybrid model predictive control. In *Handbook of Model Predictive Control*, páginas 199–220. Springer, 2019.
- [42] Wook Hyun Kwon y Soo Hee Han. *Receding horizon control: model predictive control for state models*. Springer Science & Business Media, 2006.
- [43] R. Findeosen. Lecture 07, stability of model predictive control. Lecture, Optimal and Predictive Control, TU Darmstadt, 2021.
- [44] Jan Marian Maciejowski. *Predictive control: with constraints*. Pearson education, 2002.
- [45] James Blake Rawlings, David Q Mayne, y Moritz Diehl. *Model predictive control: theory, computation, and design*, tomo 2. Nob Hill Publishing Madison, 2017.
- [46] J.W. Vásquez, G. Perez-Zuñiga, Y. Muñoz, y A. Ospino. Simultaneous occurrences and false-positives analysis in discrete event dynamic systems. *J. Comput. Sci*, 44:101162, 2020.
- [47] *MATLAB version 9.3.0.713579 (R2017b)*. The Mathworks, Inc., Natick, Massachusetts, 2017. URL <https://de.mathworks.com/help/ident/ref/lti.c2d.html>.
- [48] Otto Föllinger. Lineare abtastsysteme. In *Lineare Abtastsysteme*. Oldenbourg Wissenschaftsverlag, 5. durchgesehene auflage ed^{ón}, 1993.
- [49] H. CHEN y F. ALLGÖWER. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10): 1205–1217, 1998. ISSN 0005-1098. doi: [https://doi.org/10.1016/S0005-1098\(98\)00073-9](https://doi.org/10.1016/S0005-1098(98)00073-9).
- [50] Matthew Brand, Vijay Shilpiekandula, Chen Yao, Scott A Bortoff, Takehiro Nishiyama, Shoji Yoshikawa, y Takashi Iwasaki. A parallel quadratic programming algorithm for model predictive control. *IFAC Proceedings Volumes*, 44(1):1031–1039, 2011.

- [51] A.S. Huaman Loayza y C.G. Pérez Zuñiga. Design of a fuzzy sliding mode controller for the autonomous path-following of a quadrotor. *IEEE Lat. Am. Trans.*, 17:962–971, 2019.
- [52] Johan Åkesson. *MPCtools 1.0: Reference Manual*. Department of Automatic Control, Lund University, 2006.
- [53] Roshan Sharma. Lecture notes for the course iia 4117: Model predictive control, 2019.
- [54] D Limon, I Alvarado, T Alamo, y EF Camacho. Mpc for tracking of piecewise constant references for constrained linear systems. *IFAC Proceedings Volumes*, 38(1):135–140, 2005.
- [55] Linfeng Gou, Xianyi Zeng, Zhaohui Wang, Guangjie Han, Chuan Lin, y Xu Cheng. A linearization model of turbofan engine for intelligent analysis towards industrial internet of things. *IEEE Access*, 7:145313–145323, 2019.
- [56] Vojtech Veselý y Adrian Ilka. Gain-scheduled pid controller design. *Journal of process control*, 23(8):1141–1148, 2013.
- [57] David Ariens, B Houska, y Hans Joachim Ferreau. Acado toolkit user’s manual. *Optimization in Engineering Center (OPTEC) and Department of Electrical Engineering, KU Leuven, version, 1*, 2010.
- [58] Rien Quirynen, Milan Vukov, Mario Zanon, y Moritz Diehl. Autogenerating microsecond solvers for nonlinear mpc: a tutorial using acado integrators. *Optimal Control Applications and Methods*, 36(5):685–704, 2015.
- [59] Joel AE Andersson, Joris Gillis, Greg Horn, James B Rawlings, y Moritz Diehl. Casadi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.
- [60] Mark L. Darby. *Industrial MPC of Continuous Processes in: Baillieul J., Samad T. (eds) Encyclopedia of Systems and Control*, páginas 1–8. Springer London, London, 2020. ISBN 978-1-4471-5102-9. doi: 10.1007/978-1-4471-5102-9_242-2. URL https://doi.org/10.1007/978-1-4471-5102-9_242-2.
- [61] B. Houska, H.J. Ferreau, y M. Diehl. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.
- [62] Ernst Hairer y Gerhard Wanner. Stiff differential equations solved by radau methods. *Journal of Computational and Applied Mathematics*, 111(1-2):93–111, 1999.

- [63] Jorge Nocedal y Stephen J Wright. *Numerical optimization (2nd Edn)*. Springer, 2006.
- [64] Andreas Wächter y Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.
- [65] Sakthi Thangavel, Sankaranarayan Subramanian, Radoslav Paulen, y Sebastian Engell. Robust multi-stage nmpc under structural plant-model mismatch without full-state measurements. In *2019 18th European Control Conference (ECC)*, páginas 781–786. IEEE, 2019.
- [66] Gabriele Pannocchia, James B Rawlings, y Stephen J Wright. Inherently robust suboptimal nonlinear mpc: theory and application. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, páginas 3398–3403. IEEE, 2011.
- [67] Jing Zhao, Victor L Knoop, y Meng Wang. Two-dimensional vehicular movement modelling at intersections based on optimal control. *Transportation Research Part B: Methodological*, 138:1–22, 2020.
- [68] A Thamemul Ansari, K Tharani Raja, y H Kala. Assessment of diverse controllers for a cylindrical tank level process. *IJIRST*, 1(6), 2014.

ANEXOS

Implementación y validación del Modelo

Después de la obtención analítica de las ecuaciones de LAGRANGE se procede a la verificación por medio del MATLAB. Se comienza entonces a codificar la parte matemática. Una parte del algoritmo se muestra en el algoritmo 1:

Algoritmo 1: Ecuación de Lagrange

```
1 %% Mecanica de LAGRANGE
2 T = 1/2*m1*(dx1^2+dy1^2) + 1/2*J1*dphi1^2 + ...
    1/2*m2*(dx2^2+dy2^2) + 1/2*J2*dphi2^2;
3 U = m1*g*y1 + m2*g*y2;
4 Qphi1 = M - Mr1;
5 Qphi2 = -Mr2;
6 L = T - U;
7 %% Obtencion de la derivada seg n las coordenadas ...
    generalizadas
8 % dL/dphi1
9 L_phi1 = jacobian(L,phi1);
10 % dL/ddphi1
11 L_dphi1 = jacobian(L,dphi1);
12 % dL/dphi2
13 L_phi2 = jacobian(L,phi2);
14 % dL/ddphi2
15 L_dphi2 = jacobian(L,dphi2);
16 % Variables con t mediante Variables sin t
17 dL_dphi1_t = subs(dL_dphi1_t ,Var_t ,Var_ot);
18 dL_dphi2_t = subs(dL_dphi2_t ,Var_t ,Var_ot);
19 %% Obtencion final de la ecuacion de LAGRANGEschen
20 Sol = solve([dL_dphi1_t - L_phi1 == Qphi1, dL_dphi2_t - ...
    L_phi2 == Qphi2],[ddphi1 ,ddphi2]);
21 output1 = simplify(Sol.ddphi1);
22 output2 = simplify(Sol.ddphi2);
```

Las salidas tanto output1 y output2 corresponden a $\ddot{\phi}_1$ y $\ddot{\phi}_2$ halladas en software.

Modelamiento en MATLAB/SIMULINK

Los modelos que se van a utilizar para la verificación de la parte analítica son:

- Fcn-Blocks
- Matlab-Function-Blocks y
- M-File S-Function

-Fcn-Blocks

Este tipo de bloque tiene como entrada $u(i)$. Para la verificación de las ecuaciones de movimiento serán necesarias 5 entradas y son las siguientes:

Tabla .1: Entradas Fcn-Block

Entradas $u(i)$				
$u(1) = M$	$u(2) = \varphi_1$	$u(3) = \dot{\varphi}_1$	$u(4) = \varphi_2$	$u(5) = \dot{\varphi}_2$

El modelo en MATLAB/SIMULINK se muestra en la figura .1. La entrada $u(1)$ puede ser variada y representa a M . Los valores iniciales se pueden ingresar en el bloque integral $\frac{1}{s}$.

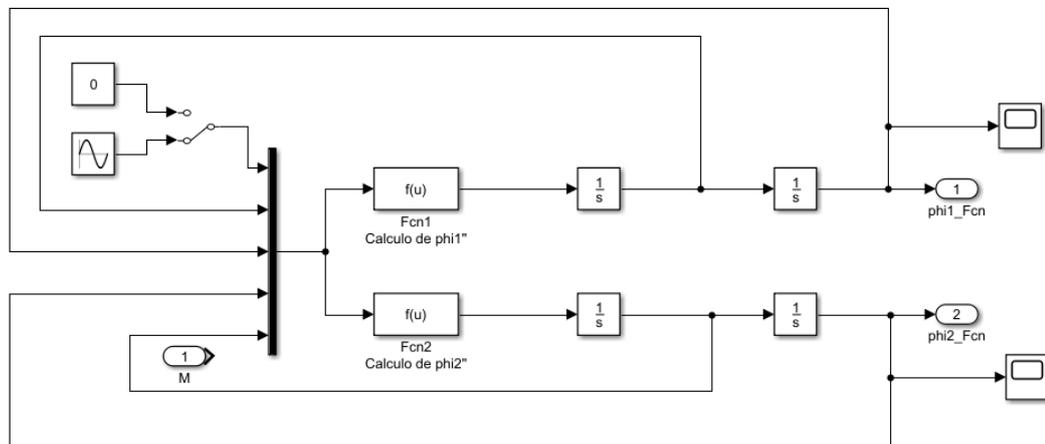


Figura .1: Modelamiento del *Sistema de péndulo doble* con Fcn-Block.

Fuente: Elaboración propia

Si hablamos de las ventajas del modelamiento por medio de este Fcn-Block es que se puede implementar directamente desde las ecuaciones diferenciales obtenidas en la parte analítica; no hay necesidad por lo tanto de transformarlas en Variable de Estado. En cuanto a la desventaja de este Fcn-Block es que no se puede realizar ninguna operación matricial. A continuación se muestran las funciones del modelo, a decir $Fcn1$ y $Fcn2$:

Algoritmo 2: Fcn1 y Fcn2

```

1 output1 =
2 -(6*(4*Rp1*u(2)*l2 - 4*u(1)*l2 + 6*Rp2*u(2)*l1*cos(u(3) ...
   - u(4)) - 6*Rp2*u(5)*l1*cos(u(3) - u(4)) + ...
   (3*g*l1*l2*m2*sin(u(3) - 2*u(4)))/2 + ...
   (3*u(2)^2*l1^2*l2*m2*sin(2*u(3) - 2*u(4)))/2 + ...
   2*u(5)^2*l1*l2^2*m2*sin(u(3) - u(4)) + ...
   2*g*l1*l2*m1*sin(u(3)) + ...
   (5*g*l1*l2*m2*sin(u(3)))/2)/(l1^2*l2*(8*m1 + 15*m2 - ...
   9*m2*cos(2*u(3) - 2*u(4)))) \\
3
4 output2 =
5 (6*(4*Rp2*u(2)*l1*m1 - 6*u(1)*l2*m2*cos(u(3) - u(4)) + ...
   12*Rp2*u(2)*l1*m2 - 4*Rp2*u(5)*l1*m1 - ...
   12*Rp2*u(5)*l1*m2 + 6*u(2)^2*l1^2*l2*m2^2*sin(u(3) - ...
   u(4)) - 3*g*l1*l2*m2^2*sin(u(4)) + ...
   (3*u(5)^2*l1*l2^2*m2^2*sin(2*u(3) - 2*u(4)))/2 + ...
   3*g*l1*l2*m2^2*sin(2*u(3) - u(4)) + ...
   6*Rp1*u(2)*l2*m2*cos(u(3) - u(4)) + ...
   2*u(2)^2*l1^2*l2*m1*m2*sin(u(3) - u(4)) - ...
   (g*l1*l2*m1*m2*sin(u(4)))/2 + ...
   (3*g*l1*l2*m1*m2*sin(2*u(3) - ...
   u(4)))/2)/(l1*l2^2*m2*(8*m1 + 15*m2 - ...
   9*m2*cos(2*u(3) - 2*u(4))))

```

Las funciones *Fcn1* y *Fcn2* contienen a las entradas $u(1)..u(5)$ que reemplazan a las variables mencionadas en la .1. Los parámetros ($g, m1, m2, l_1, l_2, R_{P1}, R_{P2}$) deben ser inicializadas antes de empezar la simulación.

-Matlab-Function-Blocks

A fin de utilizar la Matlab-Function-Block debemos transformar las ecuaciones diferenciales de movimiento a ecuaciones diferenciales de primer orden. La secuencia de las variables se toma el orden de las misma como el de la Fcn-Block y es la siguiente:

Tabla .2: Entradas Matlab-Function-Block

Entradas x(i)			
$x(1) = \varphi_1$	$x(2) = \dot{\varphi}_1$	$x(3) = \varphi_2$	$x(4) = \dot{\varphi}_2$

Como se puede apreciar en la figura .2 a diferencia del modelo con Fcn-Block, se pueden realizar operaciones matriciales.

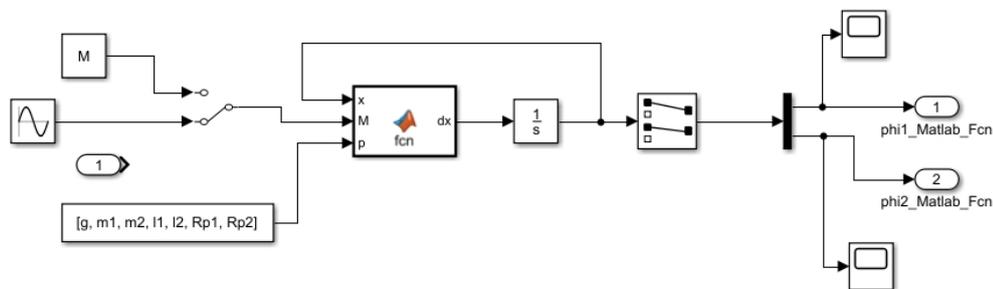


Figura .2: Modelamiento del Sistema de péndulo doble con Matlab Fcn-Block

Fuente: Elaboración propia

-S-Function

Con el modelamiento por medio de S-Function, se puede programar directamente en Matlab M, C; C++, Fortran entre otros. Un punto a tomar a cuenta es que se toma la S-Function Level 2. El modelamiento de S-Function se puede ver en la figura .3:

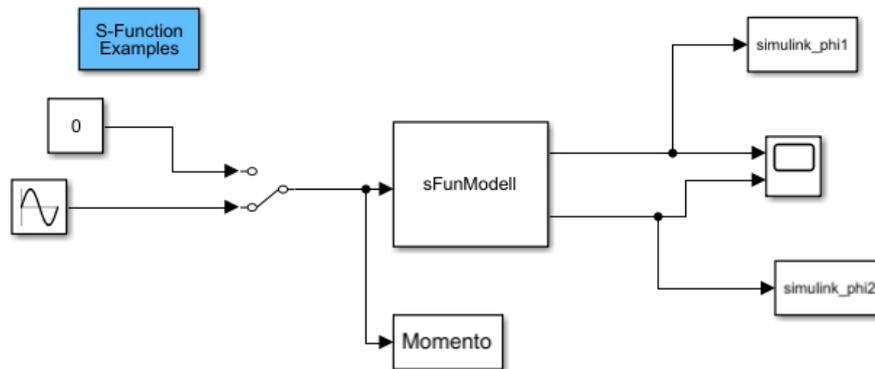


Figura .3: Modelamiento del *Sistema de péndulo doble* con S-Function

Fuente: Elaboración propia

Simulación

- Configuración de parámetros

Para la simulación se usa el Solver ODE45 (Dormand-Prince) esto se puede apreciar en la figura .4 en el "Tab" Solver. Algo importante a notar en la cantidad máxima de pasos (Step Size) (en el tab Max step size). Por lo general, se fija en Auto y entre 0 a 10 segundos se calcularán aproximadamente alrededor de 145 puntos. A fin de mejorar la precisión y aumentar la cantidad de puntos, se deberá reducir la cantidad máxima de pasos a 0.01. Por lo tanto, obtenemos alrededor de 1000 puntos.

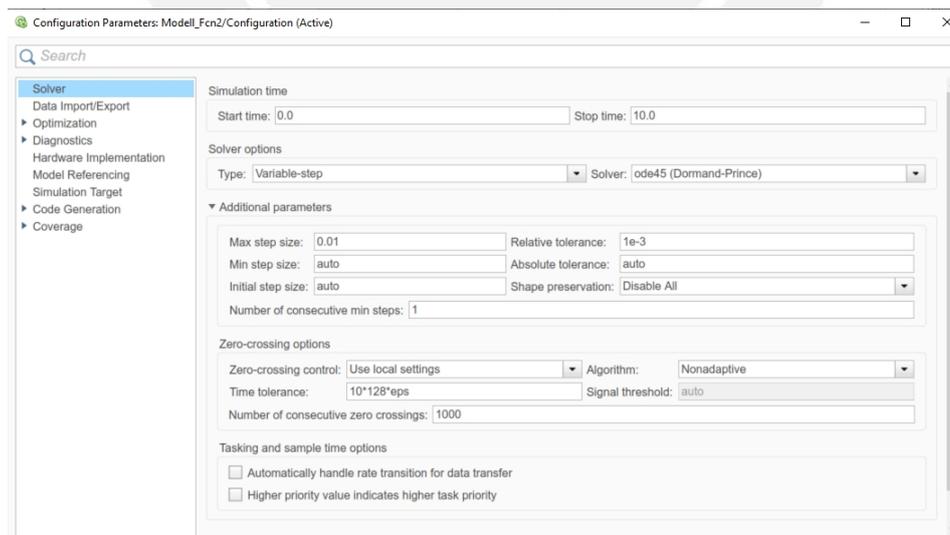


Figura .4: Configuración de parámetros.

Fuente: Elaboración propia

Diseño de un GUI para el Observador y Controlador de Estado

A fin de automatizar y graficar el movimiento de nuestro *Sistema de péndulo doble* ya sea controlado por el controlador de estado junto con el observador se realiza en MATLAB un programa visual donde el usuario puede no solo calcular los valores de la ley de control, sino como los polos del sistema a lazo cerrado. Cabe resaltar que de la misma forma se puede modificar según se necesite las matrices de ponderación Q y R. En el caso del Observador este GUI también contempla los valores propios del observador, los cuales deberán ser 10 veces mas pequeños que los que se obtienen de la parte real de los valores propios del sistema a lazo cerrado. Como se puede apreciar en la parte media se puede modificar también a criterio los puntos de operación de ambos péndulos. Y en el parte inferior izquierda se pueden también modificar los puntos iniciales de todas las variables de posición y velocidad de los péndulos.

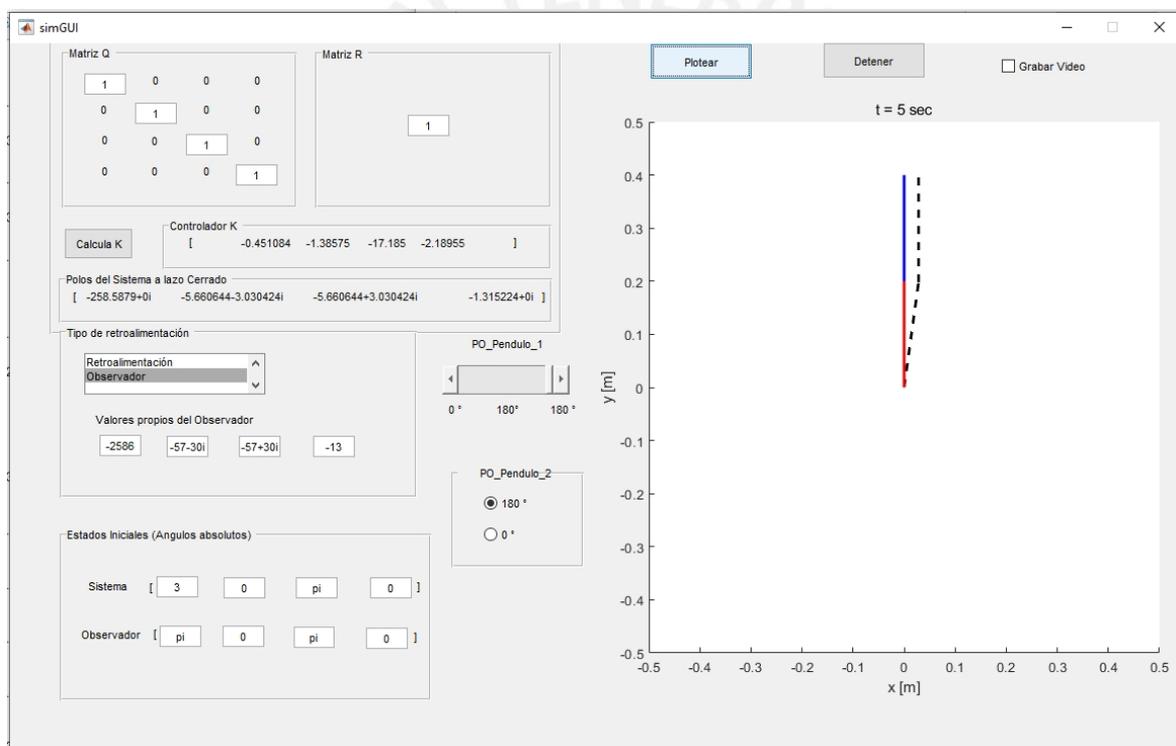


Figura .5: Interfaz gráfica para el Observador y el Sistema controlado

Fuente: Elaboración propia

Una característica visual importante del GUI para el *Sistema de péndulo doble*, es que muestra en las líneas punteadas los puntos iniciales de los péndulos y en colores tanto rojo (péndulo 1) y azul (péndulo 2) mostrarían su posición final. La última característica también importante es que el movimiento del péndulo doble puede ser grabado a fin de analizar el movimiento que se realizó.

Instalación de ACADO en MATLAB

Para empezar a dar detalles sobre el *solver* debemos entender qué es ACADO. En resumen, ACADO es un toolkit escrito en C++ que provee una variedad de algoritmos para control predictivo como el MPC. Los integradores para la simulación como Runge-Kutta para las ODE's y DAE's son muy eficientes. Su carácter *open-source* lo hace manejable a fin de encontrar actualizaciones permanentes y ayuda en línea respecto a diferentes problemas en su implementación [51].

Para la instalación de ACADO en MATLAB se debe instalar la última versión en el siguiente link: <http://www.acadotoolkit.org/download.php> y un compilador reciente de C++. La versión que usaremos para nuestro compilador será MICROSOFT VISUAL C++ 2017.

Para vincular el compilador a MATLAB se usó el siguiente comando:

```
1 mex setup ;
```

MATLAB regresa la siguiente salida:

```
1 Select a compiler:
2 [1] Lcc win32 C 2.4.1 in ...
   C:\PROGRAM 1\MATLAB\R2017b\sys\lcc
3 [2] Microsoft Visual C++ 2017 Express in C:\Program ...
   Files\Microsoft Visual
4 Studio 2017
5 [0] None
6 Compiler:
```

En ese caso elegimos la opción 2 con la confirmación con el mensaje:

```
1 Compiler: Microsoft Visual C++ 2017
2 Location: C:\Program Files\Microsoft Visual Studio 2017
3 Are these correct [y]/n?
```

Escribimos Y y presionamos **enter** para confirmar. Ahora el siguiente paso es construcción de ACADO en MATLAB. Para eso, los archivos que descargamos lo colocamos en el directorio de MATLAB con el siguiente nombre, <ACADOTOOLKIT-INST-DIR> y empleamos el siguiente comando para crear el directorio:

```
1 cd interfaces/matlab/;
```

Luego compilamos ACADO para MATLAB:

```
1 make clean all;
```

Y obtenemos la siguiente respuesta después de compilar los comandos antes mencionados.

```
1 Cleaning up all ACADO files...
2 Removing ACADO folders from Matlab path...
3 Clean completed.
4 Making ACADO...
5 *Building with 'Microsoft Visual C++ 2017'.
6 MEX completed successfully.
```

