

TECNOLÓGICO NACIONAL DE MÉXICO

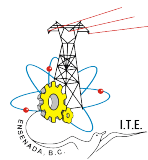
INSTITUTO TECNOLÓGICO DE ENSENADA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

MAESTRÍA EN CIENCIAS EN INGENIERÍA MECATRÓNICA

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO



IMPLEMENTACIÓN NUMÉRICA Y COMPARACIÓN DE MÉTODOS DE
CONTROL ÓPTIMO

*Trabajo de tesis presentado por HEDY CÉSAR RAMÍREZ DE ÁVILA
para obtener el grado de MAESTRO EN CIENCIAS EN INGENIERÍA MECATRÓNICA*

*Bajo la dirección de JUAN ANTONIO ROJAS QUINTERO
y la co-dirección de EUSEBIO BUGARIN CARLOS*

ENSENADA, BAJA CALIFORNIA, MÉXICO, DICIEMBRE 2021



Instituto Tecnológico de Ensenada
División de Estudios de Posgrado e Investigación

Ensenada, Baja California, **15/Diciembre/2021**

No. Oficio. D.E.P.I/18/2021

Asunto: Autorización de impresión de Trabajo de Tesis

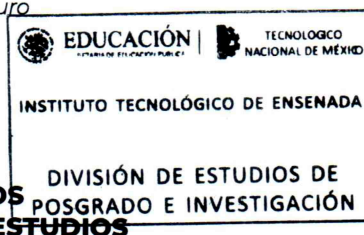
C. HEDY CÉSAR RAMÍREZ DE ÁVILA
ESTUDIANTE DE LA MAESTRÍA EN CIENCIAS
EN INGENIERÍA MECATRÓNICA
PRESENTE

Con base en el dictamen de aprobación emitido por el Comité Tutorial de la Tesis con título: IMPLEMENTACIÓN NUMÉRICA Y COMPARACIÓN DE MÉTODOS DE CONTROL ÓPTIMO, entregado por Usted para su análisis, le informamos atentamente que se AUTORIZA la impresión de dicho trabajo de tesis; en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias.

ATENTAMENTE

*Excelencia en Educación Tecnológica®
Por la Tecnología de Hoy y del Futuro*

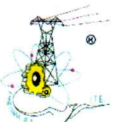
EUSEBIO BUGARÍN CARLOS
JEFE DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN



Vo. Bo.

JUAN CARLOS LEAL NODAL
SUBDIRECTOR ACADÉMICO

ccp. Archivo





Instituto Tecnológico de Ensenada
División de Estudios de Posgrado e Investigación

MAESTRÍA EN CIENCIAS EN INGENIERÍA MECATRÓNICA
DICTAMEN DEL COMITÉ TUTORIAL

Ensenada, B.C. a 15 de diciembre de 2021

DATOS DEL ESTUDIANTE			
Nombre:	Hedy César Ramírez de Ávila	No. de Control:	M05760054

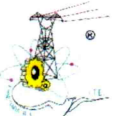
DATOS DEL PROYECTO DE TESIS	
Título:	Implementación Numérica y Comparación de Métodos de Control Óptimo
Línea de Investigación:	Robótica y Control

COMITÉ DE TESIS		
Miembro	Nombre	Firma
DIRECTOR DE TESIS (Director del comité)	Dr. Juan Antonio Rojas Quintero	
CODIRECTOR DE TESIS	Dr. Eusebio Bugarin Carlos	
MIEMBRO DEL COMITÉ	Dra. Ana Yaveni Aguilar Bustos	
MIEMBRO DEL COMITÉ	Dr. Eduardo Rodríguez Orozco	
Dictamen:	Se aprueba el trabajo de tesis, en virtud de que realizó las correcciones correspondientes conforme a las observaciones planteadas por este Comité Tutorial.	

Dr. Salvador Durazo Acevedo
Coordinador Académico

Dra. Ana Yaveni Aguilar Bustos
Presidenta del Consejo de Posgrado

STA_10_dictamen_tesis_comite_tutorial_V1.1



Bld. Tecnológico #150, Ex Ejido Chapultepec, C.P. 22780
Ensenada, Baja California, Tels. (646) 177 5680 y 177 5682
Correo electrónico: depi_ensenada@tecnm.mx
tecnm.mx/ensenada.tecnm.mx





CARTA DE CESIÓN DE DERECHOS

Ensenada, Baja California, a 15 de diciembre de 2021

Bajo protesta de decir verdad, quien suscribe Hedy César Ramírez de Ávila, con número de control M05760054, alumno de la **Maestría en Ciencias en Ingeniería Mecatrónica** y con la dirección del Dr. Juan Antonio Rojas Quintero, declaro que soy autor del trabajo original de Tesis

Implementación Numérica y Comparación de Métodos de Control Óptimo

Y otorgo mi conformidad, sin límite de temporabilidad, para ceder los derechos de reproducir y distribuir copias en su totalidad o en partes al Tecnológico Nacional de México campus Instituto Tecnológico de Ensenada (TecNM-ITE), mencionando la fuente. Lo anterior incluye que la tesis pueda ser publicada y difundida por el TecNM-ITE en los medios que considere convenientes. Asimismo, manifiesto que el TecNM-ITE queda liberado de cualquier conflicto surgido a raíz de la publicación. De igual manera, es de mi conocimiento que la publicación de la tesis no es con finalidad lucrativa, sino académica.

ATENTAMENTE

HEDY CÉSAR RAMÍREZ DE ÁVILA
ALUMNO DE LA MAESTRÍA EN CIENCIAS
EN INGENIERÍA MECATRÓNICA

STA_11_carta_cesion_derechos_V1.1



RESUMEN

Este proyecto de tesis trata sobre el estudio y análisis de distintos métodos de optimización aplicados al control de movimiento de sistemas robóticos. Entre estos métodos, se ha realizado un enfoque en el Principio del Máximo de Pontryagin, que establece las condiciones de optimalidad para obtener la mejor trayectoria de acuerdo a un índice de desempeño seleccionado.

En el marco del Principio del Máximo de Pontryagin, la elección del Hamiltoniano tiene un impacto en las variables de control resultantes. De esta forma, la función costo compone al Hamiltoniano que conduce a un comportamiento de control y movimiento deseado cuando se elige apropiadamente. Para verificar el control óptimo en el movimiento de los manipuladores robóticos, comparamos el impacto numérico de dos funciones costo en las variables de control resultantes y las posiciones de la trayectoria óptima. Las funciones costo seleccionadas se enfocan en las variables de control para que se logre un esfuerzo mínimo durante el movimiento. Se compara una función costo que se encuentra típicamente en la literatura con una que involucra los componentes del tensor de masas del robot. Se aplican a la simulación de control óptimo de dos manipuladores robóticos. Los ensayos numéricos mostraron que el tensor de masas actúa como un factor estabilizador que conduce a amplitudes de movimiento menores, mayor estabilidad numérica y tiempos de procesamiento del CPU reducidos. Por lo tanto, esta función costo es beneficiosa para la generación óptima de trayectorias y el control de los manipuladores robóticos.

Este proyecto de tesis se enmarca en el proyecto SEP-CONACYT "Investigación Científica Básica" A1-S-29874 titulado "Modelado por medio de métodos de mecánica analítica para optimizar trayectorias y diseñar controladores para sistemas robóticos". Dicho proyecto es desarrollado por los directores de tesis.

Palabras clave: Control óptimo, Función costo, Optimización de trayectorias, Manipuladores, Principio del Máximo de Pontryagin.

ABSTRACT

This thesis project deals with the study and analysis of different optimization methods applied to the motion control of robotic systems. Among these methods, it has made a focus on the Pontryagin Maximum Principle, which establishes the optimization conditions to obtain the best trajectory according to a selected performance index.

In the framework of the Pontryagin Maximum Principle, the choice of the Hamiltonian has an impact on the resulting control variables. In this way, the cost function composes the Hamiltonian which leads to a desired motion and control behavior when it is properly chosen. To verify optimal control in robotic manipulators motion, it is compared the numerical impact of two cost functions on the resulting control variables and optimal trajectory positions. The selected cost functions focus on the control variables so that minimal effort is achieved during movement. A cost function that is typically found in the literature is compared with one that involves the components of the robot's mass tensor. It is applied to the simulation of optimal control of two robotic manipulators. The numerical tests show that the mass tensor acts as a stabilizing factor leading to lower amplitudes of motion, greater numerical stability, and reduced CPU processing times. Therefore, this cost function is beneficial for optimal trajectory generation and control of robotic manipulators.

This thesis project is part of the SEP-CONACYT project "Basic Scientific Research" A1-S-29874 entitled "Modeling through analytical mechanics methods to optimize trajectories and to design controllers for robotic systems". This project is developed by the thesis supervisors.

Keywords: Optimal control, Cost function, Trajectory optimization, Manipulators, Pontryagin Maximum Principle.

DEDICATORIA

Para mi madre y hermanos que siempre han sido una fuente de alegría.

AGRADECIMIENTOS

Agrezco con cariño:

Al Instituto Tecnológico de Ensenada por ser parte de mi formación profesional.

Al Consejo Nacional de Ciencia y Tecnología por la oportunidad de seguir estudiando gracias al gran apoyo al brindarme una beca.

A mis directores de tesis, así como a los integrantes de mi comité de tesis que siempre me han brindado su apoyo y orientación.

CONTENIDO

Resumen	I
Abstract	II
Agradecimientos	IV
Contenido	VI
1. INTRODUCCIÓN	1
1.1. Antecedentes	2
1.2. Objetivos	4
1.2.1. Objetivo general	4
1.2.2. Objetivos específicos	4
1.3. Organización del documento	4
2. MARCO TEÓRICO	6
2.1. Cálculo de variaciones	6
2.1.1. La braquistócrona y el cálculo de variaciones	6
2.1.2. El Problema fundamental de los Extremos	9
2.1.3. Primeras integrales importantes	10
2.1.4. Ecuación de du Bois-Reymond y condiciones de esquina	11
2.1.5. Condiciones necesarias de Legendre y de Jacobi	12
2.1.6. Variaciones fuertes y débiles	13
2.1.7. Condición Necesaria de Weierstrass	14
2.1.8. Condiciones de Transversalidad	15
2.1.9. Integral Invariable de Hilbert	16
2.1.10. Condición fundamental suficiente	18
2.2. Sobre Problemas de Control Óptimo	19
2.2.1. Condiciones necesarias para la optimización	20
2.3. Principio del Máximo de Pontryagin (PMP)	22
2.4. Ecuaciones de control covariantes	24
2.5. Navegación de robots	25
2.6. Tiempo de cálculo y estabilidad numérica	26
3. CONTROL ÓPTIMO APLICADO A SISTEMAS ROBÓTICOS	28
3.1. Dinámica del robot	28
3.2. Dinámica de un robot manipulador de 1 GDL	29
3.3. Dinámica de un robot manipulador de 2 GDL	31
3.4. Metodología de control óptimo	34

3.4.1. Índice de desempeño y función costo	34
3.5. Impacto de la función costo en las condiciones de optimización . .	35
4. RESULTADOS DE SIMULACIÓN	37
4.1. Metodología de evaluación	37
4.2. Control óptimo de un manipulador robótico de 1 GDL	38
4.2.1. PMP aplicado al manipulador de 1 GDL	38
4.2.2. PMP aplicado al manipulador de 1 GDL (usando el tensor de masas)	39
4.3. Control óptimo de un manipulador robótico de 2 GDL	42
4.3.1. PMP aplicado al manipulador de 2 GDL	42
4.3.2. PMP aplicado al manipulador de 2 GDL (usando el tensor de masas)	45
5. CONCLUSIONES	50
Bibliografía	51
Anexos	56
A. PROBLEMAS ISOPERIMÉTRICOS	56
B. CONTROL EN TIEMPO ÓPTIMO	60
C. CÓDIGO DEL CÁLCULO EN WOLFRAM MATHEMATICA	63

LISTA DE FIGURAS

2.1. Partícula que viaja sobre un cable del punto A al punto B. 7

2.2. Trayectorias candidatas a la minimización del funcional (2.1). 8

2.3. Dos extremos con esquinas (discontinuidades) equivalentes. 11

3.1. Esquema de los manipuladores robóticos utilizados: (a) manipulador robótico de 1 GDL; (b) Manipulador robótico de 2 GDL. Los parámetros $(m_i, I_i, a_i, l_i, q_i)$ denotan respectivamente la i -ésima: masa del eslabón; momento de inercia del eslabón; distancia del centro de giro al centro de masa; longitud del eslabón; variable de configuración. 29

4.1. Control óptimo de un manipulador robótico de 1 GDL. Evolución del par RMS, potencia RMS y tiempo de cálculo del CPU necesarios para obtener una solución para diferentes valores de tiempo prescrito T . Los resultados de RMS son iguales con ECV y ambas funciones costo. El tiempo de cálculo del CPU aumenta después de que $T = 9$ s con ECV y γ_M (3.7). 41

4.2. Control óptimo de un manipulador robótico de 1 GDL. Evolución de las posiciones de las articulaciones y los momentos de torsión para cuatro valores diferentes de tiempo prescrito, $T = 0.5$ s, $T = 2.5$ s, $T = 5$ s y $T = 10$ s. Para este manipulador, las posiciones y pares de articulación obtenidos son iguales con ECV y ambas funciones costo γ_A (3.6) y γ_M (3.7). 42

4.3. Control óptimo de un manipulador robótico de 2 GDL. Evolución del par RMS, potencia RMS y el tiempo de cálculo del CPU necesarios para obtener una solución para valores diferentes de tiempo prescrito T . Los resultados de RMS son más bajos con γ_A (3.6). El tiempo de cálculo del CPU es menor con γ_M (3.7). Tenga en cuenta que los espacios en las curvas muestran que no siempre se pueden encontrar soluciones cuando se usa γ_A (3.6). Las soluciones se obtienen de forma ininterrumpida al utilizar γ_M (3.7) hasta $T = 4.9$ s. 47

4.4. Control óptimo de un manipulador robótico de 2 GDL. Evolución de las posiciones de las articulaciones y los momentos de torsión para tres valores diferentes de tiempo prescrito, $T = 0.5$ s, $T = 1.5$ s y $T = 4.9$ s. Para este manipulador, las posiciones de las articulaciones y los pares de torsión obtenidos dependen de la elección de la función costo. 48

A.1. Área encerrada por la cuerda L 57

LISTA DE TABLAS

3.1. Convención de nomenclatura de un manipulador de n GDL.	29
4.1. Valores del manipulador de 1 GDL.	39
4.2. Valores utilizados en el manipulador robótico de 2 GDL.	44
4.3. Factor de disminución en el tiempo de cómputo del CPU requerido por NDSolve para obtener una solución: relación entre funciones costo comparables usando dos procesadores diferentes.	49

1

INTRODUCCIÓN

El objetivo de la teoría del control óptimo es determinar las señales de control que harán que un proceso satisfaga las operaciones o tareas encomendadas por el usuario y al mismo tiempo minimizar (o maximizar) algún criterio de desempeño (Kirk, 2004). El control puede ser óptimo en el tiempo, es decir, alcanzar el estado deseado en un tiempo mínimo, o alcanzar este estado con costos mínimos de energía, o lograr la máxima productividad en un tiempo fijo (Grancharova y Johansen, 2005).

La teoría del control óptimo es una disciplina que tiene su origen histórico en el cálculo de variaciones, lo cual se remonta a la formulación del problema de la braquistócrona por Johann Bernoulli hace más de 300 años (Sussmann y Willems, 1997). Es importante mencionar que esta teoría se convirtió en un campo propio en la década de 1960 en conexión con el desarrollo de la exploración espacial (Pesch y Plail, 2009). De hecho, fue la ingeniería del problema de lanzar un satélite (Sputnik) a una órbita sostenida lo que generó los primeros desarrollos relevantes de la teoría. En la actualidad el control óptimo proporciona técnicas para solucionar gran cantidad de problemas en varios campos de la ciencia como: la biomecánica (Zignoli *et al.*, 2017); la economía (Chenavaz, 2017); la ingeniería aeroespacial (Longuski *et al.*, 2014); la medicina (Schättler y Ledzewicz, 2012); o la robótica avanzada (Putkaradze y Rogers, 2020; Vezvari *et al.*, 2020); entre otros. En contraste con la robótica industrial, la robótica avanzada se refiere a la parte de la robótica que estudia robots con marcadas características de autonomía y con aplicaciones que resuelven problemas de operación en ambientes adversos (espacial, submarino, nuclear, militar, etc.) o que ejecutan tareas de servicio (aplicaciones domésticas, ayuda médica, asistencia a discapacitados, agricultura, etc.).

En este trabajo de tesis se hará énfasis en el análisis de leyes de control óptimo directamente basadas en las ecuaciones de movimiento de los sistemas dinámicos bajo estudio. Lo que permite visualizar que este tema corresponderá con la intersección de diversas áreas de la ciencia como: Matemáticas, Mecáni-

ca Analítica, Mecánica Computacional, Métodos Numéricos, Optimización y Teoría de Control.

En este sentido existen dos métodos tradicionalmente considerados para optimizar los movimientos de un sistema dinámico. El Principio del Máximo de Pontryagin (PMP), que requiere la resolución de un sistema Hamiltoniano acoplado de ecuaciones diferenciales ordinarias que gobiernan las variables conjugadas y de estado (Pontryagin *et al.*, 1962); y el Principio de Optimalidad de Bellman, que requiere la solución de un sistema de ecuaciones diferenciales parciales (Bellman, 1957). Es importante mencionar que, en relación al PMP y más recientemente, en Rojas Quintero *et al.* (2013) y en Rojas-Quintero *et al.* (2021a) se ha desarrollado una alternativa de control óptimo en donde las condiciones de optimalidad conducen a ecuaciones de control geométricas (lo que significa que tienen una estructura Riemanniana (Grinfeld, 2013)).

Por lo tanto, uno de los objetivos principales de este proyecto de tesis es investigar y estudiar las técnicas clásicas de control óptimo que se encuentran presentes en la literatura; para luego implementar dichas técnicas y llevar a cabo una comparación con esta alternativa propuesta en Rojas-Quintero *et al.* (2021a). Estas comparaciones se realizarán desde un punto de vista de simulación numérica.

1.1 ANTECEDENTES

Regresando al PMP, se puede decir que éste es uno de los métodos más usados para sintetizar el control óptimo. Por ejemplo, en Asgari y Nikoobin (2020) se utiliza para resolver el objetivo del lanzamiento óptimo de un objeto con un brazo robótico; de manera que se minimiza el esfuerzo del lanzamiento en un tiempo final especificado. En Herman y Kozlowski (2006) se presentan algunas observaciones que son útiles para evaluar la dinámica de un manipulador con control óptimo. Se demostró que al describir la dinámica de un manipulador mediante dos ecuaciones de movimiento de primer orden, se pueden evaluar acoplamientos mecánicos. También se demostró que si se usan las cuasi-velocidades no normalizadas, entonces es posible calcular independientemente: la inercia transferida por cada articulación del manipulador; la deformación de la trayectoria de velocidad óptima; la energía cinética transferida por cada eslabón; y el momento de la articulación que corresponde a la cuasi-velocidad apropiada.

En Callies y Rentrop (2008) se aborda un estudio e investigación sobre el tratamiento de los problemas de control óptimo en manipuladores de eslabones rígidos por métodos indirectos. Se encontró que los enfoques basados en el PMP

proporcionan una excelente herramienta para calcular trayectorias de referencia óptima en manipuladores de múltiples eslabones con altos desempeños, o para ser aplicadas en combinación con el análisis estructural de movimientos de sistemas robóticos.

Para problemas de estados restringidos que utilizan control óptimo, se puede ver en Arutyunov y Karamzin (2020) que la atención se centra en cuestiones tales como las condiciones de regularidad y controlabilidad, la no degeneración y la normalidad del principio máximo y la continuidad del multiplicador de medidas.

El problema de seguimiento de una ruta óptima también se puede beneficiar con la teoría de control óptimo. Por ejemplo, en Kaserer *et al.* (2019) se presenta un nuevo enfoque de programación dinámica. La característica importante de este algoritmo es la posibilidad de limitar las sacudidas y las tasas de cambio del par tomando en cuenta las características de los motores e incluyendo fricción viscosa en el modelo del manipulador.

Como se ha mencionado, el PMP es muy utilizado para optimizar el movimiento de robots manipuladores. Por ejemplo, en Mirz *et al.* (2018) se aprovecha el carácter repetitivo de alta velocidad de tareas de levantar y poner (término común en manufactura, designado por “pick and place” en inglés), de manera que al agregar elementos elásticos como resortes en las articulaciones del manipulador es posible generar el movimiento deseado utilizando el movimiento natural del sistema y reduciendo el consumo de energía del manipulador. Para utilizar este método se desarrolló un algoritmo de planificación de trayectorias a la medida para obtener las condiciones de optimalidad. Estas condiciones llevan a un problema de valor de frontera en dos puntos (TPBVP por sus siglas en inglés: Two-Point Boundary-Value Problem).

Recientemente se propuso el método de función generadora para los problemas de control óptimo que proporciona una familia de trayectorias óptimas para distintas condiciones de frontera. El método propuesto reduce el esfuerzo computacional en línea para ejecutar la integración numérica requerida en los métodos de función generadora convencional. Esta metodología demuestra ser útil para generar trayectorias en línea (Hao, 2014). En Chen *et al.* (2017) se aplica el método de funciones de doble generación para optimizar el movimiento controlado de un robot manipulador. Después de establecer la dinámica del robot, se diseña el índice de rendimiento de manera que se formalice un problema de control de energía óptima. En segundo lugar, se introduce la función generadora y se emplean un par de ellas para generar trayectorias óptimas.

1.2 OBJETIVOS

1.2.1 *Objetivo general*

Investigar sobre métodos numéricos de mecánica analítica para robótica avanzada consistentes en sistemas no lineales de control óptimo para aplicaciones de navegación.

Este objetivo general debe permitir la creación de herramientas para las áreas de Ingeniería Mecánica, Teoría de Control y Mecatrónica; y debe tomar en cuenta la estructura geométrica del sistema dinámico bajo estudio.

1.2.2 *Objetivos específicos*

1. Investigar y estudiar técnicas clásicas de control óptimo presentes en la literatura.
2. Implementar dichas técnicas y efectuar una comparación entre ellas desde un punto de vista numérico (tiempo de cálculo, complejidad y estabilidad numérica).
3. Desarrollar y extender los métodos y algoritmos estudiados al control óptimo de sistemas robóticos.

1.3 ORGANIZACIÓN DEL DOCUMENTO

El capítulo 2 resume el marco teórico sobre la teoría de control óptimo. Después, en el capítulo 3, se introduce la aplicación principal de este trabajo de tesis, es decir, el control óptimo de los movimientos de dos manipuladores robóticos. En la sección 3.2 se introduce el estudio de un manipulador con un solo grado de libertad. Luego se estudiará el caso de un manipulador con dos grados de libertad en la sección 3.3. El modelo matemático de este tipo de manipuladores combina muchas de las complejidades presentes en los sistemas estudiados por la robótica avanzada; principalmente: el modelo se compone de un sistema de ecuaciones de movimiento no-lineales y los eslabones que componen al sistema inducen efectos de inercia entre ellos. En el capítulo 4 se muestran los resultados de simulación. En la sección 4.1 se presenta la metodología de evaluación, donde se muestran los pasos que se siguen para evaluar el impacto de cada función costo. Después, en la sección 4.2, se desarrolla el control óptimo para un manipulador robótico de 1 GDL y en la sección 4.3 se desarrolla el control

óptimo para un manipulador robótico de 2 GDL. Para finalizar, se muestran las conclusiones en el capítulo 5.

2

MARCO TEÓRICO

2.1 CÁLCULO DE VARIACIONES

El cálculo de variaciones es la rama de las matemáticas que considera problemas de extremos. Proporciona técnicas para determinar cuándo una integral definida particular llega a un máximo o un mínimo (o, más generalmente, las condiciones para que la integral sea *estacionaria*). El cálculo de variaciones responde principalmente a las siguientes preguntas (Hamill, 2013; Mesterton-Gibbons, 2009):

- ¿Cuál es la trayectoria que da la distancia más corta entre dos puntos en un plano? (Una línea recta).
- ¿Cuál es el camino que da la distancia más corta entre dos puntos en una esfera? (Un geodésico o “gran círculo”).
- ¿Cuál es la forma de la curva de longitud dada que encierra el área más grande? (Un círculo).
- ¿Cuál es la forma de la región del espacio que encierra el mayor volumen para un área de superficie dada? (Una esfera).

Como se puede ver, son preguntas muy elementales y la conexión con la optimización de trayectorias complejas de robots (por ejemplo) parece ser contraintuitiva. Sin embargo, la tarea consiste en expresar el problema de tal manera que se asemeje a una de las preguntas anteriores.

2.1.1 *La braquistócrona y el cálculo de variaciones*

En 1696, Johann Bernoulli publicó su solución para encontrar el camino tomado por una partícula, bajo el efecto de la gravedad, para pasar de un punto A a un punto B (ver figura 2.1) en el menor tiempo posible (Mesterton-Gibbons,

2009; Sussmann y Willems, 1997). Podemos resumir el procedimiento de la siguiente manera: se trata de minimizar una expresión del tiempo de trayectoria t dado por el funcional $J(y(x))$ según (2.1). Nótese que se le llama funcional a una función de otra función. En breve, se trata de encontrar la función $y = y(x)$ (el camino) que minimice el valor de t mediante¹

$$t = J(y(x)) = \frac{1}{\sqrt{2g}} \int_0^1 \frac{\sqrt{1 + (y')^2}}{\sqrt{1 - y}} dx \quad (2.1)$$

donde $g = 9.81 \text{ m/s}^2$ y $y' = \frac{dy}{dx}$.

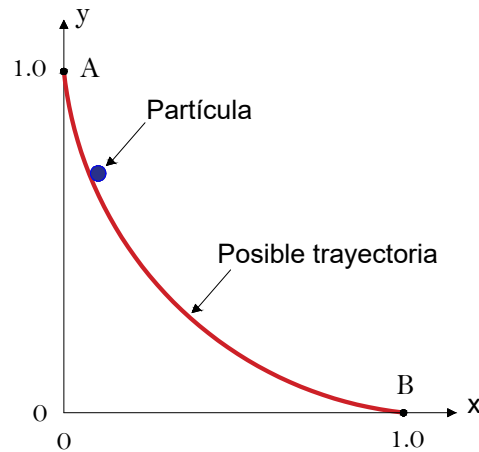


Figura 2.1: Partícula que viaja sobre un cable del punto A al punto B.

Ahora bien, al cambiar la curva sobre la cual se desliza la partícula hacia abajo, también cambiará el valor de la integral. Por lo tanto (2.1) es una función de $y = y(x)$; es decir, es una funcional (usaremos corchetes en $J[y]$, como en Mesterton-Gibbons (2009), para evitar escribir $J(y(x))$). Cambiando la curva y por una línea recta, por un cuarto de círculo, por una función de prueba ($y(x) = 1 - x^\epsilon$, $\epsilon = 0.5397$) o por una cicloide², se encuentra que el menor tiempo para $J[y]$ se obtiene cuando usamos la cicloide (ver Figura 2.2). Podemos resaltar que

¹Una introducción al cálculo de variaciones y una presentación más extensa de este problema se pueden encontrar en Hamill (2013); Mesterton-Gibbons (2009); Sussmann y Willems (1997).

²La curva que describe un punto de una circunferencia que rueda sobre una recta sin deslizar.

en (2.1), multiplicar por una constante no tiene efecto sobre el minimizador, por lo tanto se pueden omitir términos constantes; de manera que

$$J[y] = \int_0^1 \sqrt{\frac{1 + (y')^2}{1 - y}} dx.$$

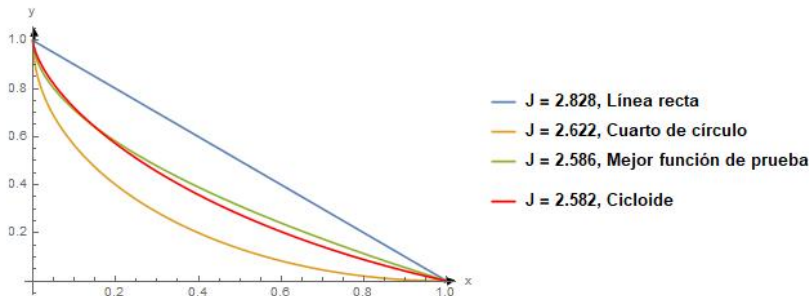


Figura 2.2: Trayectorias candidatas a la minimización del funcional (2.1).

Pero, ¿cómo podríamos saber que la cicloide es la curva que minimiza el tiempo de tránsito, en otras palabras, que la cicloide es la braquistócrona? En esta etapa no podríamos saberlo porque necesitamos el cálculo de variaciones. La pregunta será contestada en las subsecciones siguientes.

Antes de continuar, podemos plantear de manera general el problema del control óptimo. Se trata de minimizar una funcional

$$J[y] = \int_a^b F(x, y, y') dx \quad (2.2)$$

en donde $F = F(x, y, y')$ es llamada comúnmente *función costo*. El funcional integral (2.2) está sujeto a las condiciones de frontera

$$y(a) = \alpha, \quad y(b) = \beta. \quad (2.3)$$

Se busca, entre todas las curvas Γ definidas por $y = y(x)$, aquella que minimice (2.2) sujeta a (2.3). Γ debe ser suficientemente derivable al menos por trozos en el dominio $[a, b]$, tomando en cuenta que en la práctica no tendría sentido que hubiera roturas en la curva. La curva y debe de ser continua con una derivada y' que exista y sea continua excepto en tal vez algún número finito de puntos en donde brinque con alguna magnitud finita. Llamemos a esta clase de

funciones D_1 . La función $y = \bar{y}(x)$ es admisible si $y \in D_1$, pero también si satisface las condiciones (2.3). Definamos también C_1 como la clase de funciones continuamente derivables o más comúnmente llamadas “suaves”; y C_2 como la clase de funciones suaves que son doblemente continuamente derivables. Por construcción, $C_2 \subset C_1 \subset D_1$.

2.1.2 El Problema fundamental de los Extremos

Para contestar la pregunta anterior utilizamos la Ecuación de Euler-Lagrange (EEL) que es la condición fundamental (Mesterton-Gibbons, 2009) para extremar una funcional del tipo (2.2):

$$\frac{\partial F}{\partial y} - \frac{d}{dx} \left\{ \frac{\partial F}{\partial y'} \right\} = 0. \quad (2.4)$$

Note que:

- La ecuación (2.4) es una ecuación diferencial ordinaria. En ausencia de condiciones de frontera tiene una familia de soluciones de dos parámetros donde cualquiera de estas soluciones se conoce como *extremo*.
- Un extremo es *solamente* un candidato para ser minimizador de J .

Entonces, ¿por qué no llamarlo mínimo? La respuesta es que si y maximizara J , entonces tendría que minimizar $-J$; pero reemplazar F por $-F$ no tiene ningún efecto sobre la EEL. En efecto, los problemas de maximización se convierten tan fácilmente en problemas de minimización; así que la minimización es el único tipo de extremización que consideraremos. La extremalidad de una curva candidata no la convierte en extremizadora, por lo que existen tres problemas principales:

1. Hay que probar que el extremo es un minimizador.
2. El extremo puede no ser un minimizador.
3. Puede no haber extremo que cumpla con las condiciones de frontera.

Para resolver el problema principal 1 se puede proceder por minimización directa; es decir, hay que demostrar que:

$$J(\phi(x) + \epsilon\eta(x)) - J(\phi(x)) \geq 0$$

en donde $y = \phi(x)$ es la función candidata a ser minimizadora; ϵ es una variable escalar suficientemente pequeña; $\eta(x) \in C_1^3$ en $[a, b]$ tal que $\eta(a) = 0 = \eta(b)$.

Una aplicación importante para los temas relevantes a esta tesis se encuentra en Robótica, y tiene que ver con el Principio de Mínima Acción que estipula que, considerando la *acción* o *integral de acción*

$$I = \int_{t_0}^{t_1} L dt = \int_{t_0}^{t_1} (K - V) dt \quad (2.5)$$

en donde $L = K - V$ es el Lagrangiano del sistema (K y V son la energía cinética y la energía potencial del sistema, respectivamente), el movimiento del sistema dinámico cumple esta acción *estacionaria*. Al aplicar las EEL a L , se obtienen las ecuaciones de movimiento del sistema dinámico (Mesterton-Gibbons, 2009; Spong *et al.*, 2005). En general, esta acción cumple con el mínimo.

2.1.3 Primeras integrales importantes

Aunque la EEL para el funcional (2.2) es en general una ecuación diferencial ordinaria (EDO) no lineal de segundo orden, siempre se reduce a una EDO de primer orden en dos casos importantes. El primer caso es cuando $F = F(y, y')$. Utilicemos a partir de ahora la convención de derivadas parciales abreviadas como por ejemplo $F_x = \frac{\partial F}{\partial x}$. Este caso se cumple entonces cuando $F_x = 0$. En este caso el Hamiltoniano

$$H(y, y') = y'F_{y'} - F = \text{constante}. \quad (2.6)$$

El segundo caso es cuando $F = F(x, y')$, es decir, cuando $F_y = 0$. Entonces la EEL (2.4) se convierte en

$$F_{y'} = \text{constante}. \quad (2.7)$$

Las ecuaciones (2.6) y (2.7) facilitan la obtención de los extremos porque son EDO de primer orden. Sin embargo, no basta con obtener el extremo puesto que necesitamos saber si es una función suave o si por el contrario tiene esquinas (discontinuidades, ver figura 2.3) o si es una función continua por trozos. Esto se analizará en las siguientes subsecciones.

³La clase de todas las funciones continuamente derivables definidas en $[a, b]$.

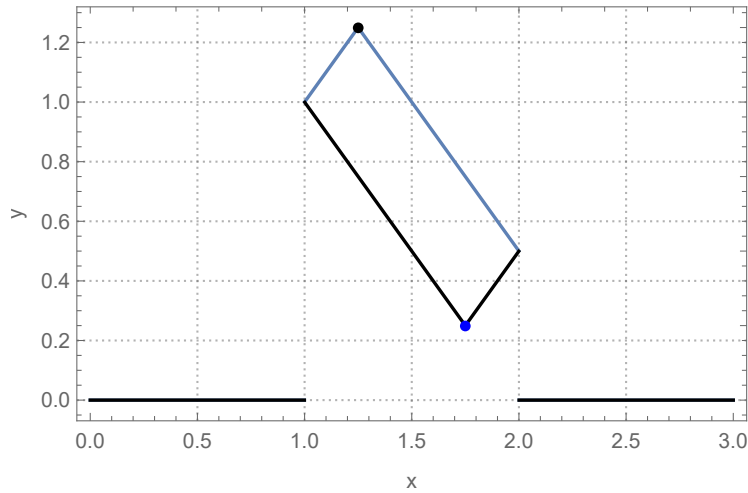


Figura 2.3: Dos extremos con esquinas (discontinuidades) equivalentes.

2.1.4 Ecuación de du Bois-Reymond y condiciones de esquina

Hasta ahora hemos supuesto que $\eta \in C_2^4$ y deducido que ϕ debe satisfacer la EEL (2.4). Ahora supongamos que $\eta \in D_1$, con lo cual se admiten esquinas (discontinuidades en el tercer argumento). Mantengamos que $F_y(x, \phi, \phi')$ es continua en las esquinas. Entonces una condición necesaria para que $\phi \in D_1$ minimice J es que

$$F_{y'}(x, \phi, \phi') = \int_a^x F_y(\xi, \phi, \phi') d\xi + C \quad (2.8)$$

en donde C es constante. Esta es la ecuación de du Bois-Reymond que por derivación adecuada, también lleva a la EEL (2.4) cuando $\phi \in C_1$. Aunque el integrando del lado derecho de la ecuación de du Bois-Reymond (2.8) es discontinuo en la esquina c porque salta de $F_\phi(c, \phi(c), \omega_1)$ a $F_\phi(c, \phi(c), \omega_2)$ la integral en sí es continua; y, por supuesto, la constante C es continua. Por tanto, el lado derecho de (2.8) debe ser continuo en c . Pero el lado derecho siempre es igual al lado izquierdo. Deducimos que el lado izquierdo también debe ser continuo. Esta necesidad de continuidad en las esquinas, lleva a la *primera condición de esquinas de Weierstrass-Erdmann*:

$$F_{y'}(c, \phi(c), \omega_1) = F_{y'}(c, \phi(c), \omega_2) \quad (2.9)$$

⁴La clase de todas las funciones continuamente derivables (suaves) definidas en $[a, b]$ que también son continuamente dos veces derivables.

en donde $c \in [a, b]$ es la abscisa de la esquina y $\omega_1 \neq \omega_2$ tal que

$$\begin{aligned}\omega_1 &= \phi'(c-) = \lim_{x \rightarrow c-} \phi'(x), \\ \omega_2 &= \phi'(c+) = \lim_{x \rightarrow c+} \phi'(x).\end{aligned}$$

Remarquemos que cualquier extremo roto debe satisfacer la EEL excepto en las esquinas; y en cualquier esquina debe cumplir con (2.9). Esta condición de esquina se puede utilizar para excluir la posibilidad de un extremo roto. Notemos también que siempre que $F_{y'y'} \neq 0$, $F_{y'}$ varía monótonicamente con respecto al tercer argumento y por lo tanto no puede tener el mismo valor para $\omega_1 \neq \omega_2$. Entonces, si

$$F_{y'y'} > 0 \quad \forall (x, y, y') \quad (2.10)$$

el problema es regular (es decir, que no tiene extremos admisibles por trozos, ver sección 2.1.5) y no admite extremos discontinuos.

La segunda condición de Weierstrass-Erdmann para extremos admisibles por trozos (o segunda condición de esquina) es que el Hamiltoniano tenga continuidad en la esquina:

$$H(c, \phi(c), \omega_1) = H(c, \phi(c), \omega_2), \quad (2.11)$$

siempre que y' salte de ω_1 a ω_2 . Entonces, combinando las dos condiciones, en cualquier esquina las cantidades $F_{y'}(x, y, y')$ y $H(x, y, y')$ deben ser ambas continuas en $x = c$, incluso cuando y' tenga discontinuidad y salte de ω_1 a ω_2 .

2.1.5 Condiciones necesarias de Legendre y de Jacobi

Definiendo

$$\begin{aligned}P(x) &= F_{\phi'\phi'}, \\ Q(x) &= F_{\phi\phi} - \frac{dF_{\phi\phi'}}{dx},\end{aligned}$$

la integral

$$J''(0) = \frac{1}{2} \int_a^b (P\eta'^2 + Q\eta^2) \geq 0$$

solamente si $P \geq 0$. Esto es una condición necesaria para que $y = \phi(x)$ produzca el mínimo en la funcional integral (2.2) y se conoce como *condición necesaria de Legendre* (CNL):

$$F_{\phi'\phi'} = F_{y'y'}(x, \phi(x), \phi'(x)) \geq 0 \quad \forall x \in [a, b] \quad (2.12)$$

Si la condición se verifica, entonces se trata de un problema regular, es decir, sin extremos admisibles por trozos, como fue mencionado en la subsección anterior. Notemos que

$$F_{\phi'\phi'} > 0 \quad \forall x \in [a, b], \quad (2.13)$$

constituye la condición *fuerte* de Legendre.

Consideremos ahora la ecuación de Jacobi (EJ), que es una EDO lineal homogénea de segundo orden (también se conoce como la ecuación arbitraria):

$$P(x)\eta'' + P'(x)\eta' = Q(x)\eta, \quad (2.14)$$

para la cual $\eta(a) = \eta(b) = 0 \quad \forall x \in [a, b]$. Si $\eta(c) = 0 \quad \forall c \in [a, b]$, entonces ϕ no es un minimizador de $J[y]$. Esto se conoce como *condición necesaria de Jacobi* (CNJ). En este contexto, c es llamado punto conjugado⁵. Por lo tanto no pueden existir puntos conjugados para que se cumpla la CNJ.

La ecuación (2.14) al ser lineal y homogénea provoca que cualquier solución no-nula que verifique la CNJ pueda ser buscada con las condiciones de frontera

$$\eta(a) = 0, \quad \eta'(a) = 1.$$

Notemos que la CNJ se cumple automáticamente siempre que $F_y = 0$, o bien que $F_{\phi'\phi'} > 0$ (condición fuerte de Legendre). En fin, remarquemos antes de continuar que ni la condición de Legendre, ni la condición de Jacobi están diseñadas para variaciones fuertes: $J(\epsilon = 0)$ es un mínimo débil.

2.1.6 Variaciones fuertes y débiles

Definamos la variación de y como la diferencia δy entre la función de prueba y_ϵ y el minimizador ϕ :

$$\delta y(x) = y_\epsilon(x) - \phi(x).$$

Definamos también la variación de y' como la diferencia $\delta y'$ entre las respectivas derivadas:

$$\delta y'(x) = y'_\epsilon(x) - \phi'(x) = \frac{d\delta y}{dx}.$$

- Para una variación fuerte, $|\delta y|$ es pequeño para todo $x \in [a, b]$ pero $|\delta y'|$ no necesita estar limitado
- Para una variación débil, $|\delta y'|$ es pequeño para todo $x \in (a, b)$

⁵En este contexto, c no define una esquina.

- Un mínimo débil es un mínimo sobre todas las variaciones débiles
- Un mínimo fuerte es un mínimo sobre todas las variaciones fuertes
- Un mínimo global es un mínimo sobre todas las variaciones sean grandes o pequeñas
- Cada mínimo global es también un mínimo fuerte, y cada mínimo fuerte es también un mínimo débil, pero no al revés.

2.1.7 Condición Necesaria de Weierstrass

Recordemos las condiciones necesarias:

- Ecuación de Euler-Lagrange (EEL), que es la condición fundamental para extremar una funcional del tipo (2.2);
- Condición Necesaria de Legendre (CNL), que es una condición necesaria para que $y = \phi(x)$ produzca el mínimo en la funcional integral (2.2);
- Condición Necesaria de Jacobi (CNJ), la cual nos permite saber si en la funcional hay puntos conjugados; es decir, nos permite saber si ϕ es un minimizador de $J[y]$.

Definimos la función de exceso E de Weierstrass por

$$E(x, y, y', \omega) = F(x, y, \omega) - F(x, y, y') - (\omega - y')F_{y'}(x, y, y').$$

Recordando que ahora $c \in [a, b]$ no es necesariamente una esquina, la condición necesaria de Weierstrass (CNW) para que $\phi \in D1$ sea un minimizador de J , es que

$$E(x, \phi(x), \phi'(x), \omega) \geq 0 \tag{2.15}$$

$\forall x \in [a, b], \forall \omega \in \mathbb{R}$. Si ϕ tiene una esquina en $x = c$, entonces interpretamos que (2.15) significa tanto $E(x, \phi(c), \phi'(c-), \omega) \geq 0$ como $E(x, \phi(c), \phi'(c+), \omega) \geq 0$ para todo $\omega \in \mathbb{R}$ (estas dos desigualdades siguen por continuidad, de tomar los límites cuando $x \rightarrow c-$ y cuando $x \rightarrow c+$). Es interesante notar que por reacción en cadena:

1. La CNW se cumple automáticamente cuando se cumple la condición fuerte de Legendre (2.13).
2. La CNW implica la CNL (2.12).
3. La CNW implica la segunda condición de esquina de Weierstrass-Erdmann (2.11)

2.1.8 Condiciones de Transversalidad

Supongamos que las condiciones de frontera no sean del tipo (2.3), es decir puntos fijos, sino puntos variables sobre alguna curva; entonces se requieren condiciones de transversalidad. Supongamos que los puntos finales de Γ_ϵ (familia de curvas candidatas) se encuentran en las curvas Λ_A (para el límite inferior) y Λ_B (para el límite superior) con ecuaciones paramétricas

$$x = x_A(\epsilon), y = y_A(\epsilon) \text{ para } \Lambda_A,$$

$$x = x_B(\epsilon), y = y_B(\epsilon) \text{ para } \Lambda_B,$$

tales que

$$y(x_A(\epsilon), \epsilon) = y_A(\epsilon),$$

$$y(x_B(\epsilon), \epsilon) = y_B(\epsilon).$$

Consideremos el punto final inferior A . Si A está fijo, entonces se debe satisfacer que $x_A = \text{constante}$ y $y_A = \text{constante}$ y por lo tanto se debe cumplir con la primera condición de transversalidad

$$dx_A = 0, \quad dy_A = 0. \quad (2.16)$$

En el caso en el que A está restringido a permanecer en una línea vertical $x = \text{constante}$, se dice que y_A es *libre*. En este caso, $dx_A = 0$ pero $dy_A \neq 0$ y por lo tanto se debe cumplir con la segunda condición de transversalidad

$$F_{\phi'} = 0 \text{ en } A. \quad (2.17)$$

Enseguida, se puede desear que A esté restringido a permanecer sobre una línea horizontal $y = \text{constante}$. Se dice entonces que x_A es libre. Entonces $dy_A = 0$ con $dx_A \neq 0$ y por lo tanto se debe cumplir con la tercera condición de transversalidad

$$H(x, \phi, \phi') = 0 \text{ en } A. \quad (2.18)$$

Finalmente, el caso más general es cuando $dx_A \neq 0$, $dy_A \neq 0$, entonces se debe cumplir con la cuarta condición de transversalidad

$$F_{\phi'} \frac{dy}{dx} = H(x, \phi, \phi') \text{ en } A \quad (2.19)$$

en donde $\frac{dy}{dx}$ es evaluado en Λ_A , la curva en donde se encuentra la condición o punto A .

2.1.9 Integral Invariable de Hilbert

Un campo de extremales es una familia de extremales de un parámetro que cubre una región abierta R del plano en el sentido de que una, y sólo una, de sus curvas pasa por cada punto de R ; por lo tanto, define una función de dos variables, que denotamos por ρ y llamamos campo de dirección de la familia. Por construcción, la pendiente de la curva que atraviesa (x, y) es

$$y' = \rho(x, y) \quad (2.20)$$

y usando una y' subindicada para denotar diferenciación con respecto al tercer argumento,

$$F_y(x, y, \rho(x, y)) - \frac{d}{dx}\{F_{y'}(x, y, \rho(x, y))\} = 0 \quad (2.21)$$

de manera idéntica, por que un extremal es una solución de la EEL. Cuando todos los extremos emanan de un solo punto límite, el campo de extremos se denomina campo central o lápiz.

Dado un campo de extremales, Γ es cualquier curva entre (a, α) y (b, β) que se encuentre completamente dentro de la región R que cubre el campo de extremales. Entonces podemos definir la integral

$$K[\Gamma] = \int_a^b \{F(x, y, \rho(x, y)) + (\frac{dy}{dx} - \rho(x, y))F_{y'}(x, y, \rho(x, y))\} dx \quad (2.22)$$

donde la integral es evaluada a lo largo de Γ .

$$H(x, y, y') = y'F_{y'}(x, y, y') - F(x, y, y') \quad (2.23)$$

y

$$p(x, y, y') = F_{y'}(x, y, y'). \quad (2.24)$$

Podemos reescribir (2.22) como

$$K[\Gamma] = \int_{\Gamma} u_1 dx + u_2 dy \quad (2.25)$$

donde tenemos las abreviaturas

$$u_1 = -H(x, y, \rho(x, y)), \quad u_2 = p(x, y, \rho(x, y)). \quad (2.26)$$

Denotando (a, α) y (b, β) por A y B , respectivamente, tenemos Γ_1 y Γ_2 dos caminos diferentes de A a B dentro de R con Γ_2 por encima de Γ_1 y sea $-\Gamma_2$ denotado

como Γ_2 atravesando en la dirección opuesta, es decir, de B a A. Entonces, según el teorema de Stokes

$$K[\Gamma_1] - K[\Gamma_2] = \iint_S \left(\frac{\partial u_2}{\partial x} - \frac{\partial u_1}{\partial y} \right) dx dy \quad (2.27)$$

donde $\mathbf{u} = u_1 \mathbf{i} + u_2 \mathbf{j}$, \mathbf{i} y \mathbf{j} son vectores unitarios ortogonales en las direcciones de los ejes x y y , $\mathbf{k} = \mathbf{i} \times \mathbf{j}$ y S es la región plana dentro de R que $\Gamma_1 \cup -\Gamma_2$ encierra. Usando ρ como abreviatura de $\rho(x, y)$ y usando una y' subindicada para denotar una derivada parcial de p o de F o cualquier derivada parcial de la misma con respecto al tercer argumento de esa función, obteniendo

$$\frac{\partial u_2}{\partial y} = p_x + p_{y'} \rho_x = F_{y'x} + F_{y'y'} \rho_x \quad (2.28)$$

de (2.24) y (2.26). De (2.23) y (2.26) se tiene

$$\begin{aligned} \frac{\partial u_1}{\partial y} &= \frac{\partial}{\partial y} \{F(x, y, \rho) - \rho F_{y'}(x, y, \rho)\} \\ &= \frac{\partial}{\partial y} \{F(x, y, \rho)\} - \frac{\partial}{\partial y} \{\rho F_{y'}(x, y, \rho)\} \\ &= \frac{\partial}{\partial y} \{F(x, y, \rho)\} - \frac{\partial \rho}{\partial y} F_{y'}(x, y, \rho) - \rho \frac{\partial}{\partial y} \{F_{y'}(x, y, \rho)\} \\ &= \{F_y + F_{y'} \rho_y\} - \{F_{y'} \rho_y + \rho (F_{y'y} + F_{y'y'} \rho_y)\} \\ &= F_y - \rho F_{y'y} - \rho F_{y'y'} \rho_y, \end{aligned}$$

restando esta ecuación de (2.28) resulta

$$\frac{\partial u_2}{\partial x} - \frac{\partial u_1}{\partial y} = F_{y'x} + \rho F_{y'y} + F_{y'y'} (\rho_x + \rho \rho_y) - F_y. \quad (2.29)$$

A través de cada punto de R , sin embargo, pasa exactamente un extremal, en el que $y' = \rho(x, y)$. Derivando respecto a x por la regla de la cadena se llega a

$$y'' = \rho_x + \rho_y \frac{dy}{dx} = \rho_x + \rho \rho_y,$$

por lo tanto (2.29) se convierte en

$$\begin{aligned} \frac{\partial u_2}{\partial x} - \frac{\partial u_1}{\partial y} &= F_{y'x} + y' F_{y'y} + F_{y'y'} y'' - F_y \\ &= \frac{d}{dx} (F_{y'}) - F_y = 0 \end{aligned}$$

por la regla de la cadena y (2.21). Y (2.27) se reduce a $K[\Gamma_1] = K[\Gamma_2]$. Por lo tanto $K[\Gamma]$ es independiente de la trayectoria: su valor depende sólo de los puntos finales A, B de la curva. Nos referimos a $K[\Gamma]$ definido por (2.22) como la Integral Invariante de Hilbert.

2.1.10 Condición fundamental suficiente

Supóngase que la curva Γ_* , con ecuación $y = \phi(x)$, punto final inferior (a, α) o A y un punto final superior (b, β) o B , es un candidato para minimizador de

$$J[y] = \int_a^b F(x, y, y') dx$$

y si Γ_* está incrustada en un campo de extremales con un campo de dirección ρ , entonces (2.22), es decir, la Integral Invariable de Hilbert

$$K[\Gamma] = \int_a^b \{F(x, y, \rho(x, y)) + (\frac{dy}{dx} - \rho(x, y))F_{y'}(x, y, \rho(x, y))\} dx$$

se puede utilizar para derivar una condición suficiente. Se debe tener en cuenta

$$\frac{dy}{dx} = \phi'(x) = \rho(x, \phi(x)) \quad (2.30)$$

en Γ_* por (2.20). Por tanto, de lo anterior

$$K[\Gamma] = \int_a^b \{F(x, \phi, \rho(x, \phi)) + 0\} dx = \int_a^b F(x, \phi(x), \phi'(x)) dx = J[\Gamma_*].$$

Entonces para cualquier curva Γ de A a B tenemos variación total

$$\Delta J = J[\Gamma] - J[\Gamma_*] = J[\Gamma] - K[\Gamma_*] = J[\Gamma] - K[\Gamma]$$

porque K es independiente de la trayectoria. La variación total es una diferencia entre integrales sobre dos curvas diferentes, como una diferencia entre integrales sobre la misma curva, por lo tanto,

$$\begin{aligned} \Delta J &= J[\Gamma] - K[\Gamma] \\ &= \int_a^b F(x, y, y') dx - \int_a^b \{F(x, y, \rho) + (y' - \rho)F_{y'}(x, y, \rho)\} dx \\ &= \int_a^b \{F(x, y, y') - F(x, y, \rho) - (y' - \rho)F_{y'}(x, y, \rho)\} dx \\ &= \int_a^b E(x, y, \rho, \omega) dx \end{aligned}$$

de (2.15), donde

$$\omega = y'$$

es la pendiente de Γ en el punto (x, y) y E denota la función de exceso de Weierstrass. Ahora debido a que Γ es cualquier curva suave a trozos de A a B que se encuentra totalmente en R , está claro que $E(x, y, \rho, \omega) \geq 0$ es una condición suficiente para $\Delta[J] \geq 0$. Es decir: si Γ_* está integrada en un campo de extremales con campo de dirección ρ y $E(x, y, \rho, \omega) \geq 0$ para todos los $\omega \in \mathbb{R}$ factibles, entonces $J[y]$ logra un mínimo fuerte en Γ_* .

1. Se considera que Γ_* está incrustada en R si R contiene todos los puntos de Γ_* con la posible excepción de A .
2. El extremal admisible Γ_* no pertenece a todos los campos de extremales que se pueden construir a partir de la familia general de soluciones de la EEL, denotada por $y = y(x, k, l)$. Si las condiciones de frontera pueden satisfacerse con $k = k^*$ y $l = l^*$, entonces dos subfamilias de un parámetro garantizadas para contener Γ_* son Γ_k , donde Γ_k tiene la ecuación $y = y(x, k) = Y(x, k, l^*)$ y Γ_l , donde Γ_l tiene la ecuación $y = y(x, l) = Y(x, k^*, l)$ y normalmente al menos uno es un campo adecuado.
3. Para un problema regular, establecer la suficiencia equivale a demostrar que Γ_* puede ser incrustada en un campo de extremales.

2.2 SOBRE PROBLEMAS DE CONTROL ÓPTIMO

No todos los problemas de minimización se prestan al tratamiento mediante el cálculo de variaciones tan fácilmente como el problema de la braquistócrona. El cálculo de variaciones requiere funciones admisibles suaves por partes; mientras que las funciones admisibles continuas por partes surgen naturalmente en una variedad de configuraciones de control de minimización de tiempo y otros, produciendo problemas que requieren los desarrollos más recientes de la teoría de control óptimo. Así, una diferencia esencial entre el cálculo de variaciones y la teoría del control óptimo es que las funciones admisibles suaves por partes dan paso a controladores admisibles continuos por partes.

El problema prototípico de control óptimo en física es conducir una partícula de masa m a lo largo del eje X en el menor tiempo posible desde $X = a$ hasta $X = b$ bajo una fuerza aplicada F_A cuya magnitud no puede exceder K ; por lo general se supone que esta partícula parte del reposo. La ecuación del movimiento de Newton produce

$$m \frac{d^2x}{dt^2} = F_A,$$

pero si L es una escala de longitud característica (quizá $L = |b - a|$), entonces, debido a que F_A y por lo tanto K tienen las dimensiones de masa x aceleración o masa por longitud ÷ tiempo al cuadrado, la cantidad $T = \sqrt{mL/K}$ debe tener la dimensión de tiempo. Entonces se puede hacer $m \frac{d^2x}{dt^2} = F_A$ adimensional escalando X y t con respecto a L y T , respectivamente, con $\hat{X} = X/L$, $\hat{t} = t/T$ obteniéndose

$$\frac{mL}{T^2} \frac{d^2\hat{X}}{d\hat{t}^2} = F_A \quad \text{o} \quad \frac{d^2\hat{X}}{d\hat{t}^2} = \frac{F_A}{K}.$$

La ecuación ahora es completamente adimensional porque la fuerza aplicada se escala con respecto a K en el lado derecho. Es conveniente escribir $u = F_A/K$, de modo que $|F_A| \leq K$ se convierte en

$$|u| \leq 1. \quad (2.31)$$

Nuestro problema de control es elegir u para minimizar el tiempo total

$$J = t_1 - t_0 = \int_{t_0}^{t_1} 1 dt. \quad (2.32)$$

De manera que se necesita una partícula gobernada por

$$\frac{d^2X}{dt^2} = u \quad (2.33)$$

y (2.31) que transferirá su desplazamiento desde $X^0 = X(t_0)$ en el tiempo inicial t_0 hasta $X^1 = X(t_1)$ en el tiempo final t_1 . Tanto la velocidad inicial como la final se suelen considerar cero. Tanto X como u dependen del tiempo t . Es preferible pensar que la partícula tiene un estado vectorial $x = (x_1, x_2)$ que consiste en su desplazamiento $x_1 = X$ y su velocidad $x_2 = \dot{X}$. Entonces $\dot{x}_1 = \dot{X} = x_2$ y $\dot{x}_2 = \ddot{X} = u$, y la ecuación diferencial ordinaria de segundo orden (2.33) puede reescribirse como dos EDOs de primer orden:

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = u. \quad (2.34)$$

Ahora nuestro problema es seleccionar u para minimizar el tiempo total (2.32) que necesita una partícula gobernada por (2.34) y (2.31) para ser transferida desde su estado inicial $x^0 = (x_1^0, x_2^0)$ a su estado final $x^1 = (x_1^1, x_2^1)$.

2.2.1 Condiciones necesarias para la optimización

Es conveniente definir conjuntos de índices

$$N^+ = \{1, 2, \dots, n\}, N = N^+ \cup \{0\} = \{0, 1, 2, \dots, n\}$$

junto con los subconjuntos U de \mathbb{R}^m y X de \mathbb{R}^n . Entonces, un problema de control más general que los considerados en la sección anterior es encontrar un vector m -dimensional de funciones de control continuas por partes

$$u(t) = (u_1(t), u_2(t), \dots, u_m(t)) \in U \subset \mathbb{R}^m$$

para transferir un vector de estado n -dimensional

$$x(t) = (x_1(t), x_2(t), \dots, x_n(t)) \in U \subset \mathbb{R}^n$$

de funciones suaves por partes desde su estado inicial

$$x(t_0) = x^0 = (x_1^0, x_2^0, \dots, x_n^0) \in X$$

a su estado final

$$x(t_1) = x^1 = (x_1^1, x_2^1, \dots, x_n^1) \in X$$

a lo largo de una trayectoria continua en X satisfaciendo

$$\dot{x}_i = f_i(x, u) \tag{2.35}$$

para todo $i \in N^+$; de tal manera que se minimice el costo funcional

$$J[u] = \int_{t_0}^{t_1} f_0(x, u) dt \tag{2.36}$$

donde f_i es una función suave de todos los argumentos $m + n$ para cada $i \in N$. Cuando sea necesario denotamos el vector (f_0, f_1, \dots, f_n) de todas estas funciones por f . Se puede demostrar que nuestros supuestos de suavidad garantizan la existencia de una solución continua única de las ecuaciones de estado (2.35) que satisfaga $x = x^0$ en $t = t_0$ para cualquier $u \in U$. Cualquier controlador de este tipo que sea capaz de transferir x desde x^0 en tiempo t_0 a x^1 en tiempo t_1 se dice que es admisible. Entre todos estos controladores buscamos uno que sea óptimo en el sentido de minimizar J . Por lo tanto, un controlador óptimo debe ser al menos admisible.

Aunque t_0 , x^0 y x^1 son fijos, en general t_1 no se especifica. Suponiendo que nuestras funciones de control son derivables, luego se procede por analogía como en la sección (2.1.2) para obtener un conjunto de condiciones necesarias. Luego reformulando esas condiciones necesarias en una forma que se aplicaría incluso a funciones de control continuas por partes. Pero entonces nuestras condiciones necesarias serán meramente una conjetura, por lo que se tendrá que

reestablecerlas mediante un enfoque completamente diferente. En consecuencia, sea $u^* = u^*(t)$ el controlador óptimo y sea $x^* = x^*(t)$ la trayectoria óptima asociada. Debido a que u^* es admisible, debemos tener

$$x^*(t_1) = x^1. \quad (2.37)$$

Sea u^* perturbada por otro controlador admisible $u(t) = u^*(t) + \delta u(t)$, donde $\|\delta u(t)\|$ es infinitesimalmente pequeño; y sea la trayectoria asociada $x(t) = x^*(t) + \delta x(t)$, donde $\|\delta x(t)\|$ también es infinitesimalmente pequeña, porque x depende continuamente de u , a través de f . Es decir,

$$x_i(t) = x_i^*(t) + \delta x_i(t) \quad (2.38)$$

para todo $i \in N^+$. Porque u es también admisible, x también debe alcanzar el estado objetivo x^1 . Sin embargo, el tiempo final no está especificado, por lo tanto, no hay ninguna razón para que x alcance el objetivo al mismo tiempo que t_1 como x^* . Más bien alcanzará el objetivo en el tiempo infinitesimalmente anterior o posterior $t_1 + \delta t$, donde $|\delta t|$ también es pequeño.

2.3 PRINCIPIO DEL MÁXIMO DE PONTRYAGIN (PMP)

Uno de los enfoques tradicionales para determinar la solución de problemas de control óptimo es un método indirecto basado en PMP en el que las condiciones de optimalidad, en forma de ecuaciones adjuntas, son continuas (Liberzon, 2012). Determinaremos las soluciones al problema de control óptimo de los manipuladores robóticos en este marco.

Definamos la función Hamiltoniana óptima como:

$$H(\mathbf{u}, \mathbf{x}, \boldsymbol{\lambda}) = \boldsymbol{\lambda}^T \dot{\mathbf{x}} - \gamma(\mathbf{u}) \quad (2.39)$$

donde \mathbf{x} es el vector de estados, $\boldsymbol{\lambda}$ es el vector de co-estados (también llamado estados adjuntos o variables conjugadas), \mathbf{u} es el vector de entradas de control y $\gamma(\mathbf{u})$ corresponde a la función costo seleccionada.

Los estados, para el caso de nuestros manipuladores, son cantidades continuas definidas como

$$\mathbf{x} = \begin{pmatrix} \mathbf{q}^T & \boldsymbol{\zeta}^T \end{pmatrix}^T = \begin{pmatrix} \mathbf{q}^T & \dot{\mathbf{q}}^T \end{pmatrix}^T$$

donde \mathbf{q} es el vector de coordenadas articulares, $\dot{\mathbf{q}}$ es el vector de velocidades articulares y $\boldsymbol{\zeta} = \dot{\mathbf{q}}$.

Estos estados se pueden utilizar para establecer la ecuación de movimiento del robot manipulador en forma de espacio de estados como

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\boldsymbol{\zeta}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\zeta} \\ \mathbf{M}(\mathbf{q})^{-1}[\mathbf{u} - \mathbf{G}(\mathbf{q}) - \mathbf{V}(\mathbf{q}, \boldsymbol{\zeta})] \end{bmatrix} \quad (2.40)$$

donde $\mathbf{M}(\mathbf{q})^{-1}$ representa el tensor de masas inverso; \mathbf{u} es el vector de fuerzas y pares aplicadas al robot; $\mathbf{G}(\mathbf{q})$ es un vector que agrupa las fuerzas que derivan del potencial gravitacional; y $\mathbf{V}(\mathbf{q}, \dot{\mathbf{q}})$ es un vector que agrupa los efectos centrífugos y de Coriolis.

Los co-estados o variables adjuntas

$$\boldsymbol{\lambda} = \begin{pmatrix} \mathbf{p}^T & \boldsymbol{\xi}^T \end{pmatrix}^T$$

son cantidades continuas con derivadas continuas (\mathbf{p} es adjunta a \mathbf{q} y $\boldsymbol{\xi}$ es adjunta a $\boldsymbol{\zeta}$).

El PMP nos proporciona las condiciones necesarias para obtener una ley de control óptima en forma de EDO de primer orden que no son lineales para el caso del control de manipuladores robóticos. Los controladores óptimos son tales que u_i maximizan el Hamiltoniano (2.39) cuando

$$\frac{\partial H}{\partial u_i} = 0, \quad \text{para } i = 1, \dots, n,$$

donde n denota el número de grados de libertad (GDL) del sistema.

En espacio de estados, las ecuaciones de movimiento del robot se pueden expresar en forma simpléctica como condiciones de EDO no lineales de primer orden

$$\dot{q}_i(t) = \frac{\partial H}{\partial p_i}, \quad \dot{\zeta}_i(t) = \frac{\partial H}{\partial \xi_i}, \quad \text{para } i = 1, \dots, n. \quad (2.41)$$

Los co-estados (o ecuaciones adjuntas) se dan a su vez también en forma simpléctica por

$$\dot{p}_i(t) = -\frac{\partial H}{\partial q_i}, \quad \dot{\xi}_i(t) = -\frac{\partial H}{\partial \zeta_i}, \quad \text{para } i = 1, \dots, n. \quad (2.42)$$

Las condiciones de optimalidad bajo el criterio seleccionado son establecidas por (2.42). De manera que los controladores óptimos requeridos para minimizar la función integral $J(\mathbf{u}) = \int_0^T \gamma(\mathbf{u}) dt$, con T de tiempo de finalización de tarea, son obtenidos de la solución del conjunto de $4n$ EDO no lineales (2.41) y (2.42).

2.4 ECUACIONES DE CONTROL COVARIANTES

La teoría en torno a las llamadas ecuaciones de control covariantes está enmarcada en la geometría de Riemann y está formulada con tensores, ver Grinfeld (2013) para más detalles. Se puede decir que un vector y una matriz son tensores de rango 1 y 2, respectivamente (Grinfeld, 2013). Los tensores generalizan estos objetos para dimensiones superiores. Existen tensores covariantes (con índices bajos) y contravariantes (con índices altos). Para una introducción breve y completa al cálculo tensorial, el lector puede consultar la obra realizada por Fleisch (2011). Para identificar los tensores, podemos resumir que:

- los elementos de un tensor de rango 1 se designan como a_i (covariantes) o a^i (contravariantes);
- los elementos de un tensor de rango 2 se designan como A_{ij} (covariantes) o A^{ij} (contravariantes);
- los elementos de un tensor de rango 3 se designan como A_{ijk} (covariantes) o A^{ijk} (contravariantes);
- existen objetos que combinan elementos covariantes con elementos contravariantes; tales como B_γ^α o bien Γ_{jk}^i .

Nuestro problema de control óptimo se puede resumir de la siguiente manera: se desea llevar el sistema robótico de un estado inicial a un estado final en un tiempo prescrito T de manera que la acción de control minimice un índice de desempeño seleccionado. Expresando el índice de desempeño como una función integral del tipo

$$J(u) = \int_0^T \gamma(u(t)) dt \quad (2.43)$$

donde u denota la variable de control, dicha acción de control puede restringirse minimizando (2.43) ya sea aplicando técnicas variacionales tradicionales o el Principio del Máximo de Pontryagin (PMP).

La función $\gamma = \gamma(u)$ depende de la señal de control y define el costo. Esta función costo se puede seleccionar para que sea convexa para facilitar la minimización de (2.43). Además, para preservar una estructura Riemanniana, γ debe elegirse para que sea invariante y preferiblemente esté compuesta por el tensor de masas $M(q)$ (que define una métrica Riemanniana).

Al elegir el siguiente índice de desempeño

$$J(u) = \int_0^T \frac{1}{2} M_{kl}(q) u^k u^l dt, \quad (2.44)$$

la variable de control óptima evoluciona según la siguiente ecuación:

$$\left(\frac{d^2 u}{dt^2} \right)_j + R_{klj}^i \dot{q}^k \dot{q}^l u_i + (\nabla_{jk}^2 V) u^k = 0. \quad (2.45)$$

La ecuación (2.45) es un resultado que se estableció por primera vez en el marco del cálculo clásico de variaciones en Rojas Quintero *et al.* (2013). Más tarde se estableció en el marco del Principio Máximo de Pontryagin en Dubois *et al.* (2015), lo que demuestra aún más la equivalencia de ambas técnicas; y recientemente se aplicó experimentalmente al control óptimo de un manipulador robótico en Rojas-Quintero *et al.* (2021a). Nos referiremos a (2.45) como las *ecuaciones de control covariantes*.

Equipado con (2.45), el problema de control óptimo es el de resolver un sistema acoplado de EDO no lineales de segundo orden que involucra las ecuaciones de movimiento del sistema y las ecuaciones de control covariantes:

$$\begin{cases} \ddot{q}^j + \Gamma_{kl}^j \dot{q}^k \dot{q}^l + M^{jl} \partial_l V = u^j \\ \left(\frac{d^2 u}{dt^2} \right)_j + (\nabla_{jk}^2 V) u^k = R_{kjl}^i \dot{q}^k \dot{q}^l u_i, \end{cases} \quad (2.46)$$

con condiciones iniciales y finales seleccionadas para q y \dot{q} . La primera ecuación de (2.46) es contravariante en los estados q , y la segunda ecuación de (2.46) es covariante en la variable de control u . Este conjunto de ecuaciones se puede resolver con métodos de disparo apropiados, integradores basados en Runge-Kutta (Almuslimani y Vilmart, 2021) o métodos basados en elementos finitos como el presentado por Rojas-Quintero *et al.* (2021b).

2.5 NAVEGACIÓN DE ROBOTS

Cuando se estudia la geometría de los brazos de un robot, se desarrollan soluciones para los problemas de cinemática directa e inversa⁶. Las soluciones a

⁶El problema de la cinemática es describir el movimiento del manipulador sin tener en cuenta las fuerzas y los momentos de torsión que provocan el movimiento. La descripción cinemática es, por tanto, geométrica. Entonces la cinemática directa consiste en determinar la posición y orientación del órgano terminal dados los valores de las variables articulares del robot. La cinemática inversa es determinar los valores de las variables articulares, dada la posición y orientación del órgano terminal.

estos problemas dependen únicamente de la geometría intrínseca del robot y no reflejan ninguna restricción impuesta por el espacio de trabajo⁷ en el que opera el robot (Spong *et al.*, 2005). El problema de navegación de robots debe tomar en cuenta la posibilidad de colisión entre el robot y los objetos en el espacio de trabajo. Por lo que se debe abordar este problema generando, en primera instancia, caminos o rutas libres de colisiones.

De hecho, el problema de navegación se separa en tres grandes áreas:

- (a) Planificación de caminos o rutas
- (b) Generación de trayectorias
- (c) Control

El problema de la planificación de rutas se encuentra entre los más difíciles de la informática. La complejidad computacional del algoritmo completo (es decir, algoritmo que encuentra una solución en caso de existir) de planificación de rutas más conocido crece exponencialmente con el número de grados de libertad del robot. Por esta razón, para los sistemas de robots con más de unos pocos grados de libertad, en la práctica no se utilizan algoritmos completos. La planificación de rutas proporciona una descripción geométrica del movimiento del robot, pero no especifica ningún aspecto dinámico del movimiento. Por ejemplo, ¿cuáles deberían ser las velocidades y aceleraciones de las articulaciones al ir por la ruta deseada? Estas preguntas son abordadas por un generador de trayectorias. El generador de trayectorias calcula una función $q(t)$ que especifica completamente el movimiento deseado del robot a medida que avanza por el camino predeterminado. Finalmente, el control es el área que se encarga de garantizar que las etapas anteriores (caminos y trayectorias) se cumplan.

2.6 TIEMPO DE CÁLCULO Y ESTABILIDAD NUMÉRICA

El tiempo de cálculo (también llamado “tiempo de ejecución”) es el tiempo necesario para realizar un proceso de cálculo en un sistema de cómputo. Al representar un cálculo como una secuencia de aplicaciones de reglas, el tiempo de cálculo es proporcional al número de aplicaciones de reglas (Rowland, 2021).

Las ecuaciones rígidas son problemas para los que los métodos explícitos no funcionan (Hairer y Wanner, 1996). Los métodos explícitos e implícitos son enfoques utilizados en el análisis numérico para obtener aproximaciones numéricas

⁷El espacio de trabajo de un manipulador es el volumen total barrido por el órgano terminal cuando el manipulador ejecuta todos los movimientos posibles. El espacio de trabajo está limitado por la geometría del manipulador, así como por las limitaciones mecánicas de las articulaciones.

a las soluciones de ecuaciones diferenciales ordinarias y parciales dependientes del tiempo, como se requiere en las simulaciones por computadora de procesos físicos. Los métodos explícitos calculan el estado de un sistema en un momento posterior a partir del estado del sistema en el momento actual, mientras que los métodos implícitos encuentran una solución al resolver una ecuación que involucra tanto el estado actual del sistema como el posterior (Ascher *et al.*, 1997).

En los algoritmos numéricos para ecuaciones diferenciales, la preocupación es el crecimiento de errores de redondeo o de pequeñas fluctuaciones en los datos iniciales que podrían causar una gran desviación de la respuesta final de la solución exacta.

Algunos algoritmos numéricos pueden amortiguar las pequeñas fluctuaciones (errores) en los datos de entrada; otros podrían magnificar tales errores. Los cálculos que se puede probar que no magnifican los errores de aproximación se denominan numéricamente estables. Una de las tareas comunes del análisis numérico es intentar seleccionar algoritmos que sean robustos, es decir, que no produzcan un resultado muy diferente para cambios muy pequeños en los datos de entrada.

Un fenómeno opuesto es la inestabilidad. Por lo general, un algoritmo implica un método aproximado y, en algunos casos, se podría probar que el algoritmo se acercaría a la solución correcta en algún límite (cuando se utilizan números reales, no números de punto flotante). Incluso en este caso, no hay garantía de que convergerá a la solución correcta, porque los errores de truncamiento o redondeo de punto flotante pueden magnificarse, en lugar de amortiguarse, haciendo que la desviación de la solución exacta crezca exponencialmente (Engeln-Müllges y Uhlig, 1996).

3

CONTROL ÓPTIMO APLICADO A SISTEMAS ROBÓTICOS

En este capítulo se aprovecharán los fundamentos introducidos en el capítulo 2, en específico, el PMP. Se plantea enmarcar el trabajo de esta tesis en la temática de la Robótica Avanzada. La idea es realizar el control óptimo de robots manipuladores con al menos dos esquemas de control óptimo distintos para comparar su desempeño. Por un lado se aplicará el control óptimo basado en ecuaciones de control covariantes introducido en Rojas-Quintero *et al.* (2021a), y se tomará como referencia. Se realizarán comparaciones en simulación con dos manipuladores. El primero, de un solo GDL servirá para dar a comprender el funcionamiento de los esquemas de control óptimo presentados sin lidiar con las especificidades de un sistema dinámico complejo. El segundo, de 2 GDL, es un robot de tipo serial que posee todas las no-linealidades y efectos inerciales característicos de cualquier sistema robótico avanzado.

3.1 DINÁMICA DEL ROBOT

Para derivar el modelo dinámico de los robots bajo estudio se seguirán los procedimientos estándar que se pueden encontrar en Spong *et al.* (2005) y en Kelly *et al.* (2005). Los manipuladores robóticos que estudiamos son del tipo serial en el que un actuador se coloca entre dos eslabones consecutivos e induce un movimiento rotacional. El manipulador robótico se rige por ecuaciones diferenciales ordinarias no lineales de movimiento de segundo orden de la siguiente forma:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \mathbf{u} \quad (3.1)$$

donde $\mathbf{M}(\mathbf{q})$ es el tensor de masas del sistema que es una forma bilineal simétrica definida positiva dependiente de las configuraciones \mathbf{q} (las coordenadas articulares). Para el manipulador de 1 GDL, es un solo componente escalar, pero para el manipulador de 2 GDL, es de rango 2. $\mathbf{V}(\mathbf{q}, \dot{\mathbf{q}})$ es el vector que contiene los efectos centrífugos y de Coriolis. $\mathbf{G}(\mathbf{q})$ representa la acción de la gravedad

VARIABLE	DESCRIPCIÓN
m_i	Masa del eslabón i
I_i	Momento de inercia del eslabón i
a_i	Distancia entre el centro de giro i y el centro de masa i
l_i	Longitud del eslabón i
q_i	Variable de configuración i (posición angular)

Tabla 3.1: Convención de nomenclatura de un manipulador de n GDL.

de la Tierra en el sistema y \mathbf{u} es el vector de fuerzas y pares de entrada (la señal de control).

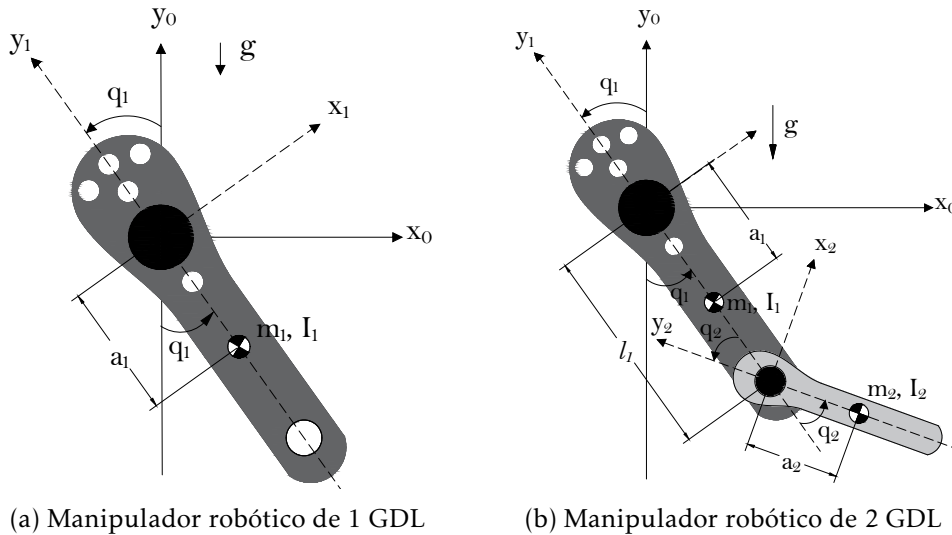


Figura 3.1: Esquema de los manipuladores robóticos utilizados: (a) manipulador robótico de 1 GDL; (b) Manipulador robótico de 2 GDL. Los parámetros (m_i, I_i, a_i, l_i, q_i) denotan respectivamente la i -ésima: masa del eslabón; momento de inercia del eslabón; distancia del centro de giro al centro de masa; longitud del eslabón; variable de configuración.

3.2 DINÁMICA DE UN ROBOT MANIPULADOR DE 1 GDL

Se utiliza el manipulador de 1 GDL mostrado en la Figura 3.1(a). La descripción de las variables que aparecen en la Figura 3.1(a) se muestran de manera

generalizada en la Tabla 3.1. Se presentará el estudio cinemático y dinámico a continuación.

Para este manipulador, establecemos los siguientes valores: $m_1 = 2.4 \text{ kg}$; $a_1 = 0.4 \text{ m}$; $I_1 = 1 \text{ kg m}^2$. Para este manipulador robótico de 1 GDL, la ecuación de movimiento la obtenemos retomando el Principio de Mínima Acción (ver ecuación (2.5)), aplicamos la EEL (2.4) al Lagrangiano $L(q, \dot{q})$ del sistema definido como la diferencia entre la energía cinética $K(q, \dot{q})$ y la energía potencial $V(q)$. Así, $L(q, \dot{q}) = K(q, \dot{q}) - V(q)$ y

$$\frac{d}{dt} \frac{\partial L(q, \dot{q})}{\partial \dot{q}_1} - \frac{\partial L(q, \dot{q})}{\partial q_1} = u_1$$

donde u_1 es el par aplicado a la articulación 1. Tomando $K(q, \dot{q}) = \frac{1}{2} m_1 v^2$, $V(q) = m_1 g h$; donde m_1 es la masa del eslabón, v es la velocidad del centro de masa, g es la constante gravitacional y h es la altura del centro de masa del manipulador. Entonces para este manipulador se tiene que:

$$K(q, \dot{q}) = \frac{1}{2} m_1 v^T v + \frac{1}{2} I \dot{q}_1^2.$$

Tomando $v = \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \end{bmatrix}$, se tiene que

$$\begin{aligned} x_1 &= a_1 \text{sen}(q_1) \Rightarrow \dot{x}_1 = a_1 \cos(q_1) \dot{q}_1 \\ y_1 &= a_1 \cos(q_1) \Rightarrow \dot{y}_1 = -a_1 \text{sen}(q_1) \dot{q}_1. \end{aligned}$$

Así

$$K(q, \dot{q}) = \frac{1}{2} m_1 a_1^2 \dot{q}_1^2 + \frac{1}{2} I \dot{q}_1^2,$$

y la energía potencial $V(q)$ es

$$V(q) = -m_1 g a_1 \cos(q_1);$$

con lo cual $L(q, \dot{q})$ se expresa como

$$L(q, \dot{q}) = K(q, \dot{q}) - V(q) = \frac{1}{2} m_1 a_1^2 \dot{q}_1^2 + \frac{1}{2} I \dot{q}_1^2 + a_1 g m_1 \cos(q_1).$$

Ahora se puede establecer los componentes de la EEL aplicada a este Lagrangiano:

$$\frac{\partial L(q, \dot{q})}{\partial q_1} = -a_1 m_1 g \text{sen}(q_1),$$

$$\frac{\partial L(q, \dot{q})}{\partial \dot{q}_1} = m_1 a_1^2 \dot{q}_1 + I_1 \dot{q}_1$$

y

$$\frac{d}{dt} \frac{\partial L(q, \dot{q})}{\partial \dot{q}_1} = m_1 a_1^2 \ddot{q}_1 + I_1 \ddot{q}_1$$

que llevan a la ecuación que rige el movimiento de este sistema

$$M \ddot{q}_1 + a_1 g m_1 \text{sen}(q_1) = u_1 \quad (3.2)$$

donde $M = m_1 a_1^2 + I_1 = 1.384 \text{ kg m}^2$ (único componente del tensor de masas, que es constante). Esta ecuación de movimiento se utilizará dentro del esquema de control óptimo ya que se busca minimizar la funcional integral (índice de desempeño)

$$J(u) = \int_a^b \gamma(u(q, \dot{q}, \ddot{q}, t)) dt$$

bajo las condiciones

$$\begin{aligned} q(a) &= A, & q(b) &= B \\ \dot{q}(a) &= 0, & \dot{q}(b) &= 0. \end{aligned} \quad (3.3)$$

La funcional costo γ será semejante a la empleada en Rojas-Quintero *et al.* (2021a) y que pone en juego el tensor de masas del sistema. Las constantes a y b son los instantes de inicio y fin del movimiento deseado; A y B son las posiciones angulares al inicio y al final del movimiento. Nótese que se utilizarán velocidades nulas al inicio y al final del movimiento.

3.3 DINÁMICA DE UN ROBOT MANIPULADOR DE 2 GDL

Adicionalmente se extenderá este estudio a un manipulador de 2 GDL ya que este mecanismo presenta todas las no-linealidades y efectos de inercia representativos de un sistema robótico avanzado. Se utilizará el manipulador de 2 GDL mostrado en la Figura 3.1(b).

Para este robot manipulador, la ecuación (3.1) conduce a dos ecuaciones de movimiento.

Retomando el Principio de Mínima Acción (ver ecuación (2.5)), se aplica la EEL (2.4) al Lagrangiano $L(q, \dot{q})$ del sistema,

$$\frac{d}{dt} \frac{\partial L(q, \dot{q})}{\partial \dot{q}_i} - \frac{\partial L(q, \dot{q})}{\partial q_i} = u_i \text{ con } i = 1, 2.$$

Cada uno de los dos eslabones de este mecanismo tiene una energía cinética dada por

$$K_1(q, \dot{q}) = \frac{1}{2} m_1 v_1^T v_1 + \frac{1}{2} I_1 \dot{q}_1^2$$

para el primer eslabón, y

$$K_2(q, \dot{q}) = \frac{1}{2} m_2 v_2^T v_2 + \frac{1}{2} I_2 [\dot{q}_1 + \dot{q}_2]^2$$

para el segundo eslabón, en donde v_1 y v_2 son las velocidades del centro de masa del eslabón 1 y 2, respectivamente. Enseguida se tiene que

$$x_1 = a_1 \sin(q_1) \Rightarrow \dot{x}_1 = a_1 \cos(q_1) \dot{q}_1$$

$$y_1 = -a_1 \cos(q_1) \Rightarrow \dot{y}_1 = a_1 \sin(q_1) \dot{q}_1$$

$$x_2 = l_1 \sin(q_1) + a_2 \sin(q_1 + q_2) \Rightarrow \dot{x}_2 = l_1 \cos(q_1) \dot{q}_1 + a_2 \cos(q_1 + q_2) [\dot{q}_1 + \dot{q}_2]$$

$$y_2 = -l_1 \cos(q_1) - a_2 \cos(q_1 + q_2) \Rightarrow \dot{y}_2 = l_2 \sin(q_1) \dot{q}_1 + a_2 \sin(q_1 + q_2) [\dot{q}_1 + \dot{q}_2].$$

La energía cinética del primer eslabón está dada por

$$K_1(q, \dot{q}) = \frac{1}{2} m_1 a_1^2 \dot{q}_1^2 + \frac{1}{2} I_1 \dot{q}_1^2.$$

Para el segundo eslabón, el producto escalar, el cuadrado de su velocidad lineal, está dado por

$$\begin{aligned} v_2^T v_2 &= [\dot{x}_2, \dot{y}_2] \begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = \dot{x}_2^2 + \dot{y}_2^2 \\ &= \{l_1 \cos(q_1) \dot{q}_1 + a_2 \cos(q_1 + q_2) [\dot{q}_1 + \dot{q}_2]\}^2 \\ &\quad + \{-l_1 \sin(q_1) \dot{q}_1 - a_2 \sin(q_1 + q_2) [\dot{q}_1 + \dot{q}_2]\}^2 \\ &= a_2^2 + l_1^2 + 2a_2 l_1 \cos(q_2) \dot{q}_1^2 + 2a_2(a_2 + l_1 \cos(q_2)) \dot{q}_1 \dot{q}_2 + a_2^2 \dot{q}_2^2. \end{aligned}$$

Con lo cual se obtiene la expresión de la energía cinética del segundo eslabón

$$\begin{aligned} K_2(q, \dot{q}) &= \frac{1}{2} m_2 \{a_2^2 + l_1^2 + 2a_2 l_1 \cos(q_2)\} \dot{q}_1^2 \\ &\quad + 2a_2(a_2 + l_1 \cos(q_2)) \dot{q}_1 \dot{q}_2 + a_2^2 \dot{q}_2^2 + \frac{1}{2} I_2 [\dot{q}_1 + \dot{q}_2]^2. \end{aligned}$$

Las energías potenciales son respectivamente

$$V_1(q) = -m_1 g a_1 \cos(q_1),$$

$$V_2(q) = -m_2 g [l_1 \cos(q_1) + a_2 \cos(q_1 + q_2)].$$

Al tomar estas energías cinéticas y potenciales, se puede construir el Lagrangiano del sistema

$$\begin{aligned}
 L(q, \dot{q}) &= K_1(q, \dot{q}) + K_2(q, \dot{q}) - (V_1(q) + V_2(q)) \\
 &= a_1 g m_1 \cos(q_1) + g m_2 (l_1 \cos(q_1) + a_2 \cos(q_1 + q_2)) + \frac{1}{2} I_1 \dot{q}_1^2 \\
 &\quad + \frac{1}{2} a_1^2 m_1 \dot{q}_1^2 + \frac{1}{2} I_2 (\dot{q}_1 + \dot{q}_2)^2 + \frac{1}{2} m_2 ((a_2^2 + l_1^2 + 2a_2 l_1 \cos(q_2)) \dot{q}_1^2 \\
 &\quad + 2a_2 (a_2 + l_1 \cos(q_2)) \dot{q}_1 \dot{q}_2 + a_2^2 \dot{q}_2^2);
 \end{aligned}$$

sobre el cual se pueden efectuar las derivaciones requeridas por la EEL y que se exponen a continuación:

$$\begin{aligned}
 \frac{\partial L(q, \dot{q})}{\partial \dot{q}_1} &= (I_1 + I_2 + a_1^2 m_1 + (a_2^2 + l_1^2) m_2 + 2a_2 l_1 m_2 \cos(q_2)) \dot{q}_1 \\
 &\quad + (I_2 + a_2^2 m_2 + a_2 l_1 m_2 \cos(q_2)) \dot{q}_2; \\
 \frac{d}{dt} \frac{\partial L(q, \dot{q})}{\partial \dot{q}_1} &= -2a_2 l_1 m_2 \sin(q_2) \dot{q}_1 \dot{q}_2 - a_2 l_1 m_2 \sin(q_2) \dot{q}_2^2 \\
 &\quad + (I_1 + I_2 + a_1^2 m_1 \cos(q_2)) \ddot{q}_1 + (a_2^2 + l_1^2) m_2 + 2a_2 l_1 m_2 \\
 &\quad + (I_2 + a_2^2 m_2 + a_2 l_1 m_2 \cos(q_2)) \ddot{q}_2; \\
 \frac{\partial L(q, \dot{q})}{\partial \dot{q}_2} &= a_2 l_1 m_2 \cos(q_2) \dot{q}_1 + (I_2 + a_2^2 m_2) (\dot{q}_1 + \dot{q}_2); \\
 \frac{d}{dt} \frac{\partial L(q, \dot{q})}{\partial \dot{q}_2} &= -a_2 l_1 m_2 \sin(q_2) \dot{q}_1 \dot{q}_2 + a_2 l_1 m_2 \cos(q_2) \ddot{q}_1 \\
 &\quad + (I_2 + a_2^2 m_2) (\ddot{q}_1 + \ddot{q}_2); \\
 \frac{\partial L(q, \dot{q})}{\partial q_1} &= -g((a_1 m_1 + l_1 m_2) \sin(q_1) + a_2 m_2 \sin(q_1 + q_2)); \\
 \frac{\partial L(q, \dot{q})}{\partial q_2} &= -a_2 m_2 (g \sin(q_1 + q_2) + l_1 \sin(q_2) \dot{q}_1 (\dot{q}_1 + \dot{q}_2)).
 \end{aligned}$$

Con estos desarrollos, se llega finalmente a las expresiones que rigen el movimiento de este sistema:

$$\begin{aligned}
 & g(m_1 a_1 + m_2 l_1) \sin(q_1) + m_2 g a_2 \sin(q_1 + q_2) - 2a_2 l_1 m_2 \sin(q_2) \dot{q}_1 \dot{q}_2 \\
 & - a_2 l_1 m_2 \sin(q_2) \dot{q}_2^2 + (I_1 + I_2 + a_1^2 m_1 + (a_2^2 + l_1^2) m_2 + 2a_2 l_1 m_2 \cos(q_2)) \ddot{q}_1 \\
 & \quad + (I_2 + a_2^2 m_2 + a_2 l_1 m_2 \cos(q_2)) \ddot{q}_2 = u_1 \\
 & a_2 g m_2 \sin(q_1 + q_2) + a_2 l_1 m_2 \sin(q_2) \dot{q}_1^2 + a_2 l_1 m_2 \cos(q_2) \dot{q}_1 \dot{q}_2 \\
 & \quad + (I_2 + a_2^2 m_2) (\ddot{q}_1 + \ddot{q}_2) = u_2
 \end{aligned} \tag{3.4}$$

En forma matricial compacta se tiene

$$\begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} V_1(q, \dot{q}) \\ V_2(q, \dot{q}) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} G_1(q) \\ G_2(q) \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

donde

$$\begin{aligned}
 M_{11} &= I_1 + I_2 + a_1^2 m_1 + (a_2^2 + l_1^2) m_2 + 2a_2 l_1 m_2 \cos(q_2) \\
 M_{12} &= I_2 + a_2^2 m_2 + a_2 l_1 m_2 \cos(q_2) \\
 M_{21} &= I_2 + a_2^2 m_2 + a_2 l_1 m_2 \cos(q_2) \\
 M_{22} &= I_2 + a_2^2 m_2 \\
 V_1 &= -m_2 l_1 a_2 \sin(q_2) (2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2); \\
 V_2 &= m_2 l_1 a_2 \sin(q_2) \dot{q}_1^2; \\
 G_1 &= g(a_1 m_1 + l_1 m_2) \sin(q_1) + m_2 g a_2 \sin(q_1 + q_2) \\
 G_2 &= m_2 g a_2 \sin(q_1 + q_2).
 \end{aligned}$$

Como con el manipulador de 1 GDL anteriormente expuesto, estas ecuaciones se utilizarán dentro de los esquemas de control óptimo.

3.4 METODOLOGÍA DE CONTROL ÓPTIMO

3.4.1 Índice de desempeño y función costo

Se desea llevar el sistema articulado robótico de un estado de posiciones y velocidades iniciales $(q(0), \dot{q}(0))$ a un estado final $(q(T), \dot{q}(T))$ en un tiempo prescrito T . Las intensidades de par u requeridas para lograr esto, se pueden restringir u obtener minimizando una función integral del tipo

$$J(\mathbf{u}) = \int_0^T \gamma(\mathbf{u}) dt. \tag{3.5}$$

La integral funcional (3.5) se considera el Índice de Desempeño (ID), donde γ es la función costo seleccionada. Esta función costo se debe determinar de tal manera que (3.5) puede minimizarse eficazmente. Por lo tanto, generalmente se elige convexa. En este caso, entonces dependerá de los pares del robot \mathbf{u} .

Se ha notado, en la literatura, que cuando se busca minimizar las intensidades de par, γ generalmente se elige como:

$$\gamma_A = \frac{1}{2} \mathbf{u}^T \mathbf{A} \mathbf{u} \quad (3.6)$$

donde \mathbf{A} es una matriz de ponderación de coeficiente constante y diagonal (Eriksson y Nordmark, 2010; Ghasemi *et al.*, 2012; Kaphle y Eriksson, 2008); a menudo tomada como la identidad (Asgari y Nikoobin, 2020; Eriksson, 2007; Mirz *et al.*, 2018; Nikoobin y Moradi, 2011; Vezvari *et al.*, 2020). Se realizaron simulaciones utilizando esta función costo (3.6). Sin perder la generalidad, se centrará en el caso donde \mathbf{A} es la identidad.

Ahora, como se ha propuesto en Dubois *et al.* (2015); Rojas-Quintero *et al.* (2021a,b), también se utilizará la siguiente función costo

$$\gamma_M = \frac{1}{2} \mathbf{u}^T \mathbf{M}^{-1} \mathbf{u}. \quad (3.7)$$

Esta función costo hace que $J(\mathbf{u})$ sea invariante ante un cambio de coordenadas. Se realizaron simulaciones con esta función costo para cuantificar la mejora en comparación con γ_A .

3.5 IMPACTO DE LA FUNCIÓN COSTO EN LAS CONDICIONES DE OPTIMIZACIÓN

Centrándose en el caso de un manipulador robótico de 1 GDL para ilustrar el impacto de la función costo propuesta en las condiciones de optimización. En este caso, el tensor de masas tiene una única componente constante. Por lo tanto, γ_A y γ_M conducen exactamente al mismo sistema de 4 EDO que describen los controladores y estados óptimos.

Sin embargo, cuando se toma γ_A , entonces la condición de optimalidad de Pontryagin, es decir, el Hamiltoniano se maximiza cuando

$$u_1 = \frac{\xi_1}{M}. \quad (3.8)$$

Por el contrario, cuando se toma γ_M , entonces se cumple la condición de optimalidad de Pontryagin cuando

$$u_1 = \xi_1.$$

Esto significa que cuando los componentes del tensor de masa inverso están involucrados en la función costo de (3.5), el segundo co-estado (o parámetro adjunto), describe directamente los pares de torsión del sistema, como se presenta en Dubois *et al.* (2015). Esto es significativo porque cuando γ_M se utiliza, no es necesario reconstruir la variable de control después del cálculo.

Es importante señalar que si se utiliza γ_A (3.6) para el control óptimo de un manipulador robótico de n GDL, la expresión de pares inevitablemente se vuelve mucho más complicada que (3.8). Por el contrario, siempre que se utiliza γ_M (3.7) para el control óptimo de un manipulador robótico de n GDL,

$$u_i = \xi_i, \quad \text{para } i = 1, \dots, n. \quad (3.9)$$

4

RESULTADOS DE SIMULACIÓN

4.1 METODOLOGÍA DE EVALUACIÓN

Para evaluar el impacto de cada función costo γ_A (3.6) y γ_M (3.7) en el control óptimo de estos manipuladores robóticos de 1 GDL y 2 GDL, hemos seguido una metodología específica que se articula en los pasos siguientes.

1. Resolver el sistema de $4n$ EDO (2.41) y (2.42) para diferentes valores de T (a partir de 0.5 s con incrementos de 0.1 s), tomando los mismos valores de frontera para determinar cada solución.
2. Calcular el par RMS para cada solución como

$$u_{\text{RMS}} = \sqrt{\frac{\int_0^T \sum_{i=1}^n u_i^2}{T}}.$$

3. Calcular la potencia RMS para cada solución como

$$P_{\text{RMS}} = \sqrt{\frac{\int_0^T \sum_{i=1}^n (u_i \dot{q}_i)^2}{T}}.$$

4. Determinar el tiempo de cálculo del CPU para cada trayectoria.

Se mencionan ahora algunos detalles de los pasos anteriores.

- Las soluciones se obtuvieron con el solucionador NDSolve EDO de Mathematica. Este solucionador en particular se puede configurar para que funcione con una amplia gama de métodos de solución. Se han obtenido mejores resultados configurándolo para operar con un método de integración de tiempo Runge-Kutta (Rojas-Quintero *et al.*, 2021a). Por lo tanto, todas las soluciones se proporcionaron con este método.

- El tiempo de computación del CPU denota el tiempo total que le tomó a la computadora obtener una solución. Esto se determinó con la función incorporada AbsoluteTiming de Mathematica.
- Todas las soluciones se calcularon con un procesador Intel® Core™ i7-8550U de 4 núcleos (1.8 GHz, 8 MB de la memoria caché).

4.2 CONTROL ÓPTIMO DE UN MANIPULADOR ROBÓTICO DE 1 GDL

4.2.1 PMP aplicado al manipulador de 1 GDL

Ahora, de la ecuación que rige el movimiento del manipulador de 1 GDL (3.2) se despeja la aceleración, obteniendo:

$$\ddot{\zeta}_1 = \frac{u_1 - m_1 g a_1 \sin(q_1)}{M} = \ddot{q}_1.$$

Ver el espacio de estados (2.40). Con la ecuación anterior y la función costo γ_A definida por

$$\gamma_A = \frac{u_1^2}{2},$$

ahora se puede establecer el Hamiltoniano $H(\lambda, x, u)$ (2.39) con $i = 1$

$$\begin{aligned} H(\lambda, x, u) &= \frac{u_1^2}{2} + p_1 \dot{q}_1 + \xi_1 \ddot{\zeta}_1 \\ &= \frac{u_1^2}{2} + p_1 \dot{q}_1 + \xi_1 \frac{u_1 - m_1 g a_1 \sin(q_1)}{M}. \end{aligned} \quad (4.1)$$

Cuando se aplica el PMP ($\frac{\partial H}{\partial u_1} = 0$) a (4.1) y al despejar u_1 que ahora será función de los estados se obtiene

$$u_1 = -\frac{\xi_1}{M}. \quad (4.2)$$

Sustituyendo (4.2) en (4.1) se tiene el siguiente espacio de estados

$$\begin{cases} \dot{q}_1 &= \frac{\partial H}{\partial p_1} \\ \dot{\zeta}_1 &= \frac{\partial H}{\partial \xi_1} = -\{m_1 g a_1 M \sin(q_1) + \xi_1\} M^{-2} \\ \dot{p} &= -\frac{\partial H}{\partial q_1} = a_1 g m_1 \cos(q_1) \xi_1 \\ \dot{\xi} &= -\frac{\partial H}{\partial \zeta_1} = -p_1. \end{cases} \quad (4.3)$$

Se puede observar del espacio de estados que se pasa de un sistema de EDO de segundo orden a un sistema de EDO de primer orden. Con los valores de la Tabla 4.1 se obtiene:

$$M = m_1 a_1^2 + I_1 = 1.384 \text{ kg m}^2.$$

VALOR
$m_1 = 2.4 \text{ kg}$
$I_1 = 1.0 \text{ kg m}^2$
$a_1 = 0.4 \text{ m}$
$g = 9.81 \text{ m s}^{-2}$

Tabla 4.1: Valores del manipulador de 1 GDL.

4.2.2 PMP aplicado al manipulador de 1 GDL (usando el tensor de masas)

Ahora, de la ecuación que rige el movimiento del manipulador de 1 GDL (3.2), se despeja la aceleración y se obtiene :

$$\ddot{q}_1 = \frac{u_1 - m_1 g a_1 \text{sen}(q_1)}{M} = \dot{\zeta}_1.$$

Ver el espacio de estados (2.40). Con la ecuación anterior y la función costo γ_M definida por

$$\gamma_M = \frac{u_1^2}{2M},$$

ahora se puede establecer el Hamiltoniano $H(\lambda, x, u)$ (2.39) con $i = 1$

$$\begin{aligned} H(\lambda, x, u) &= \frac{u_1^2}{2M} + p_1 \dot{q}_1 + \xi_1 \dot{\zeta}_1 \\ &= \frac{u_1^2}{2M} + p_1 \dot{q}_1 + \xi_1 \frac{u_1 - m_1 g a_1 \text{sen}(q_1)}{M}. \end{aligned} \quad (4.4)$$

Al aplicar el PMP ($\frac{\partial H}{\partial u_1} = 0$) a (4.4) y despejando u_1 que ahora será función de los estados se obtiene

$$u_1 = -\xi_1. \quad (4.5)$$

Sustituyendo (4.5) en (4.1) se tiene el siguiente espacio de estados

$$\begin{cases} \dot{q}_1 &= \frac{\partial H}{\partial p_1} \\ \dot{\zeta}_1 &= \frac{\partial H}{\partial \xi_1} = -\{m_1 g a_1 M \sin(q_1) + \xi_1\} M^{-1} \\ \dot{p} &= -\frac{\partial H}{\partial q_1} = a_1 g m_1 \cos(q_1) \xi_1 \\ \dot{\xi} &= -\frac{\partial H}{\partial \zeta_1} = -p_1. \end{cases} \quad (4.6)$$

La Figura 4.1 muestra tres conjuntos de curvas. Siguiendo el paso 1) de la metodología de la sección 4.1, obtuvimos soluciones al conjunto de 4 EDO compuesto por (2.41) y (2.42), es decir, para γ_A (3.6) el espacio de estados (4.3) y para γ_M (3.7) el espacio de estados (4.6) en el PMP, así como para ECV con valores diferentes de tiempo de trayectoria prescrito hasta $T = 20$ s. Las soluciones se obtienen fijando los siguientes valores de frontera

$$\begin{cases} \mathbf{q}(0) = q(0) = 0 \text{ rad} \\ \dot{\mathbf{q}}(0) = \zeta(0) = 0 \text{ rad s}^{-1} \\ \mathbf{q}(T) = q(T) = 0.8 \text{ rad} \\ \dot{\mathbf{q}}(T) = \zeta(T) = 0 \text{ rad s}^{-1}. \end{cases} \quad (4.7)$$

Tenga en cuenta que la estabilidad del solucionador no se vio afectada por el cambio en la función costo para este caso unidimensional. Los valores de par RMS de las trayectorias se obtuvieron siguiendo los pasos 1) y 2) de la metodología de la sección 4.1. Estos se muestran en la Figura 4.1(a) para ambas funciones costo γ_A (3.6) y γ_M (3.7) en el PMP, así como para ECV. Para este caso unidimensional, se puede ver que con ECV se obtienen los mismos valores de par RMS que con las funciones costo γ_A (3.6) y γ_M (3.7) en el PMP. Esto último debido a que el tensor de masas tiene un componente constante único. Los valores de potencia RMS de las trayectorias se obtuvieron siguiendo los pasos 1) y 3) de la metodología de la sección 4.1. Esto se muestra en la Figura 4.1(b) para el método de ECV y las funciones costo γ_A (3.6) y γ_M (3.7) en el PMP. Al igual que con los pares RMS, los valores obtenidos son iguales con ECV y con ambas funciones costo en el PMP. El tiempo de cálculo del CPU para cada trayectoria se obtuvo siguiendo los pasos 1) y 4) de la metodología de la sección 4.1. La Figura 4.1(c) muestra estos tiempos para valores diferentes del tiempo de trayectoria prescrito T . Tenga en cuenta que los valores obtenidos con ECV y con la función costo γ_M (3.7) en el PMP son muy similares hasta $T = 9.5$ s y con la función costo γ_A (3.6) en el PMP se mantiene esta similitud hasta $T = 13$ s. Posteriormente, los tiempos del CPU aumentan cuando se usa γ_M (3.7) en el PMP. Por lo tanto, γ_M

(3.7) no aumenta el rendimiento de este manipulador robótico de 1 GDL porque el único componente del tensor de masas es constante.

A continuación, considere la Figura 4.2 que muestra las curvas de posición y pares del manipulador robótico de 1 GDL durante su movimiento óptimo. Las curvas 4.2(a) muestran movimientos para $T = 0.5$ s, $T = 2.5$ s, $T = 5$ s y $T = 10$ s. Se puede apreciar la tendencia de que a medida que aumenta T : el robot oscilará alrededor de su posición inicial de cero, acercándose a su posición objetivo final con cada oscilación. Tenga en cuenta que para este caso unidimensional, la ECV ni el cambio de la función costo en el PMP da como resultado un movimiento diferente. Esto está de acuerdo con los fundamentos de control óptimo que implican que una constante de multiplicación en la función costo en el PMP no cambia el minimizador (Mesterton-Gibbons, 2009). Por lo tanto, los controladores también son iguales para los tres casos, ECV y ambas funciones costo (γ_A y γ_M) en el PMP, como se muestra en la Figura 4.2(b).

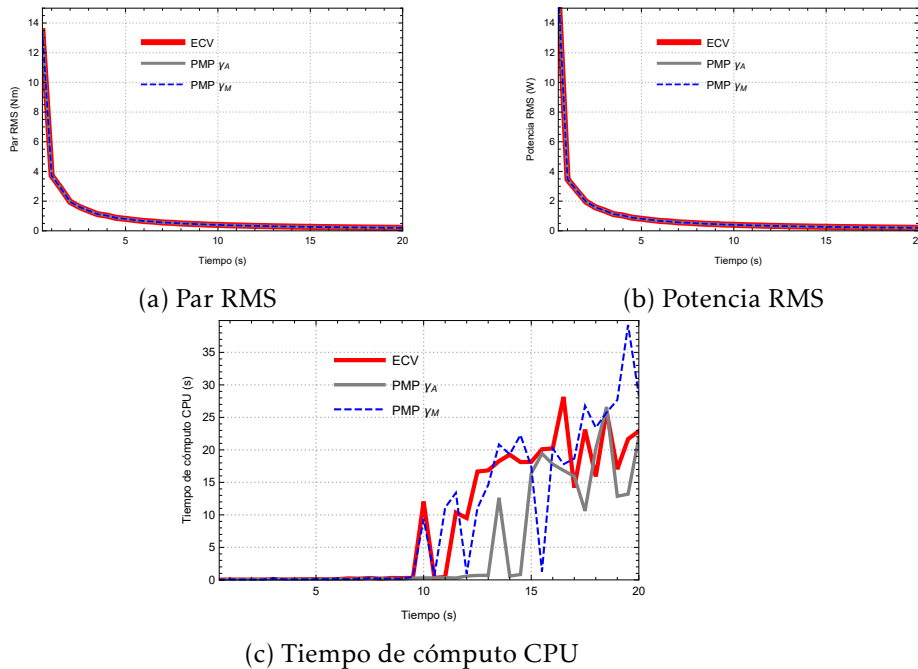


Figura 4.1: Control óptimo de un manipulador robótico de 1 GDL. Evolución del par RMS, potencia RMS y tiempo de cálculo del CPU necesarios para obtener una solución para diferentes valores de tiempo prescrito T . Los resultados de RMS son iguales con ECV y ambas funciones costo. El tiempo de cálculo del CPU aumenta después de que $T = 9$ s con ECV y γ_M (3.7).

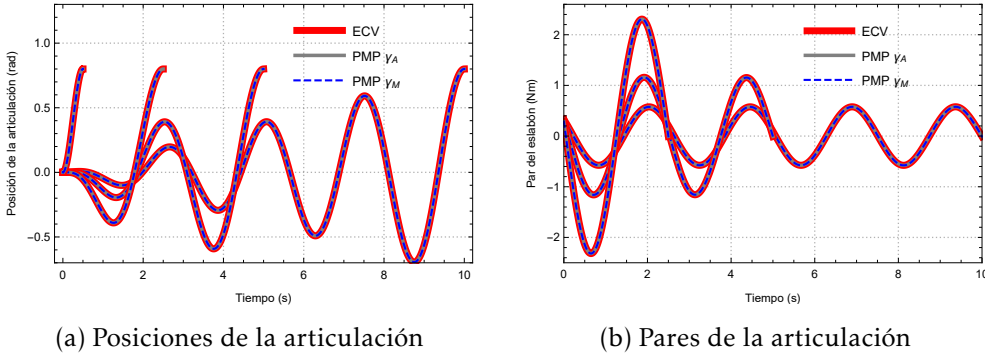


Figura 4.2: Control óptimo de un manipulador robótico de 1 GDL. Evolución de las posiciones de las articulaciones y los momentos de torsión para cuatro valores diferentes de tiempo prescrito, $T = 0.5$ s, $T = 2.5$ s, $T = 5$ s y $T = 10$ s. Para este manipulador, las posiciones y pares de articulación obtenidos son iguales con ECV y ambas funciones costo γ_A (3.6) y γ_M (3.7).

4.3 CONTROL ÓPTIMO DE UN MANIPULADOR ROBÓTICO DE 2 GDL

Ahora se considera el caso de un manipulador robótico de 2 GDL. Los resultados del inciso anterior confirmaron que γ_M (3.7) no da como resultado un aumento del rendimiento porque el tensor de masa tiene un componente constante único. Sin embargo, este no es el caso de ningún robot de n GDL para el cual $n > 1$. En tales casos, el tensor de masa $\mathbf{M} = \mathbf{M}(\mathbf{q})$ depende de la configuración. Naturalmente, el tensor de masas no será simplemente un factor de escala para robots con $n > 1$ GDL y, por lo tanto, se puede esperar que los resultados cambien al usar ECV y γ_M (3.7) comparadas con γ_A en el PMP (3.6).

4.3.1 PMP aplicado al manipulador de 2 GDL

Ahora que se tienen las ecuaciones que rigen el movimiento del manipulador de 2 GDL (3.4) y resolviendo simultáneamente se obtienen los valores de las aceleraciones \ddot{q}_1 y \ddot{q}_2

$$\begin{aligned}
 \zeta_1 = & \{ -(I_2 + m_2 a_2^2)(-u_1 + u_2 + g(m_1 a_1 + m_2 l_1) \text{sen}(q_1)) \\
 & + m_2 l_1 a_2 \cos(q_2)(-u_2 + m_2 g a_2 \text{sen}(q_1 + q_2)) \\
 & + \frac{1}{2} a_2 l_1 m_2 (a_2 l_1 m_2 \text{sen}(2q_2) \zeta_1^2 + 2(I_2 + a_2^2 m_2) \text{sen}(q_2) (\zeta_1 + \zeta_2)^2) \} / \\
 & \{ (I_2 + m_2 a_2^2)(I_1 + m_1 a_1^2 + m_2 l_1^2) - m_2^2 l_1^2 a_2^2 \cos(q_2)^2 \} = \ddot{q}_1
 \end{aligned} \quad (4.8)$$

$$\begin{aligned}
 \ddot{\zeta}_2 = & \{-(I_2 + m_2 a_2^2)u_1 + (I_1 + I_2 + m_1 a_1^2 + (a_2^2 + l_1^2)m_2)u_2 \\
 & + g(I_2 + m_2 a_2^2)(m_1 a_1 + m_2 l_1) \text{sen}(q_1) \\
 & - m_2 g a_2 (I_1 + m_1 a_1^2 + m_2 l_1^2) \text{sen}(q_1 + q_2) \\
 & + m_2 l_1 a_2 \cos(q_2)(-u_1 + 2u_2 + g(a_1 m_1 + l_1 m_2) \text{sen}(q_1) \\
 & - m_2 g a_2 \text{sen}(q_1 + q_2)) + m_2 l_1 a_2 \text{sen}(q_2)(-(I_1 + I_2 + m_1 a_1^2 \\
 & + (a_2^2 + l_1^2)m_2 + 2m_2 l_1 a_2 \cos(q_2))\zeta_1^2 \\
 & - 2(I_2 + m_2 a_2^2 + m_2 l_1 a_2 \cos(q_2))\zeta_1 \zeta_2 - (I_2 + m_2 a_2^2 + m_2 l_1 a_2 \cos(q_2))\zeta_2^2)\} / \\
 & \{(I_2 + m_2 a_2^2)(I_1 + m_1 a_1^2 + m_2 l_1^2) - m_2^2 l_1^2 a_2^2 \cos(q_2)^2\} = \ddot{q}_2
 \end{aligned} \tag{4.9}$$

que se utilizarán junto con (2.39) para construir el Hamiltoniano $H(\lambda, x, u)$

$$H(\lambda, x, u) = \gamma_A + p_1 \dot{q}_1 + \xi_1 \dot{\zeta}_1 + p_2 \dot{q}_2 + \xi_2 \dot{\zeta}_2$$

donde γ_A es el índice de desempeño definido por

$$\gamma_A = \frac{1}{2}(u_1^2 + u_2^2).$$

Por lo que ahora el Hamiltoniano resultante es

$$H(\lambda, x, u) = \frac{u_1^2 + u_2^2}{2} + p_1 \dot{q}_1 + \xi_1 \dot{\zeta}_1 + p_2 \dot{q}_2 + \xi_2 \dot{\zeta}_2. \tag{4.10}$$

Al aplicar el PMP a la ecuación (4.10) para u_1 y u_2 , es decir, $\frac{\partial H}{\partial u_1} = 0$ y $\frac{\partial H}{\partial u_2} = 0$ y resolviendo simultáneamente para encontrar u_1 y u_2 se obtiene

$$\begin{aligned}
 u_1 = & \{-I_2 \xi_1 - m_2 a_2^2 \xi_1 + I_2 \xi_2 + m_2 a_2^2 \xi_2 + m_2 l_1 a_2 \cos(q_2) \xi_2\} / \\
 & \{I_1 I_2 + I_2 m_1 a_1^2 + I_1 m_2 a_2^2 + I_2 m_2 l_1^2 + m_2 m_1 a_1^2 a_2^2 + m_2^2 l_1^2 a_2^2 - m_2^2 l_1^2 a_2^2 \cos(q_2)^2\}
 \end{aligned} \tag{4.11}$$

$$\begin{aligned}
 u_2 = & \{I_2 \xi_1 + a_2^2 m_2 \xi_1 + a_2 l_1 m_2 \cos(q_2) \xi_1 - I_1 \xi_2 - I_2 \xi_2 - a_1^2 m_1 \xi_2 \\
 & - a_2^2 m_2 \xi_2 - l_1^2 m_2 \xi_2 - 2a_2 l_1 m_2 \cos(q_2) \xi_2\} / \\
 & \{I_1 I_2 + a_1^2 I_2 m_1 + a_2^2 I_1 m_2 + I_2 l_1^2 m_2 + a_1^2 a_2^2 m_1 m_2 + a_2^2 l_1^2 m_2^2 - a_2^2 l_1^2 m_2^2 \cos(q_2)^2\}.
 \end{aligned} \tag{4.12}$$

Sustituyendo (4.11) y (4.12) en (4.8), (4.9) y (4.10) se tiene el siguiente espacio de estados que se muestra en forma simpléctica:

$$\left\{ \begin{array}{l} \dot{q}_1 = \frac{\partial H}{\partial p_1} \\ \dot{\zeta}_1 = \frac{\partial H}{\partial \xi_1} \\ \dot{q}_2 = \frac{\partial H}{\partial p_2} \\ \dot{\zeta}_2 = \frac{\partial H}{\partial \xi_2} \\ \dot{p}_1 = -\frac{\partial H}{\partial q_1} \\ \dot{\xi}_1 = -\frac{\partial H}{\partial \zeta_1} \\ \dot{p}_2 = -\frac{\partial H}{\partial q_2} \\ \dot{\xi}_2 = -\frac{\partial H}{\partial \zeta_2} \end{array} \right. \quad (4.13)$$

Ahora se tiene un conjunto de ocho EDO de primer orden que se resolverán usando los valores de la Tabla 4.2; así como tomando los siguientes valores de frontera

$$\left\{ \begin{array}{l} \mathbf{q}(0) = (q_1(0), q_2(0)) = (0, 0) \text{ rad} \\ \dot{\mathbf{q}}(0) = (\zeta_1(0), \zeta_2(0)) = (0, 0) \text{ rad s}^{-1} \\ \mathbf{q}(T) = (q_1(T), q_2(T)) = (0.8, 1.0) \text{ rad} \\ \dot{\mathbf{q}}(T) = (\zeta_1(T), \zeta_2(T)) = (0, 0) \text{ rad s}^{-1} \end{array} \right. \quad (4.14)$$

VALOR
$m_1 = 23.902 \text{ kg}$
$m_2 = 3.880 \text{ kg}$
$I_1 = 1.266 \text{ kg m}^2$
$I_2 = 0.093 \text{ kg m}^2$
$a_1 = 0.091 \text{ m}$
$a_2 = 0.048 \text{ m}$
$l_1 = 0.45 \text{ m}$

Tabla 4.2: Valores utilizados en el manipulador robótico de 2 GDL.

4.3.2 PMP aplicado al manipulador de 2 GDL (usando el tensor de masas)

Ahora que se tienen las ecuaciones que rigen el movimiento del manipulador de 2 GDL (3.4) y resolviendo simultáneamente se obtienen los valores de las aceleraciones \ddot{q}_1 y \ddot{q}_2

$$\begin{aligned} \ddot{\zeta}_1 = & \{-(I_2 + m_2 a_2^2)(-u_1 + u_2 + g(m_1 a_1 + m_2 l_1) \text{sen}(q_1)) \\ & + m_2 l_1 a_2 \cos(q_2)(-u_2 + m_2 g a_2 \text{sen}(q_1 + q_2)) \\ & + \frac{1}{2} a_2 l_1 m_2 (a_2 l_1 m_2 \text{sen}(2q_2) \zeta_1^2 + 2(I_2 + a_2^2 m_2) \text{sen}(q_2)(\zeta_1 + \zeta_2)^2)\} / \\ & \{(I_2 + m_2 a_2^2)(I_1 + m_1 a_1^2 + m_2 l_1^2) - m_2^2 l_1^2 a_2^2 \cos(q_2)^2\} = \ddot{q}_1 \end{aligned} \quad (4.15)$$

$$\begin{aligned} \ddot{\zeta}_2 = & \{-(I_2 + m_2 a_2^2)u_1 + (I_1 + I_2 + m_1 a_1^2 + (a_2^2 + l_1^2)m_2)u_2 \\ & + g(I_2 + m_2 a_2^2)(m_1 a_1 + m_2 l_1) \text{sen}(q_1) \\ & - m_2 g a_2 (I_1 + m_1 a_1^2 + m_2 l_1^2) \text{sen}(q_1 + q_2) \\ & + m_2 l_1 a_2 \cos(q_2)(-u_1 + 2u_2 + g(a_1 m_1 + l_1 m_2) \text{sen}(q_1) \\ & - m_2 g a_2 \text{sen}(q_1 + q_2)) + m_2 l_1 a_2 \text{sen}(q_2)(-(I_1 + I_2 + m_1 a_1^2 + (a_2^2 + l_1^2)m_2 \\ & + 2m_2 l_1 a_2 \cos(q_2) \zeta_1^2 \\ & - 2(I_2 + m_2 a_2^2 + m_2 l_1 a_2 \cos(q_2)) \zeta_1 \zeta_2 - (I_2 + m_2 a_2^2 + m_2 l_1 a_2 \cos(q_2)) \zeta_2^2)\} / \\ & \{(I_2 + m_2 a_2^2)(I_1 + m_1 a_1^2 + m_2 l_1^2) - m_2^2 l_1^2 a_2^2 \cos(q_2)^2\} = \ddot{q}_2 \end{aligned} \quad (4.16)$$

que se utilizarán junto con la ecuación (2.39) para construir el Hamiltoniano $H(\lambda, x, u)$

$$H(\lambda, x, u) = \gamma_M + p_1 \dot{q}_1 + \xi_1 \dot{\zeta}_1 + p_2 \dot{q}_2 + \xi_2 \dot{\zeta}_2$$

donde γ_M es el índice de desempeño definido por

$$\gamma_M = \frac{1}{2M} (u_1^2 + u_2^2).$$

Por lo que ahora el Hamiltoniano resultante es

$$H(\lambda, x, u) = \frac{u_1^2 + u_2^2}{2M} + p_1 \dot{q}_1 + \xi_1 \dot{\zeta}_1 + p_2 \dot{q}_2 + \xi_2 \dot{\zeta}_2. \quad (4.17)$$

Aplicando el PMP a la ecuación (4.17) para u_1 y u_2 , es decir, $\frac{\partial H}{\partial u_1} = 0$ y $\frac{\partial H}{\partial u_2} = 0$ y resolviendo simultáneamente para encontrar u_1 y u_2 se obtiene

$$u_1 = \xi_1 \quad (4.18)$$

$$u_2 = \xi_2. \quad (4.19)$$

Sustituyendo (4.18) y (4.19) en (4.15), (4.16) y (4.17) se tiene el siguiente espacio de estados que se muestra en forma simpléctica:

$$\begin{cases} \dot{q}_1 = \frac{\partial H}{\partial p_1} \\ \dot{\zeta}_1 = \frac{\partial H}{\partial \xi_1} \\ \dot{q}_2 = \frac{\partial H}{\partial p_2} \\ \dot{\zeta}_2 = \frac{\partial H}{\partial \xi_2} \\ \dot{p}_1 = -\frac{\partial H}{\partial q_1} \\ \dot{\xi}_1 = -\frac{\partial H}{\partial \zeta_1} \\ \dot{p}_2 = -\frac{\partial H}{\partial q_2} \\ \dot{\xi}_2 = -\frac{\partial H}{\partial \zeta_2}. \end{cases} \quad (4.20)$$

Ahora se tiene un conjunto de ocho EDOs de primer orden que se resolverán usando los valores de la Tabla 4.2; así como con los valores de frontera (4.14).

La Figura 4.3 muestra tres conjuntos de curvas. Siguiendo el paso 1) de la metodología en la sección 4.1, obtuvimos soluciones al conjunto de 8 EDO compuesto por (2.41) y (2.42), es decir, para γ_A (3.6) el espacio de estados (4.13) y para γ_M (3.7) el espacio de estados (4.20) en el PMP, así como para ECV, con valores diferentes del tiempo de trayectoria prescrito T . Hasta $T = 2.4$ s cuando se usa γ_A (3.6) con el PMP, hasta $T = 4.7$ s con ECV y hasta $T = 4.9$ s cuando se usa γ_M (3.7) con el PMP. Sorprendentemente, no se pudieron obtener soluciones que cumplen con los valores de frontera (4.14) para $T > 2.4$ s en el caso de γ_A (3.6). De hecho, existen huecos en las curvas correspondientes de la Figura 4.3 para el caso de γ_A : soluciones que cumplen eficazmente los valores de frontera (4.14) sólo se puede obtener dentro de los siguientes intervalos de T : $[0.5 \text{ s}, 1.2 \text{ s}]$, $[1.5 \text{ s}, 1.6 \text{ s}]$ y 2.4 s . Esto es causado por problemas numéricos como la rigidez, que comúnmente surgen en un control óptimo (Grancharova y Johansen, 2005). Por el contrario, las soluciones que cumplen los valores de frontera requeridos podrían obtenerse fácilmente hasta $T = 4.9$ s cuando se usa γ_M (3.7) en el PMP, como se muestra en las curvas de la Figura 4.3. Por lo tanto, para este manipulador robótico de 2 GDL, la estabilidad del solucionador se vió afectada al usar ECV y por el cambio en la función costo en el PMP. Por lo tanto, se puede ver que

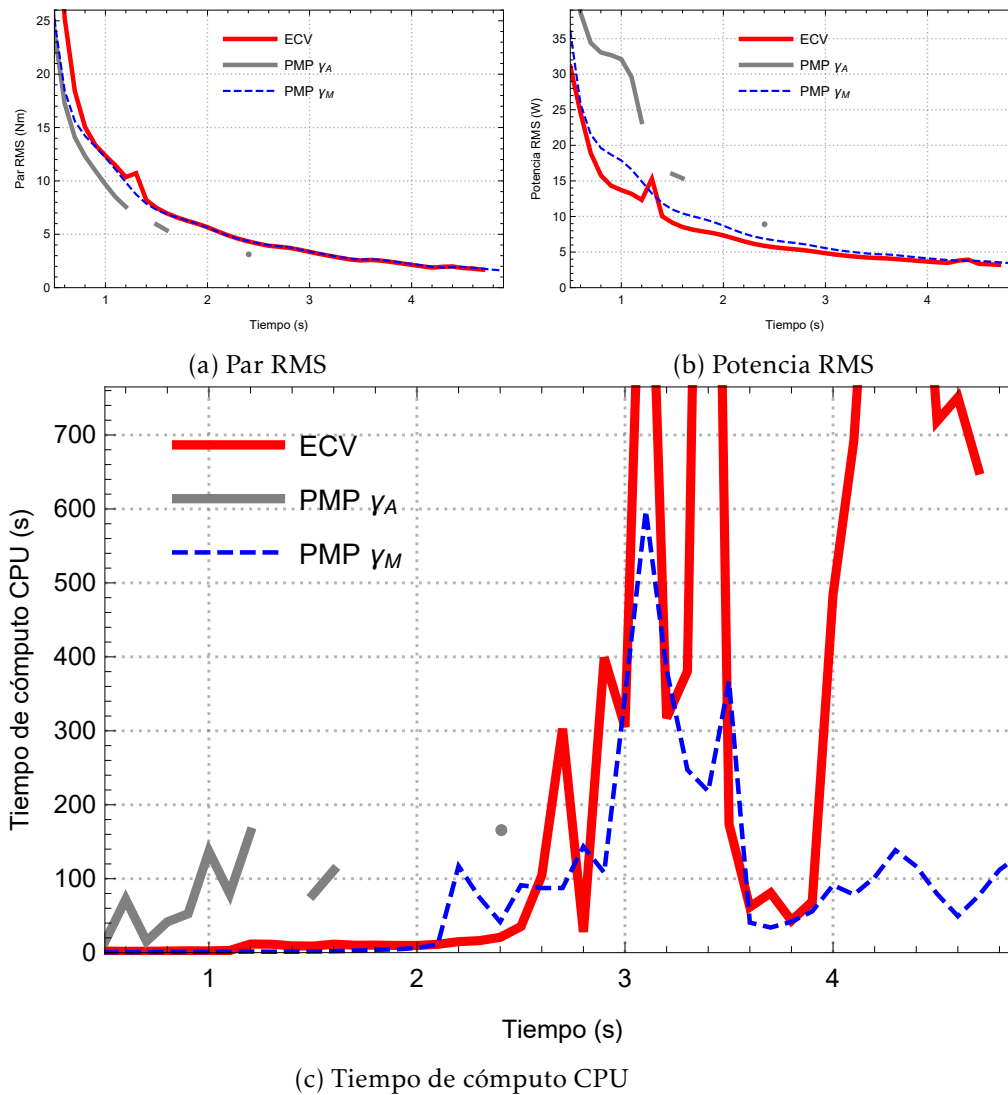


Figura 4.3: Control óptimo de un manipulador robótico de 2 GDL. Evolución del par RMS, potencia RMS y el tiempo de cálculo del CPU necesarios para obtener una solución para valores diferentes de tiempo prescrito T . Los resultados de RMS son más bajos con γ_A (3.6). El tiempo de cálculo del CPU es menor con γ_M (3.7). Tenga en cuenta que los espacios en las curvas muestran que no siempre se pueden encontrar soluciones cuando se usa γ_A (3.6). Las soluciones se obtienen de forma ininterrumpida al utilizar γ_M (3.7) hasta $T = 4.9$ s.

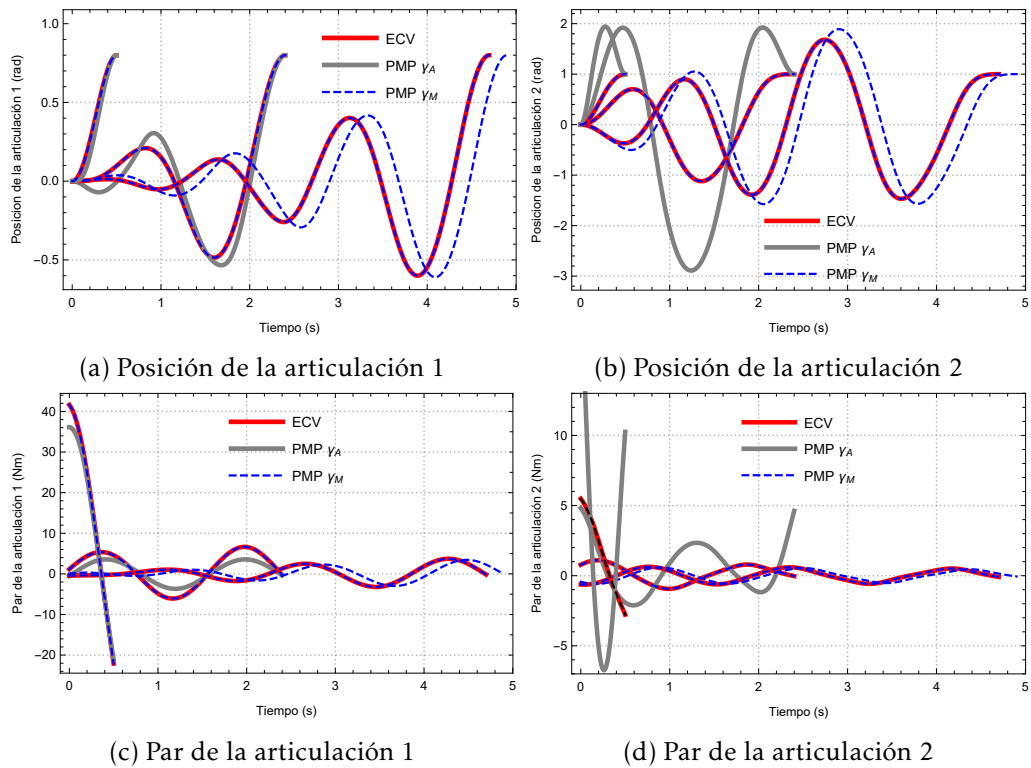


Figura 4.4: Control óptimo de un manipulador robótico de 2 GDL. Evolución de las posiciones de las articulaciones y los momentos de torsión para tres valores diferentes de tiempo prescrito, $T = 0.5$ s, $T = 1.5$ s y $T = 4.9$ s. Para este manipulador, las posiciones de las articulaciones y los pares de torsión obtenidos dependen de la elección de la función costo.

al usar ECV y la función costo propuesta γ_M (3.7) en el PMP aportan estabilidad con respecto a un aumento en el tiempo de trayectoria prescrito T .

Las trayectorias del par RMS y los valores de potencia RMS se muestran respectivamente en las figuras 4.3(a) y 4.3(b) para ECV y ambas funciones costo en el PMP. Se puede ver que γ_A conduce a un menor consumo de par. Sin embargo, γ_M en lugar, conduce a un menor consumo de potencia, pero es con ECV que se obtiene el menor consumo de potencia. Además, el tiempo de cómputo del CPU se reduce significativamente al usar γ_M en el PMP para un valor fijo de T , como se muestra en la Figura 4.3(c). La Tabla 4.3 muestra cómo el tiempo de cálculo del CPU requerido por el solucionador de EDO para obtener una solución se reduce con γ_M . Esta tabla indica la relación entre el tiempo que le

toma al solucionador obtener una solución con γ_A y el tiempo que le toma con γ_M . Aquí, obtuvimos adicionalmente tiempos de cómputo del CPU con un procesador Intel® Core™ i9-9880H de 8 núcleos a 2.3 GHz para obtener tiempos alternativos (sólo los tiempos de cómputo del CPU dependen del procesador). En la Tabla 4.3 se redondearon los valores al número entero más cercano y podemos ver cómo se redujeron los tiempos de cómputo del CPU: con γ_M en vez de γ_A , el solucionador encuentra una solución al menos 4 veces más rápido y 51 veces más rápido en promedio con el procesador i7-8550U (73 veces más rápido en promedio con el procesador i9-9880H). Por lo tanto, está claro que γ_M trae un aumento de rendimiento en términos de estabilidad numérica y tiempo de cómputo del CPU.

Relación	$\frac{t_{\gamma_A}}{t_{\gamma_M}}$ i7-8550U	$\frac{t_{\gamma_A}}{t_{\gamma_M}}$ i9-9880H
Valor base	4	5
Promedio	51	73

Tabla 4.3: Factor de disminución en el tiempo de cómputo del CPU requerido por NDSolve para obtener una solución: relación entre funciones costo comparables usando dos procesadores diferentes.

Ahora se evalúa el impacto de ECV y las funciones costos γ_A y γ_M en el PMP en la tendencia de posiciones óptimas y momentos de torsión a medida que aumenta el tiempo de trayectoria prescrito T . Las Figuras 4.4(a) y 4.4(b) muestran respectivamente esta tendencia para la primera y segunda articulaciones del robot, a valores fijos de T : 0.5 s, 2.4 s, 4.7 s y 4.9 s. Como se mencionó anteriormente, no se pudo obtener una trayectoria para $T = 4.9$ s con ECV ni con γ_A (3.6). También se puede ver que para la articulación 2 (Figura 4.4(b)) la amplitud del movimiento se reduce considerablemente cuando se usa ECV y γ_M en el PMP. De hecho, se puede ver que con ECV y γ_M la tendencia de ambas posiciones conjuntas a medida que aumenta T , se vuelve muy similar. Por lo tanto, al usar ECV y el tensor de masas en γ_M en el PMP actúa como un factor estabilizador que asegura que ambas articulaciones se comporten de manera similar con amplitudes de movimiento reducidas en comparación con los movimientos resultantes del uso de γ_A en el PMP. Lo mismo se aplica a los pares de torsión del robot como se muestra en las figuras 4.4(c) y 4.4(d).

5

CONCLUSIONES

En este trabajo de tesis se investigó sobre métodos numéricos de mecánica analítica para robótica avanzada que permitieron la creación de herramientas para las áreas de Ingeniería Mecánica, Teoría de Control y Mecatrónica. Se investigó y estudiaron técnicas clásicas de control óptimo presentes en la literatura tales como el Cálculo de Variaciones y el Principio del Máximo de Pontryagin. Se han implementado dichas técnicas y se efectuó una comparación entre ellas desde un punto de vista numérico, se han desarrollado y extendido los métodos y algoritmos estudiados al control óptimo de sistemas robóticos.

En el marco del Principio del Máximo de Pontryagin se han comparado dos funciones costo en un solucionador de EDO, una comúnmente usada en la literatura γ_A y otra propuesta por Rojas-Quintero *et al.* (2021a) γ_M , la cual incluye el tensor de masas inverso; pero que no habían sido comparadas numéricamente. Estas dos a su vez se compararon con el método de ecuaciones covariantes, encontrando que si bien con la función costo γ_A se obtiene menor consumo de par durante el movimiento óptimo de los manipuladores robóticos (ver figuras 4.4(c) y 4.4(d)) las ecuaciones de control óptimas que gobiernan exhiben rigidez hasta el punto de que las soluciones sólo se pueden encontrar para un conjunto limitado de intervalos de tiempo prescrito T (ver figura 4.3(c)). Además, las amplitudes de movimiento tienden a ser más altas con esta función costo en particular (ver figuras 4.4(a) y 4.4(b)). Debería ser posible ponderar cada unión con una función costo similar a γ_A aumentando o reduciendo los valores de la matriz de peso. Sin embargo, la determinación de estos valores no es sistemática. A la inversa, al usar los componentes del tensor de masas en la función costo γ_M que actúan como un factor estabilizador, se obtiene un mayor número de tiempos prescritos de trayectoria, un menor consumo de potencia y una menor amplitud de movimiento debido a que el tensor de masas inverso proporciona cierta estabilidad numérica que no había con γ_A . Es importante aclarar que todos estos beneficios obtenidos con la función costo γ_M sólo se pueden observar cuando $n > 1$; es decir, cuando el robot manipulador es de 2 GDL o más. También hay

que aclarar que los resultados obtenidos con las ecuaciones covariantes son muy similares a los obtenidos con la función costo γ_M , ya que se obtienen resultados de tiempo de trayectoria prescrita T hasta 4.7 s (con la función costo γ_M se obtienen valores hasta $T = 4.9$ s); por otro lado es con las ECV que se obtiene un menor consumo de potencia RMS (ver figura 4.3(c)).

Para trabajos futuros se espera poder llevar a cabo cálculos en los que se incluyan fuerzas restrictivas como la fricción, así como realizar ensayos en un robot manipulador de más de 2 GDL y poder realizar experimentos en un brazo robótico real.

BIBLIOGRAFÍA

- Almuslimani, I. y Vilmart, G. (2021) . Explicit stabilized integrators for stiff optimal control problems. *SIAM Journal on Scientific Computing*, 43(2):A721–A743.
- Arutyunov, A. y Karamzin, D. (2020) . A survey on regularity conditions for state-constrained optimal control problems and the non-degenerate maximum principle. *Journal of Optimization Theory and Applications*, 184(3):697–723.
- Ascher, U. M., Ruuth, S. J., y Spiteri, R. J. (1997) . Implicit-explicit runge-kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2):151–167. Special Issue on Time Integration.
- Asgari, M. y Nikoobin, A. (2020) . Analysis of optimal dynamic manipulation for robotic manipulator based on pontryagin’s minimum principle. *Arabian Journal for Science and Engineering*, 45:9159–9169.
- Bellman, R. E. (1957) . *Dynamic Programming*. Princeton University Press, Princeton, New Jersey.
- Callies, R. y Rentrop, P. (2008) . Optimal control of rigid-link manipulators by indirect methods. *GAMM-Mitteilungen*, 31(1):27–58.
- Chen, D., Wang, B., y Wang, L. (2017) . Optimal control of robot manipulators via double generating functions. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2337–2342.
- Chenavaz, R. (2017) . Dynamic quality policies with reference quality effects. *Applied Economics*, 49(32):3156–3162.
- Dubois, F., Fortuné, D., Rojas Quintero, J. A., y Vallée, C. (2015) . Pontryagin calculus in riemannian geometry. In Nielsen, F. y Barbaresco, F., editors, *Geometric Science of Information*, pages 541–549, Cham. Springer International Publishing.
- Engeln-Müllges, G. y Uhlig, F. (1996) . *Numerical Algorithms with C*. Springer-Verlag, Berlin, Heidelberg.

- Eriksson, A. (2007) . Temporal finite elements for target control dynamics of mechanisms. *Computers & Structures*, 85(17):1399–1408. Computational Structures Technology.
- Eriksson, A. y Nordmark, A. (2010) . Temporal finite element formulation of optimal control in mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 199(25):1783–1792.
- Fleisch, D. A. (2011) . *A Student's Guide to Vectors and Tensors*. Student's Guides. Cambridge University Press.
- Ghasemi, M., Kashiri, N., y Dardel, M. (2012) . Time-optimal trajectory planning of robot manipulators in point-to-point motion using an indirect method. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 226:473–484.
- Grancharova, A. y Johansen, T. A. (2005) . *Survey of Explicit Approaches to Constrained Optimal Control*, pages 47–97. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Grinfeld, P. (2013) . *Introduction to Tensor Analysis and the Calculus of Moving Surfaces*. Springer New York, New York, NY.
- Hairer, E. y Wanner, G. (1996) . *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer Series in Computational Mathematics N°14. Springer-Verlag Berlin Heidelberg, 2 edition.
- Hamill, P. (2013) . *A Student's Guide to Lagrangians and Hamiltonians*. Student's Guides. Cambridge University Press.
- Hao, Z. (2014) . Optimal trajectory generation via double generating functions and application to biped robots. In *Nagoya University*.
- Herman, P. y Kozłowski, K. (2006) . Evaluation of manipulator dynamics based on optimal controller. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 6122–6127.
- Kaphle, M. y Eriksson, A. (2008) . Optimality in forward dynamics simulations. *Journal of biomechanics*, 41(6):1213–21.
- Kaserer, D., Gattringer, H., y Muller, A. (2019) . Nearly optimal path following with jerk and torque rate limits using dynamic programming. *IEEE Transactions on Robotics*, 35(2):521–528.

- Kelly, R., Santibáñez Davila, V., y Loría Perez, J. A. (2005) . *Control of Robot Manipulators in Joint Space*. Advanced Textbooks in Control and Signal Processing. Springer-Verlag London.
- Kirk, D. (2004) . *Optimal Control Theory an Introduction*. Dover.
- Liberzon, D. (2012) . *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton University Press, Princeton, New Jersey.
- Longuski, J. M., Guzmán, J. J., y Prussing, J. E. (2014) . *Optimal Control with Aerospace Applications*. Number 32 in Space Technology Library. Springer New York.
- Mesterton-Gibbons, M. (2009) . *A primer on the calculus of variations and optimal control theory*, volume 50. American Mathematical Society, Providence, Rhode Island.
- Mirz, C., Scholer, F., Barreto, J. P., y Corves, B. (2018) . Optimal control based path planning for parallel kinematic manipulators utilising natural motion. In *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pages 223–228. IEEE.
- Nikoobin, A. y Moradi, M. (2011) . Optimal balancing of robot manipulators in point-to-point motion. *Robotica*, 29:233–244.
- Pesch, H. J. y Plail, M. (2009) . The maximum principle of optimal control: A history of ingenious ideas and missed opportunities. *Control and Cybernetics*, 38:973–995.
- Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., y Mishchenko, E. F. (1962) . *The Mathematical Theory of Optimal Processes*. Interscience Publishers, New York.
- Putkaradze, V. y Rogers, S. (2020) . On the Optimal Control of a Rolling Ball Robot Actuated by Internal Point Masses. *Journal of Dynamic Systems, Measurement, and Control*, 142(5).
- Rojas-Quintero, J. A., Rojas-Estrada, J. A., Villalobos-Chin, J., Santibáñez, V., y Bugarin, E. (2021a) . Optimal controller applied to robotic systems using covariant control equations. *International Journal of Control*, 0(0):1–14.
- Rojas Quintero, J. A., Vallée, C., Gazeau, J., Seguin, P., y Arsicault, M. (2013) . An alternative to pontryagin’s principle for the optimal control of jointed arm robots. In *21ème Congrès Français de Mécanique*, Courbevoie, France. AFM.

- Rojas-Quintero, J. A., Villalobos-Chin, J., y Santibanez, V. (2021b) . Optimal control of robotic systems using finite elements for time integration of covariant control equations. *IEEE Access*, 9:104980–105001.
- Rowland, T. (2021) . Computation time.
- Schättler, H. y Ledzewicz, U. (2012) . *Geometric Optimal Control*, volume 38 of *Interdisciplinary Applied Mathematics*. Springer New York.
- Spong, M. W., Hutchinson, S., y Vidyasagar, M. (2005) . *Robot Modeling and Control*. Wiley.
- Sussmann, H. J. y Willems, J. C. (1997) . 300 years of optimal control: from the brachistochrone to the maximum principle. *IEEE Control Systems Magazine*, 17(3):32–44.
- Vezvari, M. R., Nikoobin, A., y Ghoddosian, A. (2020) . Zero-power balancing a two-link robot manipulator for a predefined point-to-point task. *Journal of Mechanical Science and Technology*, 34(6):2585–2595.
- Zignoli, A., Biral, F., Pellegrini, B., Jinha, A., Herzog, W., y Schena, F. (2017) . An optimal control solution to the predictive dynamics of cycling. *Sport Sciences for Health*, 13(2):381–393.

A

PROBLEMAS ISOPERIMÉTRICOS

El clásico problema isoperimétrico es el de encontrar la forma de una longitud dada de cuerda para encerrar el área más grande posible (ver Figura A.1). Suponiendo que un trozo de cuerda de longitud L está unido a los puntos $(a, 0)$ y $(b, 0)$ y debe estar completamente en la región donde $y > 0$; suponemos que $L > b - a > 0$. Entonces el problema isoperimétrico se convierte en el de minimizar

$$J[y] = - \int_a^b y \, dx \quad (\text{A.1})$$

sujeto a las condiciones de frontera $y(a) = 0 = y(b)$ y la ecuación de restricción

$$\int_0^L 1 \, ds = \int_a^b \sqrt{1 + (y')^2} \, dx = L. \quad (\text{A.2})$$

Empezamos con el problema más general de minimizar

$$J[y] = \int_a^b F(x, y, y') \, dx \quad (\text{A.3})$$

sujeto a las condiciones de frontera

$$y(a) = \alpha \text{ y } y(b) = \beta$$

y la ecuación de restricción

$$I[y] = L,$$

donde

$$I[y] = \int_a^b G(x, y, y') \, dx. \quad (\text{A.4})$$

Si I tiene un valor mínimo m , entonces estamos obligados a suponer que

$$L > m.$$

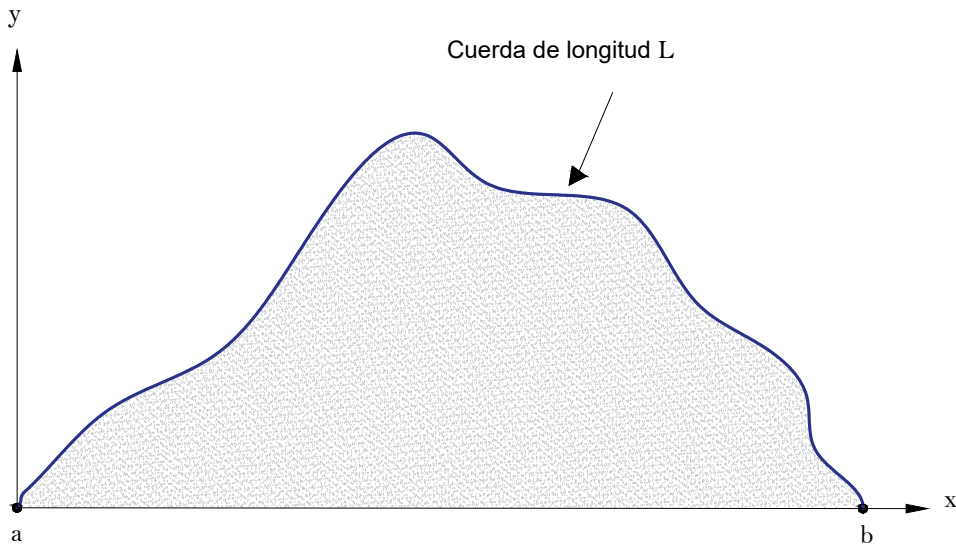


Figura A.1: Área encerrada por la cuerda L .

Por que si $L = m$, entonces el problema carece de interés; mientras que si $L < m$, entonces el problema no tiene solución. En el caso del problema clásico, por ejemplo, suponemos que $L > b - a > 0$. Buscamos una condición necesaria para la curva admisible Γ_0 definida por $y = \phi(x)$ con

$$\phi(a) = \alpha, \phi(b) = \beta$$

y

$$\int_a^b G(x, \phi(x), \phi'(x)) dx = L$$

para minimizar el funcional $J[y]$ definido por (A.3). Es decir, suponemos la existencia de la función minimizadora ϕ , y luego preguntamos que propiedades

debe tener ϕ inevitablemente en virtud de ser el minimizador. Consideramos una familia de curvas de prueba de dos parámetros Γ definida por

$$y = y_\epsilon(x) = \phi(x) + \epsilon_1 \eta_1(x) + \epsilon_2 \eta_2(x) \quad (\text{A.5})$$

donde el subíndice ϵ ahora se interpreta como un vector bidimensional de parámetros (ϵ_1, ϵ_2) . Tanto ϵ_1 como ϵ_2 pueden ser positivos o negativos, η_1 y η_2 son funciones admisibles arbitrarias, es decir, $\eta_i \in D_1$ con

$$\eta_i(a) = 0 = \eta_i(b) \text{ para } i = 1, 2.$$

Sigue siendo coherente utilizar Γ_0 para la curva de minimización $y = \phi(x)$: ahora interpretamos el subíndice 0 como el vector 0 bidimensional. Por supuesto, $\epsilon = (0, 0)$ designa la función minimizadora

$$J[\Gamma_0] \leq J[\gamma_\epsilon] \quad (\text{A.6})$$

sujeto a

$$I[\Gamma_\epsilon] = L \quad (\text{A.7})$$

para todos los η admisibles. Tan pronto se elige un particular η_1 y η_2 , $J[\Gamma_\epsilon]$ se convierte en una función bivariada estándar de ϵ_1 y ϵ_2 .

Por tanto podemos reescribir (A.6) y (A.7) como

$$J(0, 0) \leq J(\epsilon_1, \epsilon_2) \quad (\text{A.8})$$

sujeto a

$$I(\epsilon_1, \epsilon_2) = L$$

donde al sustituir de (A.5) en (A.3) y (A.4)

$$J(\epsilon_1, \epsilon_2) = \int_a^b F(x, \phi(x) + \epsilon_1 \eta_1(x) + \epsilon_2 \eta_2(x), \phi'(x) + \epsilon_1 \eta_1'(x) + \epsilon_2 \eta_2'(x)) dx$$

y

$$I(\epsilon_1, \epsilon_2) = \int_a^b F(x, \phi(x) + \epsilon_1 \eta_1(x) + \epsilon_2 \eta_2(x), \phi'(x) + \epsilon_1 \eta_1'(x) + \epsilon_2 \eta_2'(x)) dx;$$

porque ambas, ϵ_1 y ϵ_2 , pueden ser positivos o negativos. (A.8) implica que $J(\epsilon_1, \epsilon_2)$ debe tener un mínimo interior restringido donde $\epsilon_1 = 0 = \epsilon_2$. Se desprende del cálculo ordinario de funciones bivariadas que proporcionó

$$I_{\epsilon_2}(0, 0) \neq 0$$

(donde el subíndice denota diferenciación parcial), debe existir un multiplicador de Lagrange λ tal que

$$J_{\epsilon_1}(0,0) - \lambda I_{\epsilon_1}(0,0) = 0 = J_{\epsilon_2}(0,0) - \lambda I_{\epsilon_2}(0,0).$$

Una condición necesaria para que Γ_0 minimice $J[y]$ sujeto a $I[y] = L$ y $y(a) = \alpha$, $y(b) = \beta$ es que exista un multiplicador de Lagrange λ tal que $y = \phi(x)$ satisfaga la EEL no para F sino para

$$\Psi(x, y, y') = F(x, y, y') - \lambda G(x, y, y').$$

para el problema isoperímetro clásico, de (A.1) y (A.2) se sigue que

$$\Psi(x, y, y') = -y - \lambda \sqrt{1 + (y')^2}$$

es independiente de x . Entonces, por (2.6), una primera integral de la EEL es

$$y' \frac{\partial \Psi}{\partial y'} - \Psi = \frac{\lambda}{\sqrt{1 + (y')^2}} + y = \text{constante} = l.$$

B

CONTROL EN TIEMPO ÓPTIMO

Una clase importante de problemas de control óptimo concierne al control óptimo en el tiempo de un sistema lineal: las ecuaciones de estado son lineales en x , y el costo es el tiempo que tarda en transferir x desde x^0 a su estado final que suponemos que es el origen. El problema descrito en la sección 2.2 es de este tipo. Un problema más general con $m = 1$ y $n = 2$ es el de encontrar un escalar continuo a trozos $u \in [-1, 1]$ para transferir $x = (x_1, x_2)$ que satisfaga

$$\begin{aligned}\dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + b_1u, \\ \dot{x}_2 &= a_{12}x_1 + a_{22}x_2 + b_2u;\end{aligned}\tag{B.1}$$

o en notación matricial

$$\dot{x} = Ax + bu\tag{B.2}$$

con $x = [x_1 \ x_2]^T$, $b = [b_1 \ b_2]^T$ y $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ de x^0 a 0 (el vector bidimensional cero) de tal manera que se minimice

$$J = \int_0^{t_1} 1 \, dt = t_1.$$

El Hamiltoniano es

$$H = -1 + (a_{11}x_1 + a_{12}x_2)\lambda_1 + (a_{21}x_1 + a_{22}x_2)\lambda_2 + \sigma u$$

donde

$$\sigma(t) = b_1\lambda_1(t) + b_2\lambda_2(t)\tag{B.3}$$

define la función de conmutación; y las ecuaciones de co-estado producen

$$\begin{aligned}\dot{\lambda}_1 &= -\frac{\partial H}{\partial x_1} = -a_{11}\lambda_1 - a_{21}\lambda_2 \\ \dot{\lambda}_2 &= -\frac{\partial H}{\partial x_2} = -a_{12}\lambda_1 - a_{22}\lambda_2;\end{aligned}$$

o en notación matricial

$$\dot{\lambda} = -A^T \lambda$$

con $\lambda = [\lambda_1 \ \lambda_2]^T$. Para empezar, supongamos que la matriz A en (B.2) tiene dos eigenvalores distintos digamos r_1 y r_2 distintos de cero. Entonces A^T también tiene eigenvalores r_1 y r_2 implicando que la solución de $\dot{\lambda} = -A^T \lambda$ son combinaciones lineales de $e^{r_1 t}$ y de $e^{r_2 t}$, y por lo tanto de (B.3) que σ es una combinación lineal de $e^{r_1 t}$ y $e^{r_2 t}$, digamos

$$\sigma(t) = k_1 e^{r_1 t} + k_2 e^{r_2 t}.$$

De ello se deduce que σ puede desaparecer máximo una vez, en el tiempo

$$t_s = \frac{1}{r_2 - r_1} \ln \left(-\frac{k_1}{k_2} \right)$$

(y sólo si k_1 y k_2 tienen signos opuestos). También se sigue del Principio de Pontryagin que

$$u^*(t) = \text{sgn}(\sigma(t)) \quad (\text{B.4})$$

es constante por partes, con máximo una conmutación de control. Así de (B.1) y (B.4) las ecuaciones que gobiernan un arco potencialmente óptimo son

$$\begin{aligned} \dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + b_1, \\ \dot{x}_2 &= a_{12}x_1 + a_{22}x_2 + b_2 \end{aligned} \quad (\text{B.5})$$

($u^* = 1$) con un punto de equilibrio P^+ en

$$(x_1, x_2) = \left(\frac{a_{12}b_2 - a_{22}b_1}{\det(A)}, \frac{a_{21}b_1 - a_{11}b_2}{\det(A)} \right)$$

siempre que $\det(A) \neq 0$; o

$$\begin{aligned} \dot{x}_1 &= a_{11}x_1 + a_{12}x_2 - b_1, \\ \dot{x}_2 &= a_{12}x_1 + a_{22}x_2 - b_2 \end{aligned} \quad (\text{B.6})$$

($u^* = -1$) con un punto de equilibrio P^- en

$$(x_1, x_2) = \left(\frac{a_{22}b_1 - a_{12}b_2}{\det(A)}, \frac{a_{11}b_2 - a_{21}b_1}{\det(A)} \right).$$

De nuevo si $\det(A) \neq 0$. Los eigenvalores de A determinan la naturaleza de esos puntos de equilibrio y, por tanto, el patrón de trayectorias del plano de fase,

que describimos como positivas o negativas según surjan como soluciones de (B.5) o (B.6), respectivamente.

Un sistema dinámico lineal con un equilibrio de nodo estable es controlable desde cualquier lugar en \mathbb{R}^2 , mientras que uno de equilibrio de nodo inestable es controlable sólo desde una pequeña región que contiene el origen. ¿Qué pasa con un equilibrio de punto de silla? La intuición sugiere que la región desde la cual es controlable debería ser de alguna manera intermedia entre una pequeña región que contiene el origen y la totalidad de \mathbb{R}^2 porque un punto de silla es un equilibrio parcialmente estable, intermedio entre un nodo estable (dos eigenvalores negativos) y un nodo inestable (cero eigenvalores negativos). La intuición resulta ser correcta. Hasta ahora hemos considerado solo problemas de control de tiempo óptimo para los cuales los eigenvalores de la matriz A son reales. Entonces hay como máximo una conmutación de control. Sin embargo, son posibles múltiples conmutaciones cuando los eigenvalores de A son complejos.



CÓDIGO DEL CÁLCULO EN WOLFRAM MATHEMATICA

Desarrollo de ecuaciones (procedimiento explícito)

El algoritmo es el siguiente:

1. Despejar \ddot{q}_i de las ecuaciones de movimiento y poner dichas ecuaciones en espacio de estados. En este paso, se pasa de un sistema de EDO de segundo orden a un sistema de EDO de primer orden.
2. Establecer el Hamiltoniano H tal que $H = y + \lambda_i x_i$ (y es la función costo).
3. Despejar u_i del PMP: $\frac{\partial H}{\partial u_i} = 0$ que ahora será función de los estados.
4. Inyectar los u_i en los estados del paso 1.
5. Establecer las ecuaciones adjuntas como $\dot{\lambda}_i = -\frac{\partial H}{\partial x_i}$

Paso 1.

```

In[1]:= Solve[{o4 Sin[q1[t]] + o5 Sin[q1[t] + q2[t]] - o3 Sin[q2[t]] q2'[t] (2 q1'[t] + q2'[t]) + (o1 + o2 + 2 o3 Cos[q2[t]]) q1''[t] + (o2 + o3 Cos[q2[t]]) q2''[t] == u1[t],
  o5 Sin[q1[t] + q2[t]] + o3 Sin[q2[t]] q1'[t]^2 + (o2 + o3 Cos[q2[t]]) q1''[t] + o2 q2''[t] == u2[t]}]
/. {q1[t] -> x1[t], q1'[t] -> x2[t], q2[t] -> x3[t], q2'[t] -> x4[t]},
{q1''[t], q2''[t]}]

```

```

In[2]:= qppRule =

```

$$\left\{ \left\{ q1''[t] \rightarrow \frac{1}{o1 o2 - o3^2 \cos[x3[t]]^2} \left(-o2 o4 \sin[x1[t]] + o3 o5 \cos[x3[t]] \sin[x1[t] + x3[t]] + o2 u1[t] - (o2 + o3 \cos[x3[t]]) u2[t] + o3 \sin[x3[t]] (o3 \cos[x3[t]] x2[t]^2 + o2 (x2[t] + x4[t])^2) \right), \right. \right.$$

$$q2''[t] \rightarrow \frac{1}{o1 o2 - o3^2 \cos[x3[t]]^2} \left(o4 (o2 + o3 \cos[x3[t]]) \sin[x1[t]] - o5 (o1 + o3 \cos[x3[t]]) \sin[x1[t] + x3[t]] - (o2 + o3 \cos[x3[t]]) u1[t] + (o1 + o2 + 2 o3 \cos[x3[t]]) u2[t] - o3 \sin[x3[t]] \left((o1 + o2 + 2 o3 \cos[x3[t]]) x2[t]^2 + 2 (o2 + o3 \cos[x3[t]]) x2[t] \cdot x4[t] + (o2 + o3 \cos[x3[t]]) x4[t]^2 \right) \right) \left. \right\};$$

```

In[3]:= x1p = x2[t];

```

```

In[4]:= x1p = x2[t];
x2p = q1'[t] /. qppRule;
x3p = x4[t];
x4p = q2'[t] /. qppRule;

```

Paso 2.

```

In[5]:= (* H = \gamma \sum (\lambda_i x_i) *)
H[t_] := (u1[t]^2 + u2[t]^2) / 2 + x5[t] x1p + x6[t] x2p + x7[t] x3p + x8[t] x4p

```

Paso 3.

```

In[6]:= (* OH/Ou1=0 para despejar u1 *)
(* OH/Ou2=0 para despejar u2 *)

```

```

uExpr = {Solve[D[H[t], u1[t]] == 0, u1[t]][[1, 1]], Solve[D[H[t], u2[t]] == 0, u2[t]][[1, 1]]}

```

```

Out[6]:= {u1[t] -> (-o2 x6[t] + o2 x8[t] + o3 Cos[x3[t]] x8[t]) / (o1 o2 - o3^2 Cos[x3[t]]^2), u2[t] -> (o2 x6[t] + o3 Cos[x3[t]] x6[t] - o1 x8[t] - o2 x8[t] - 2 o3 Cos[x3[t]] x8[t]) / (o1 o2 - o3^2 Cos[x3[t]]^2)}

```

```

In[7]:= qppRule /. uExpr

```

```

In[8]:= {{q1''[t] ->
  1 / (o1 o2 - o3^2 Cos[x3[t]]^2) (
    -o2 o4 Sin[x1[t]] + o3 o5 Cos[x3[t]] Sin[x1[t] + x3[t]] + o3 Sin[x3[t]] (o3 Cos[x3[t]] x2[t]^2 + o2 (x2[t] + x4[t])^2) -
    (o2 + o3 Cos[x3[t]]) (o2 x6[t] + o3 Cos[x3[t]] x6[t] - o1 x8[t] - o2 x8[t] - 2 o3 Cos[x3[t]] x8[t]),
    o2 (-o2 x6[t] + o2 x8[t] + o3 Cos[x3[t]] x8[t]) / (o1 o2 - o3^2 Cos[x3[t]]^2)
  ),
  q2''[t] ->
  1 / (o1 o2 - o3^2 Cos[x3[t]]^2) (
    o4 (o2 + o3 Cos[x3[t]]) Sin[x1[t]] - o5 (o1 + o3 Cos[x3[t]]) Sin[x1[t] + x3[t]] -
    o3 Sin[x3[t]] ((o1 + o2 + 2 o3 Cos[x3[t]]) x2[t]^2 + 2 (o2 + o3 Cos[x3[t]]) x2[t] \cdot x4[t] + (o2 + o3 Cos[x3[t]]) x4[t]^2) +
    (o1 + o2 + 2 o3 Cos[x3[t]]) (o2 x6[t] + o3 Cos[x3[t]] x6[t] - o1 x8[t] - o2 x8[t] - 2 o3 Cos[x3[t]] x8[t]) /
    (o1 o2 - o3^2 Cos[x3[t]]^2)
  )
}

```

```

q2'[t] +

$$\frac{1}{\theta_1 \theta_2 - \theta_3^2 \cos[x_3[t]]^2} \left\{ \theta_4 (\theta_2 + \theta_3 \cos[x_3[t]]) \sin[x_1[t]] - \theta_5 (\theta_1 + \theta_3 \cos[x_3[t]]) \sin[x_1[t] + x_3[t]] - \right.$$


$$\frac{\theta_3 \sin[x_3[t]] \left\{ (\theta_1 + \theta_2 + 2 \theta_3 \cos[x_3[t]]) x_2[t]^2 + 2 (\theta_2 + \theta_3 \cos[x_3[t]]) x_2[t] - x_4[t] + (\theta_2 + \theta_3 \cos[x_3[t]]) x_4[t]^2 + \right.$$


$$\left. (\theta_1 + \theta_2 + 2 \theta_3 \cos[x_3[t]]) (\theta_2 x_6[t] + \theta_3 \cos[x_3[t]] x_6[t] - \theta_1 x_8[t] - \theta_2 x_8[t] - 2 \theta_3 \cos[x_3[t]] x_8[t]) - \right.$$


$$\left. \frac{\theta_1 \theta_2 - \theta_3^2 \cos[x_3[t]]^2}{\theta_1 \theta_2 - \theta_3^2 \cos[x_3[t]]^2} \right\} \left. \right\}$$


Paso 4.
In[ ]:= x1p = x2[t]
x2p = q1'[t] /. qppRule /. uExpr
x3p = x4[t]
x4p = q2'[t] /. qppRule /. uExpr

Paso 5.
In[ ]:= D[-H[t] /. uExpr, x1[t]]
D[-H[t] /. uExpr, x2[t]]
D[-H[t] /. uExpr, x3[t]]
D[-H[t] /. uExpr, x4[t]]

```

Ecuaciones diferenciales y parámetros del robot

Ecuaciones diferenciales del PMP en espacio de estados

```

In[ ]:= PMPeq = {x1'[t] == x2[t],
x2'[t] == 
$$\frac{1}{\theta_1 \theta_2 - \theta_3^2 \cos[x_3[t]]^2} \left\{ -\theta_2 \theta_4 \sin[x_1[t]] + \theta_3 \theta_5 \cos[x_3[t]] \sin[x_1[t] + x_3[t]] + \theta_3^2 \cos[x_3[t]] \sin[x_3[t]] x_2[t]^2 + \theta_2 \theta_3 \sin[x_3[t]] (x_2[t] + x_4[t])^2 + \right.$$


$$\left. x_6[t] - 2 x_8[t] - (-2 \theta_2 (\theta_1 + 2 \theta_2 + 2 \theta_3 \cos[x_3[t]]) x_6[t] + 2 (\theta_2 (3 \theta_1 + 2 \theta_2) + (\theta_1 + 4 \theta_2) \theta_3 \cos[x_3[t]]) x_8[t]) / (-2 \theta_1 \theta_2 + \theta_3^2 + \theta_3^2 \cos[2 x_3[t]]) \right\},$$

x3'[t] == x4[t],
x4'[t] ==

```

Ecuaciones diferenciales y parámetros del robot

Ecuaciones diferenciales del PMP en espacio de estados

```

In[ ]:= PMPeq = {x1'[t] == x2[t],
x2'[t] == 
$$\frac{1}{\theta_1 \theta_2 - \theta_3^2 \cos[x_3[t]]^2} \left\{ -\theta_2 \theta_4 \sin[x_1[t]] + \theta_3 \theta_5 \cos[x_3[t]] \sin[x_1[t] + x_3[t]] + \theta_3^2 \cos[x_3[t]] \sin[x_3[t]] x_2[t]^2 + \theta_2 \theta_3 \sin[x_3[t]] (x_2[t] + x_4[t])^2 + \right.$$


$$\left. x_6[t] - 2 x_8[t] - (-2 \theta_2 (\theta_1 + 2 \theta_2 + 2 \theta_3 \cos[x_3[t]]) x_6[t] + 2 (\theta_2 (3 \theta_1 + 2 \theta_2) + (\theta_1 + 4 \theta_2) \theta_3 \cos[x_3[t]]) x_8[t]) / (-2 \theta_1 \theta_2 + \theta_3^2 + \theta_3^2 \cos[2 x_3[t]]) \right\},$$

x3'[t] == x4[t],
x4'[t] ==

$$\frac{1}{\theta_1 \theta_2 - \theta_3^2 \cos[x_3[t]]^2} \left\{ \theta_4 (\theta_2 + \theta_3 \cos[x_3[t]]) \sin[x_1[t]] - \theta_5 (\theta_1 + \theta_3 \cos[x_3[t]]) \sin[x_1[t] + x_3[t]] - \right.$$


$$\frac{\theta_3 \sin[x_3[t]] \left\{ (\theta_1 + \theta_2 + 2 \theta_3 \cos[x_3[t]]) x_2[t]^2 + 2 (\theta_2 + \theta_3 \cos[x_3[t]]) x_2[t] - x_4[t] + (\theta_2 + \theta_3 \cos[x_3[t]]) x_4[t]^2 + \right.$$


$$\left. (\theta_2 + \theta_3 \cos[x_3[t]]) (\theta_2 x_6[t] - (\theta_2 + \theta_3 \cos[x_3[t]]) x_8[t]) + (\theta_1 + \theta_2 + 2 \theta_3 \cos[x_3[t]]) ((\theta_2 + \theta_3 \cos[x_3[t]]) x_6[t] - (\theta_1 + \theta_2 + 2 \theta_3 \cos[x_3[t]]) x_8[t]) / \right.$$


$$\left. \frac{(\theta_1 \theta_2 - \theta_3^2 \cos[x_3[t]]^2)}{\theta_1 \theta_2 - \theta_3^2 \cos[x_3[t]]^2} \right\},$$

x5'[t] ==

$$\frac{1}{\theta_1 \theta_2 - \theta_3^2 \cos[x_3[t]]^2} \left\{ (\theta_2 \theta_4 \cos[x_1[t]] - \theta_3 \theta_5 \cos[x_3[t]] \cos[x_1[t] + x_3[t]]) x_6[t] + (-\theta_4 \cos[x_1[t]] (\theta_2 + \theta_3 \cos[x_3[t]]) + \theta_5 (\theta_1 + \theta_3 \cos[x_3[t]]) \cos[x_1[t] + x_3[t]]) x_8[t] \right\},$$

x6'[t] ==

$$\frac{1}{\theta_1 \theta_2 - \theta_3^2 \cos[x_3[t]]^2} \left\{ (-\theta_1 \theta_2 + \theta_3^2 \cos[x_3[t]]^2) x_5[t] + \right.$$


$$\left. 2 \theta_3 \sin[x_3[t]] (-((\theta_2 + \theta_3 \cos[x_3[t]]) x_2[t] + \theta_2 x_4[t]) x_6[t] + ((\theta_1 + \theta_2 + 2 \theta_3 \cos[x_3[t]]) x_2[t] + (\theta_2 + \theta_3 \cos[x_3[t])) x_4[t]) x_8[t]) \right\},$$

x7'[t] == 
$$16 \theta_3 \left\{ (\theta_1 \theta_2^2 + \theta_3 \cos[x_3[t]] (\theta_1 \theta_2 + \theta_3 \cos[x_3[t]] (3 \theta_2 + \theta_3 \cos[x_3[t]]))) \sin[x_3[t]] + 2 \theta_2^2 \theta_3 \sin[2 x_3[t]] \right\} x_6[t]^2 +$$


$$\theta_3 x_6[t] \left\{ -2 (-2 \theta_1 \theta_2 + \theta_3^2 + \theta_3^2 \cos[2 x_3[t]]) (-2 \theta_5 \cos[x_1[t]] (\theta_3^2 + (-2 \theta_1 \theta_2 + \theta_3^2) \cos[2 x_3[t]]) + 4 \theta_2 (\theta_3 \theta_4 - \theta_1 \theta_5) \sin[x_1[t]] \sin[2 x_3[t]] + \right.$$


$$\left. 6 \theta_3^3 (-2 \theta_1 \theta_2 + \theta_3^2) x_2[t]^2 + 8 \theta_3 \left\{ 2 \theta_1^2 \theta_2^2 - 2 \theta_1 \theta_2 \theta_3^2 + \theta_3^4 \right\} \cos[2 x_3[t]] x_2[t]^2 + 2 \theta_3^3 (-2 \theta_1 \theta_2 + \theta_3^2) \cos[4 x_3[t]] x_2[t]^2 \right\}$$


```

```
x7[t] = (16 o3 ((o1 o2^2 + o3 Cos[x3[t]] (o1 o2 + o3 Cos[x3[t]] (3 o2 + o3 Cos[x3[t]]))) Sin[x3[t]] + 2 o2^2 o3 Sin[2 x3[t]]) x6[t]^2 +
o3 x6[t] (-2 o1 o2 + o3^2 + o3^2 Cos[2 x3[t]]) (-2 o5 Cos[x1[t]] (o3^2 + (-2 o1 o2 + o3^2) Cos[2 x3[t]]) + 4 o2 (o3 o4 - o1 o5) Sin[x1[t]] Sin[2 x3[t]]) +
6 o3^3 (-2 o1 o2 + o3^2) x2[t]^2 + 8 o3 (2 o1^2 o2^2 - 2 o1 o2 o3^2 + o3^4) Cos[2 x3[t]] x2[t]^2 + 2 o3^3 (-2 o1 o2 + o3^2) Cos[4 x3[t]] x2[t]^2 +
2 o2 (8 o1^2 o2^2 - 16 o1 o2 o3^2 + 7 o3^4) Cos[x3[t]] (x2[t] + x4[t])^2 + 3 o2 o3^4 Cos[3 x3[t]] (x2[t] + x4[t])^2 - o2 o3^4 Cos[5 x3[t]] (x2[t] + x4[t])^2 -
4 (o1 + 4 o2) (4 o1 o2 + 3 o3^2) Sin[x3[t]] x8[t] - 16 o3 (4 o2 (o1 + o2) + o3^2) Sin[2 x3[t]] x8[t] - 12 (o1 + 4 o2) o3^2 Sin[3 x3[t]] x8[t] - 8 o3^3 Sin[4 x3[t]] x8[t] +
x8[t] (o3^3 (o3 o4 - o1 o5) Sin[x1[t]] Sin[5 x3[t]] - 2 (8 o1^2 o2^2 - 16 o1 o2 o3^2 + 7 o3^4) Cos[x3[t]] (o1 o5 Cos[x1[t]] + (o1 + o2) o3 x2[t]^2 + 2 o2 o3 x2[t] x4[t] + o2 o3 x4[t]^2) -
3 o3^4 Cos[3 x3[t]] (o1 o5 Cos[x1[t]] + (o1 + o2) o3 x2[t]^2 + 2 o2 o3 x2[t] x4[t] + o2 o3 x4[t]^2) +
o3^4 Cos[5 x3[t]] (o1 o5 Cos[x1[t]] + (o1 + o2) o3 x2[t]^2 + 2 o2 o3 x2[t] x4[t] + o2 o3 x4[t]^2) -
6 o3^3 (-2 o1 o2 + o3^2) (o5 Cos[x1[t]] + o3 (2 x2[t]^2 + 2 x2[t] x4[t] + x4[t]^2)) -
8 o3 (2 o1^2 o2^2 - 2 o1 o2 o3^2 + o3^4) Cos[2 x3[t]] (o5 Cos[x1[t]] + o3 (2 x2[t]^2 + 2 x2[t] x4[t] + x4[t]^2)) -
2 o3^3 (-2 o1 o2 + o3^2) Cos[4 x3[t]] (o5 Cos[x1[t]] + o3 (2 x2[t]^2 + 2 x2[t] x4[t] + x4[t]^2)) + 3 o3^3 Sin[3 x3[t]] (o3 (o3 o4 - o1 o5) Sin[x1[t]] + 4 (2 o1 + 3 o2) x8[t]) +
2 o3^3 Sin[4 x3[t]] (2 o2 (o3 o4 - o1 o5) Sin[x1[t]] + 5 o3 x8[t]) + 2 Sin[x3[t]] ((8 o1^2 o2^2 - o3^4) (-o3 o4 + o1 o5) Sin[x1[t]] + 2 (2 o1 + 3 o2) o3 (4 o1 o2 + 3 o3^2) x8[t]) +
4 o3 Sin[2 x3[t]] (2 o2 (-2 o1 o2 + o3^2) (o3 o4 - o1 o5) Sin[x1[t]] + o3 (2 (2 o1 + o2) (o1 + 4 o2) + 5 o3^2) x8[t])) / (16 (-o1 o2 + o3^2 Cos[x3[t]]^2)^3),
x8[t] = (1 / (o1 o2 - o3^2 Cos[x3[t]]^2)) (-2 o2 o3 Sin[x3[t]] (x2[t] + x4[t]) x6[t] + (-o1 o2 + o3^2 Cos[x3[t]]^2) x7[t] + 2 o3 (o2 + o3 Cos[x3[t]]) Sin[x3[t]] (x2[t] + x4[t]) x8[t]);
```

Cálculo de variables, parámetros y tensor de masas

```
in[ ]:= a = {0.091, 0.048}; (*m, distancia entre el centro de giro y el CM del sólido*)
l = 0.45;
m = {23.902, 3.880}; (*hg, masa del sólido*)
Izz = {1.266, 0.093}; (*kg.m^2, momento de inercia expresado en el CM*)
g = 9.81; (*m.s^-2, aceleración de la gravedad*)
params = {
o1 -> m[[1]] + a[[1]]^2 + Izz[[1]] + m[[2]] + l^2, o2 -> m[[2]] + a[[2]]^2 + Izz[[2]], o3 -> m[[2]] + a[[2]] + l, o4 -> (m[[1]] + a[[1]] + m[[2]] + l) * g,
o5 -> m[[2]] + a[[2]] + g, o6 -> 2.288, o7 -> 0.175
};
q[t_] := {q1[t], q2[t]};
u[t_] := {u1[t], u2[t]};
M = {{o1 + o2 + 2 o3 Cos[q[t] [[2]]], o2 + o3 + Cos[q[t] [[2]]], {o2 + o3 + Cos[q[t] [[2]]], o2}};
IH = 1 / (o1 + o2 - o3^2 * (Cos[q[t] [[2]]]^2) / (M[[2, 2]], -M[[1, 2]], {-M[[1, 2]], M[[1, 1]]}));
```

```
z o3 (-2 o1 o2 + o3^2) Cos[4 x3[t]] (o5 Cos[x1[t]] + o3 (2 x2[t]^2 + 2 x2[t] x4[t] + x4[t]^2)) + 3 o3^3 Sin[3 x3[t]] (o3 (o3 o4 - o1 o5) Sin[x1[t]] + 4 (2 o1 + 3 o2) x8[t]) +
2 o3^3 Sin[4 x3[t]] (2 o2 (o3 o4 - o1 o5) Sin[x1[t]] + 5 o3 x8[t]) + 2 Sin[x3[t]] ((8 o1^2 o2^2 - o3^4) (-o3 o4 + o1 o5) Sin[x1[t]] + 2 (2 o1 + 3 o2) o3 (4 o1 o2 + 3 o3^2) x8[t]) +
4 o3 Sin[2 x3[t]] (2 o2 (-2 o1 o2 + o3^2) (o3 o4 - o1 o5) Sin[x1[t]] + o3 (2 (2 o1 + o2) (o1 + 4 o2) + 5 o3^2) x8[t])) / (16 (-o1 o2 + o3^2 Cos[x3[t]]^2)^3),
x8[t] = (1 / (o1 o2 - o3^2 Cos[x3[t]]^2)) (-2 o2 o3 Sin[x3[t]] (x2[t] + x4[t]) x6[t] + (-o1 o2 + o3^2 Cos[x3[t]]^2) x7[t] + 2 o3 (o2 + o3 Cos[x3[t]]) Sin[x3[t]] (x2[t] + x4[t]) x8[t]);
```

Cálculo de variables, parámetros y tensor de masas

```
in[ ]:= a = {0.091, 0.048}; (*m, distancia entre el centro de giro y el CM del sólido*)
l = 0.45;
m = {23.902, 3.880}; (*hg, masa del sólido*)
Izz = {1.266, 0.093}; (*kg.m^2, momento de inercia expresado en el CM*)
g = 9.81; (*m.s^-2, aceleración de la gravedad*)
params = {
o1 -> m[[1]] + a[[1]]^2 + Izz[[1]] + m[[2]] + l^2, o2 -> m[[2]] + a[[2]]^2 + Izz[[2]], o3 -> m[[2]] + a[[2]] + l, o4 -> (m[[1]] + a[[1]] + m[[2]] + l) * g,
o5 -> m[[2]] + a[[2]] + g, o6 -> 2.288, o7 -> 0.175
};
q[t_] := {q1[t], q2[t]};
u[t_] := {u1[t], u2[t]};
M = {{o1 + o2 + 2 o3 Cos[q[t] [[2]]], o2 + o3 + Cos[q[t] [[2]]], {o2 + o3 + Cos[q[t] [[2]]], o2}};
IH = 1 / (o1 + o2 - o3^2 * (Cos[q[t] [[2]]]^2) / (M[[2, 2]], -M[[1, 2]], {-M[[1, 2]], M[[1, 1]]}));
V = -o4 + Cos[q[t] [[1]]] - o5 + Cos[q[t] [[1]]] + q[t] [[2]]];
```

Pruebas numéricas

Evolución par RMS y función costo

```
in[ ]:= CC = {x1[0.] = 0., x3[0.] = 0., x1[T] = 0.8, x3[T] = 1.0, x2[0.] = 0., x4[0.] = 0., x2[T] = 0., x4[T] = 0};
sol = Table[NDSolve[{PMPeq, ICs} /. params, {x1, x2, x3, x4, x5, x6, x7, x8}, {t, 0, T}], Method -> {"TimeIntegration" -> "ExplicitRungeKutta"} // AbsoluteTiming,
```

Calculo de par, J, potencia

Cálculo del índice de desempeño

```

inf:= Jpmp = Table[{0.1+i, 0.5*NIntegrate[{u51[[i, 1]], u51[[i, 2]]}.(u51[[i, 1]], u51[[i, 2]]), {t, 0, 0.1+i}], {i, 1, 12}];
Jpmp315 = Table[{0.1+i+1.4, 0.5*NIntegrate[{u52[[i, 1]]^2+(u52[[i, 2]]^2, {t, 0, 0.1+i+1.4}], {i, 1, 2}];
Jpmp324 = Table[{0.1+i+2.3, 0.5*NIntegrate[{u53[[i, 1]]^2+(u53[[i, 2]]^2, {t, 0, 0.1+i+2.3}], {i, 1, 4}];
    
```

Cálculo de los pares RMS

```

inf:= RMSpmp = Table[{0.1+i, Sqrt[NIntegrate[{u51[[i, 1]], u51[[i, 2]]}.(u51[[i, 1]], u51[[i, 2]]), {t, 0, 0.1+i}]/(0.1+i)], {i, 1, 12}];
RMSpmp15 = Table[{0.1+i+1.4, Sqrt[NIntegrate[{u52[[i, 1]], u52[[i, 2]]}.(u52[[i, 1]], u52[[i, 2]]), {t, 0, 0.1+i+1.4}]/(0.1+i+1.4)], {i, 1, 2}];
RMSpmp24 = Table[{0.1+i+2.3, Sqrt[NIntegrate[{u53[[i, 1]], u53[[i, 2]]}.(u53[[i, 1]], u53[[i, 2]]), {t, 0, 0.1+i+2.3}]/(0.1+i+2.3)], {i, 1, 4}];
    
```

Cálculo de la potencia RMS

```

potRMSy3 = Table[{0.1+i, Sqrt[NIntegrate[{u51[[i, 1]]*x2[t] /. sol[[i, 2, 1]]^2+(u51[[i, 2]]*x4[t] /. sol[[i, 2, 1]]^2, {t, 0, 0.1+i}]/(0.1+i)], {i, 1, 12}];
potRMSy31516 = Table[{0.1+i+1.4, Sqrt[NIntegrate[{u52[[i, 1]]*x2[t] /. sol2[[i, 2, 1]]^2+(u52[[i, 2]]*x4[t] /. sol2[[i, 2, 1]]^2, {t, 0, 0.1+i+1.4}]/(0.1+i+1.4)], {i, 1, 2}];
potRMSy324 = Table[{0.1+i+2.3, Sqrt[NIntegrate[{u53[[i, 1]]*x2[t] /. sol3[[i, 2, 1]]^2+(u53[[i, 2]]*x4[t] /. sol3[[i, 2, 1]]^2, {t, 0, 0.1+i+2.3}]/(0.1+i+2.3)], {i, 1, 4}];
    
```

Tablas del tiempo de cálculo

```

cpuTimePMP = Table[{0.1+i, sol[[i, 1]], {i, 12}];
cpuTimePMP1516 = Table[{0.1+i, sol2[[i, 1]], {i, 2}];
cpuTimePMP24 = Table[{0.1+i, sol3[[i, 1]], {i, 4}];
    
```

Gráficas

Curvas

```

inf:= TCPMPyA = ListLinePlot[{cputimePMPy3, cputimePMPy315, cputimePMPy324}, PlotStyle -> {{Gray, Thickness[0.010]}, {Gray, Thickness[0.010]}, {Gray, Thickness[0.010]}},
PlotLegends -> Placed[{"PMP ya"}, {0.50, .90}], FrameLabel -> {"Tiempo prescrito T (s)", "Tiempo cómputo CPU (s)"}, PlotRange -> {{0.5, 2.5}, {0, 179.0}},
PlotTheme -> "Detailed"]

PARMPyA = ListLinePlot[{parPMPy3, parPMPy315, parPMPy324}, PlotStyle -> {{Gray, Thickness[0.010]}, {Gray, Thickness[0.010]}, {Gray, Thickness[0.010]}},
PlotLegends -> Placed[{"PMP ya"}, {0.50, .90}], FrameLabel -> {"Tiempo prescrito T (s)", "Torque RMS (Nm)"}, PlotRange -> {{0.5, 2.5}, {0, 24}], PlotTheme -> "Detailed"]

IDESMPyA = ListLinePlot[{Jpmpy3, Jpmpy315, Jpmpy324}, PlotStyle -> {{Gray, Thickness[0.010]}, {Gray, Thickness[0.010]}, {Gray, Thickness[0.010]}},
PlotLegends -> Placed[{"PMP ya"}, {0.50, .90}], FrameLabel -> {"Tiempo prescrito T (s)", "Desempeño"}, PlotRange -> {{0.5, 2.5}, {0, 149.0}}, PlotTheme -> "Detailed"]

POTMPyA = ListLinePlot[{potRMSy3, potRMSy31516, potRMSy324}, PlotStyle -> {{Gray, Thickness[0.010]}, {Gray, Thickness[0.010]}, {Gray, Thickness[0.010]}},
PlotLegends -> Placed[{"PMP ya"}, {0.50, .90}], FrameLabel -> {"Tiempo prescrito T (s)", "Potencia RMS (W)"}, PlotRange -> {{0.5, 2.5}, {0, 54.0}}, PlotTheme -> "Detailed"]

(*Posición del eslabon en 0.5, 2.5, 5 y 10 segundos *)
POSPMPyA1 = Show[
Plot[x1[t] /. sol[[5, 2]], {t, 0, .5}, PlotStyle -> {Gray, Thickness[0.010]},
FrameLabel -> {"Tiempo prescrito T (s)", "Posición de la articulación 1 (rad)"}, PlotTheme -> "Detailed", PlotLegends -> Placed[{"PMP ya"}, {0.7, .9}]],

Plot[x1[t] /. sol[[10, 2]], {t, 0, 1.0}, PlotStyle -> {Gray, Thickness[0.010]},
Plot[x1[t] /. sol2[[1, 2]], {t, 0, 1.5}, PlotStyle -> {Gray, Thickness[0.010]},
Plot[x1[t] /. sol3[[1, 2]], {t, 0, 2.4}, PlotStyle -> {Gray, Thickness[0.010]},
PlotRange -> {{0, 2.5}, {-6, .8}}]

POSPMPyA2 = Show[
Plot[x3[t] /. sol[[5, 2]], {t, 0, .5}, PlotStyle -> {Gray, Thickness[0.010]},
FrameLabel -> {"Tiempo prescrito T (s)", "Posición de la articulación 2 (rad)"}, PlotTheme -> "Detailed", PlotLegends -> Placed[{"PMP ya"}, {0.5, .9}]],

Plot[x3[t] /. sol[[10, 2]], {t, 0, 1.0}, PlotStyle -> {Gray, Thickness[0.010]},
Plot[x3[t] /. sol2[[1, 2]], {t, 0, 1.5}, PlotStyle -> {Gray, Thickness[0.010]},
Plot[x3[t] /. sol3[[1, 2]], {t, 0, 2.4}, PlotStyle -> {Gray, Thickness[0.010]},
PlotRange -> {{0, 2.5}, {-2.6, 4.1}}]

(*Torque del eslabon en 0.5, 2.5, 5 y 10 segundos *)
PARMPyA1 = Show[
Plot[u51[[5, 1]], {t, 0, .5}, PlotStyle -> {Gray, Thickness[0.010]},
FrameLabel -> {"Tiempo prescrito T (s)", "Torque de la articulación 1 (Nm)"}, PlotTheme -> "Detailed", PlotLegends -> Placed[{"PMP ya"}, {0.5, .9}]],

Plot[u51[[10, 1]], {t, 0, 1.0}, PlotStyle -> {Gray, Thickness[0.010]},
Plot[u52[[1, 1]], {t, 0, 1.5}, PlotStyle -> {Gray, Thickness[0.010]},
Plot[u53[[1, 1]], {t, 0, 2.4}, PlotStyle -> {Gray, Thickness[0.010]},
PlotRange -> {{0, 2.5}, {-21, 36}}]

PARMPyA2 = Show[
Plot[u51[[5, 2]], {t, 0, .5}, PlotStyle -> {Gray, Thickness[0.010]},
FrameLabel -> {"Tiempo prescrito T (s)", "Torque de la articulación 2 (Nm)"}, PlotTheme -> "Detailed", PlotLegends -> Placed[{"PMP ya"}, {0.5, .9}]],

Plot[u51[[10, 2]], {t, 0, 1.0}, PlotStyle -> {Gray, Thickness[0.010]},
Plot[u52[[1, 2]], {t, 0, 1.5}, PlotStyle -> {Gray, Thickness[0.010]},
Plot[u53[[1, 2]], {t, 0, 2.4}, PlotStyle -> {Gray, Thickness[0.010]},
PlotRange -> {{0, 2.5}, {-10, 23}}]
Export[NotebookDirectory[] <> "2GDL_Tiempo_de_calculo_ya.pdf", TCPMPyA];
Export[NotebookDirectory[] <> "2GDL_Par_ya.pdf", PARMPyA];
Export[NotebookDirectory[] <> "2GDL_Power_ya.pdf", POTMPyA];
Export[NotebookDirectory[] <> "2GDL_Position_x1_0.5_2.4_4.9_seg_ya.pdf", POSMPyA1];
Export[NotebookDirectory[] <> "2GDL_Position_x3_0.5_2.4_4.9_seg_ya.pdf", POSMPyA2];
Export[NotebookDirectory[] <> "2GDL_u1_0.5_2.7_4.9_seg_ya.pdf", PARMPyA1];
Export[NotebookDirectory[] <> "2GDL_u2_0.5_2.7_4.9_seg_ya.pdf", PARMPyA2];
    
```