

Universidad Autónoma de Baja California

Facultad de Ingeniería
Área de Posgrado



MODELADO Y CONTROL DE ROBOTS MANIPULADORES EN CADENA CINEMÁTICA ABIERTA

TESIS

que presenta para obtener el grado de Maestro en Ciencias e Ingeniería

ADRIÁN FLORES MENDOZA

DIRECTOR DE TESIS

Dr. RAÚL RASCÓN CARMONA

Diciembre de 2023, Mexicali, Baja California

AGRADECIMIENTOS

Quiero agradecer a todos los que hicieron posible el desarrollo de este trabajo de investigación, los maestros que me dieron clases y que sus consejos me sirvieron mucho para tener ideas claras y consejos que fomentaron mi creatividad. A la administración de posgrado de la Universidad Autónoma de Baja California que me facilitó mucho las instalaciones para poder ir a trabajar y estudiar. Especialmente quiero agradecer al Doctor Raúl Rascón Carmona, mi director de tesis, maestro y mentor, por todo el apoyo que me ha brindado, de manera profesional, así como, personal, gracias por creer en mí. También quiero agradecer a mi familia: mi papá, mi mamá, y mis dos hermanas, porque siempre han estado ahí para mí, dándome motivación para seguir adelante y cumplir mis sueños.

Por último, y más importante, quiero agradecer a Dios porque sé que sin él, todo esto no sería posible.

Resumen

Este trabajo de investigación para el desarrollo de la tesis, tiene dos enfoques, el primero es el modelado de robots manipuladores completamente actuados en cadena cinemática abierta utilizando la metodología de Euler-Lagrange, posteriormente se desarrolló el diseño e implementación de algoritmos de control para seguimiento de trayectorias junto con un algoritmo para estimar perturbaciones externas acotadas. Cabe destacar que para el algoritmo de control de seguimiento de trayectorias, se omite el uso de observadores, filtros o algoritmos diferenciadores. Se analiza la estabilidad en lazo cerrado utilizando herramientas del teorema de Lyapunov.

El segundo enfoque de esta investigación es el desarrollo de un programa hecho en MATLAB donde se han podido verificar los resultados teóricos a través de simulaciones numéricas acompañadas de una animación, la cual permite al usuario una visualización parcial del comportamiento del sistema. Por otro lado, el programa también permite ver en una serie de gráficas los siguientes elementos: torque, posición, velocidad, error, retrato de fase y el comportamiento del algoritmo estimador de perturbaciones externas.

Abstract

In this research work for the development of the thesis, has two approaches, the first is in the modeling of fully actuated manipulator robots in open kinematic chain using the Euler-Lagrange methodology, then developed the design and implementation of control algorithms for trajectory tracking along with an algorithm for estimating bounded external disturbances. It should be noted that for the trajectory tracking control algorithm, the use of observers, filters or differentiating algorithms is omitted. Closed-loop stability is analyzed using tools from Lyapunov's theorem.

The second focus of this research is on the development of a program made in MATLAB where the theoretical results have been verified through numerical simulations accompanied by an animation, which allows the user a partial visualization of the system behavior. On the other hand, the program also allows seeing in a series of graphs the following elements: torque, position, velocity, error, phase portrait and the behavior of the external disturbance estimator algorithm.

Índice general

AGRADECIMIENTOS	I
Resumen	II
Abstract	III
Índice general	V
Índice de figuras	VII
Índice de tablas	VIII
Índice de abreviaturas	IX
Introducción	1
Planteamiento del Problema	1
Justificación de la Investigación	2
Objetivos	3
1. Estado del arte	4
1.1. Introducción	4
1.2. Antecedentes históricos	7
1.3. Definición de robótica	9
1.4. Clasificación de robots	9
1.5. Robots manipuladores	10
1.5.1. Morfología de un robot manipulador	11
1.5.2. Clasificación de robots manipuladores	12
1.5.3. Componentes de un robot manipulador	15
1.6. Conceptos de robótica	17
1.6.1. Cinemática y dinámica de robots manipuladores	19
1.6.2. Dinámica del robot	20
1.6.3. Beneficios de realimentación de sistemas	21
1.6.4. Espacio de operación y espacio articular	22
2. Antecedentes teóricos	23
2.1. Marco Teórico	23
2.2. Preliminares Matemáticos	25
2.2.1. Matrices	25
2.2.2. Jacobiano	26
2.2.3. Modelado en espacio de estados	27
2.2.4. Valores Propios	27
2.2.5. Funciones y matrices positivas definidas	28
2.3. Teoría estabilidad de Lyapunov	29
2.3.1. Teorema de Sylvester	29

2.3.2.	Estabilidad en el Sentido de Lyapunov	29
2.3.3.	Método directo de Lyapunov	30
2.4.	Dinámica	32
2.4.1.	Euler-Lagrange	32
2.4.2.	Propiedades del modelo dinámico	34
2.5.	Robot de dos grados de libertad	37
2.5.1.	Modelo cinemático directo	37
2.5.2.	Modelo cinemático inverso	38
2.5.3.	Dinámica	39
3.	Desarrollo del problema	43
3.1.	Desarrollo	43
3.1.1.	Diseño del controlador	43
3.1.2.	Análisis de estabilidad	44
3.1.3.	Estimador de perturbaciones	44
3.2.	Metodología para simulación	46
3.2.1.	Generación de trayectoria	46
3.2.2.	Trayectorias cartesianas	49
3.2.3.	Programa MATLAB	61
3.2.4.	Animación	66
3.3.	Programa final	70
3.3.1.	Funciones del programa	71
4.	Análisis de Resultados	85
4.1.	Simulaciones en MATLAB	85
4.1.1.	Trayectoria articular: senos y cosenos	86
4.1.2.	Trayectoria articular: infinito-orejona	88
4.1.3.	Trayectoria cartesiana: corazón	92
4.1.4.	Análisis trayectorias cartesianas: corazón	98
4.1.5.	Trayectoria cartesiana: Hipocicloide	99
4.1.6.	Análisis trayectorias cartesianas: Hipocicloides	105
4.1.7.	Trayectoria cartesiana: Figuras de Lissajous	106
4.1.8.	Experimento físico	109
5.	Conclusiones	111
	Bibliografía	112

Índice de figuras

1.1. Clasificación de robots	9
1.2. Brazo humano y brazo robótico [20]	11
1.3. Tipos de articulaciones [18]	11
1.4. Configuración cartesiana [21]	12
1.5. Configuración cilíndrica [18]	12
1.6. Configuración antropomórfica [21]	13
1.7. Configuración esférica [21]	13
1.8. Configuración SCARA [21]	14
1.9. Configuración paralela [21]	14
1.10. Partes de un robot manipulador	15
1.11. Tipos de motores	16
1.12. Retroalimentación de sistema	21
1.13. Espacio cartesiano y Espacio articular	22
2.1. Diagrama robot de dos grados de libertad con dos articulaciones rotacionales [36]	37
3.1. Trayectoria articular con función seno y coseno	47
3.2. Trayectoria cartesiana: Figura en forma de corazón	48
3.3. Epicloide. $a=0.3$; $b=0.1$; $h=0.2$	50
3.4. Hipotrocoide. $R=0.5$; $r=0.3$; $d=0.5$	51
3.5. Epicloide. $r_1=0.3$; $r_2=0.1$	52
3.6. Nefroide. $r_1 = 0.25$; $r_2 = 0.125$	53
3.7. Cardioide. $r_1 = 0.1$; $r_2 = 0.1$	54
3.8. Deltoide. $r_1 = 0.6$; $r_2 = 0.2$	55
3.9. Astroide. $r_1 = 0.4$; $r_2 = 0.1$	56
3.10. Hipocicloide, $k=3.8$. $r_1 = 0.6$; $r_2 = 0.15789$	56
3.11. Hipocicloide, $k=5.5$. $r_1 = 0.42$; $r_2 = 0.07636$	57
3.12. Espiral de Arquimides. $a = 0.01$	57
3.13. Lemniscata de Geronó.	58
3.14. Curva de Lissajous. $a = 1$; $b = 4$	59
3.15. Curva de Lissajous. $a = 1$; $b = 5$	59
3.16. Curva de Lissajous. $a = 2$; $b = 3$	60
3.17. Curva de Lissajous. $a = 5$; $b = 6$	60
3.18. Trayectoria cartesiana: Figura en forma de círculo	62
3.19. Trayectoria cartesiana: Figura en forma de espiral	63
3.20. Trayectoria cartesiana: Figura hipocicloide	63
3.21. Trayectoria cartesiana: Figura hipocicloide en forma de estrella	64
3.22. Trayectoria cartesiana: Figura en forma de flor	64
3.23. Trayectoria cartesiana: Figura en forma de flor	65
3.24. Animación generada a partir de una trayectoria cartesiana	68
3.25. Animación generada a partir de una trayectoria articular	69
3.26. Robot SCARA de UABC	70
3.27. Animación vista en plano X-Y	82

3.28. Gráfica de posiciones y velocidades articulares	83
3.29. Gráficas de torque, errores y perturbaciones	83
3.30. Gráfica de plano fase	84
4.1. Animación vista en plano X-Y	86
4.2. Gráfica de posiciones y velocidades articulares	87
4.3. Gráficas de torque, errores y perturbaciones	87
4.4. Trayectoria articular: infinito-orejona	88
4.5. Trayectoria articular: infinito-orejona, posición y velocidad	89
4.6. Trayectoria articular: infinito-orejona, torque, error y perturbacion	89
4.7. Trayectoria articular: infinito-orejona, plano fase articulación 1	90
4.8. Trayectoria articular: infinito-orejona, plano fase articulación 2	90
4.9. Trayectoria cartesiana: Corazón caso global	92
4.10. Trayectoria cartesiana: Corazón, caso global, posición y velocidad	93
4.11. Trayectoria cartesiana: Corazón, caso global, torque, error y perturbacion	93
4.12. Trayectoria cartesiana: Corazón, caso global, plano fase articulación 1	94
4.13. Trayectoria cartesiana: Corazón, caso global, plano fase articulación 2	94
4.14. Trayectoria cartesiana: Corazón caso local	95
4.15. Trayectoria cartesiana: Corazón, caso local, posición y velocidad	96
4.16. Trayectoria cartesiana: Corazón, caso local, torque, error y perturbación	96
4.17. Trayectoria cartesiana: Corazón, caso local, plano fase articulación 1	97
4.18. Trayectoria cartesiana: Corazón, caso local, plano fase articulación 2	97
4.19. Comparación de trayectorias cartesianas: corazón.	98
4.20. Trayectoria cartesiana: Hipocicloide.	99
4.21. Trayectoria cartesiana: Hipocicloide, posiciones y velocidades	100
4.22. Trayectoria cartesiana: Hipocicloide, torque, errores y perturbaciones	100
4.23. Trayectoria cartesiana: Hipocicloide, plano fase articulación 1	101
4.24. Trayectoria cartesiana: Hipocicloide, plano fase articulación 2	101
4.25. Trayectoria cartesiana: Hipocicloide caso local.	102
4.26. Trayectoria cartesiana: Hipocicloide, caso local, posiciones y velocidades.	103
4.27. Trayectoria cartesiana: Hipocicloide, caso local, torques, errores y perturbaciones.	103
4.28. Trayectoria cartesiana: Hipocicloide, caso local, retrato de fase articulación 1.	104
4.29. Trayectoria cartesiana: Hipocicloide, caso local, retrato de fase articulación 2.	104
4.30. Comparación de trayectorias cartesianas: Hipocicloides	105
4.31. Trayectoria cartesiana: Figura de Lissajous	106
4.32. Trayectoria cartesiana: Figura de Lissajous, posición y velocidad	107
4.33. Trayectoria cartesiana: Figura de Lissajous, torque, errores y perturbaciones	107
4.34. Trayectoria cartesiana: Figura de Lissajous, plano fase articulación 2	108
4.35. Trayectoria cartesiana: Figura de Lissajous, plano fase articulación 2	108
4.36. Experimento en robot SCARA	109
4.37. Gráfica de posición	109

Índice de tablas

1.1. Tipos de sensores para robots [18]	16
4.1. Parámetros robot manipulador SCARA de UABC	85
4.2. Ganancias del controlador	85
4.3. Parámetros de la trayectoria articular simple.	86
4.4. Parámetros de la trayectoria articular: infinito-modificado.	88
4.5. Parámetros de trayectoria: corazón.	92
4.6. Parámetros de trayectoria: corazón en 4 cuadrante	95
4.7. Parámetros de trayectoria: hipocicloide	99
4.8. Parámetros de trayectoria: hipocicloide cuarto cuadrante.	102
4.9. Parámetros de trayectoria: Figura de Lissajous.	106

Índice de abreviaturas

GDL	Grados de libertad
R	Campo de números reales
R^n	Espacio vectorial de dimensión n con números reales
$\ A\ $	Norma de la matriz A
EDO	Ecuación diferencial ordinaria
$ A $	Determinante de la matriz A
A^T	Matriz traspuesta
$A < 0$	Matriz definida negativa
$A \leq$	Matriz semidefinida negativa
$V(x) > 0$	Forma cuadrática definida positiva
$V(x) < 0$	Forma cuadrática definida negativa
$V(x) \leq 0$	Forma cuadrática semidefinida negativa
\mathcal{L}	Lyapunov
\mathcal{J}	Jacobiano

Introducción

Planteamiento del problema

Un problema de control particular en el área de robots manipuladores es el diseño de algoritmos de control para seguimiento de trayectorias con solo mediciones de posición, en donde , algunos trabajos previos sobre control para seguimiento de trayectorias en robots manipuladores son [1] [2] [3] [4] [5]. Comúnmente en el diseño de controladores para seguimiento de trayectorias en robots manipuladores, la velocidad articular debe estar disponible para retroalimentación, lo cual puede ser una desventaja si no se cuenta con sensores de velocidad disponibles. Esta situación se puede resolver utilizando un observador de velocidad, o un diferenciador. Cuando se utiliza un observador para reconstruir la velocidad, pueden surgir algunos problemas como, utilización de muchos recursos computacionales, problemas para sintonizar las ganancias, una señal de respuesta transitoria no adecuada, además de que puede ser en algunos casos difícil de implementar el observador, principalmente cuando no se tienen los parámetros exactos del robot. Por otra parte, los observadores pueden formar lazos de control en el software que se implemente, por lo que el sistema en lazo cerrado se puede volver inestable bajo ciertas condiciones, ver [6].

Por ende, el principal problema a resolver es el de desarrollar un controlador para el problema de seguimiento trayectoria, en donde solo se usen mediciones de posición como retroalimentación, omitiendo el uso de diferenciadores u observadores antes mencionados.

Hay algunos trabajos recientes sobre el control para seguimiento de trayectoria en robots manipuladores. Por ejemplo, un controlador para resolver el problema de seguimiento sin mediciones de velocidad, en un sistema de un grado de libertad modelado utilizando el método de Euler-Lagrange se presenta en [7]. También en [5], se aborda el problema de control para seguimiento de salida en sistemas tipo Euler-Lagrange a través de una ley de control dinámica. En [8], el control de seguimiento utilizando medición de salida fue abordado y resuelto para una clase de sistemas de Euler-Lagrange con un grado relativo arbitrariamente alto. En [16], el conocido control PD más compensación se utilizó para resolver el problema de control de seguimiento de salida en sistemas tipo Euler-Lagrange, solamente se requería que el sistema tuviera fricción viscosa sin importar que tan grande o pequeño fuera el cociente de fricción.

Justificación de la investigación

Este proyecto nace de la necesidad del desarrollo y aplicación de controladores robustos, que sean capaces de sobrellevar o mitigar los efectos causados por incertidumbres paramétricas, perturbaciones externas. Por otra parte, se considera solamente medición de salida: en este caso es la posición de los eslabones del robot manipulador. Para lograr un error de seguimiento de trayectoria igual a cero no se considera el desarrollo e implementación de observadores. Todas las prestaciones antes mencionadas hacen de este controlador un diseño en el estado del arte. El diseño de un controlador con estas características conllevará a una optimización con respecto a los algoritmos de control previamente existentes. Por otra parte, es posible publicar en una revista de alto impacto el diseño e implementación de un control robusto para robots manipuladores diseñado en el estado del arte.

Objetivos

En este trabajo de investigación se pretende desarrollar un controlador para robots manipuladores en cadena cinemática abierta, modelando su dinámica con la metodología de Euler-Lagrange. Dicho controlador omitirá el uso de observadores, filtros o algoritmos diferenciadores. Se demostrará estabilidad del sistema en lazo cerrado utilizando herramientas de Lyapunov véase [8] [9] [10].

Así mismo, se desarrollará un programa en MATLAB en el que se pueda simular y ver una animación del robot manipulador en su forma esquemática. Las trayectorias a implementarse serán obtenidas en espacio de operación cartesiana y se convertirán a espacio articular.

Objetivos específicos

- Obtención de trayectorias en espacio articular y espacio cartesiano.
- Diseñar el algoritmo de control considerando que el modelo tiene incertidumbre paramétrica. Por otro lado, el robot manipulador está sujeto a perturbaciones externas específicas.
- Diseño de una prueba que permita tener estabilidad en lazo cerrado.
- Desarrollo de programa en MATLAB en donde el usuario pueda observar una animación del comportamiento del robot manipulador.
- Hacer simulaciones numéricas en MATLAB del robot manipulador.

Capítulo 1 | Estado del arte

1.1. Introducción

Los robots ya están entre nosotros. Desde que se ideó la creación de un mecanismo capaz de poder hacer tareas más eficientes y que facilitaran el trabajo, el ser humano se ha esmerado en poder desarrollar artefactos útiles principalmente para uso en producción y mejora de procesos en el sector industrial, haciendo así que estas máquinas creadas puedan agilizar la manufactura de diversos productos, especialmente los que tienen alta demanda de consumo.

Conforme se ha avanzado en la teoría de control, diseño mecánico y en ciencias de la computación sobre como poder modelar los procesos físicos, adaptándolos abstractamente a modelos matemáticos, ha permitido el desarrollo de diversos tipos de robots capaces de auxiliar, resolver y llevar a cabo tareas que puedan ser peligrosas, monótonas o que exijan mucho a una persona.

Partiendo de un ejemplo sencillo son los robots aspiradores, que está dentro de la categoría de robot de servicio, que como bien lo dice su nombre, son robots aspiradores que están programados para recorrer la casa aspirando, con un sistema de detección de obstáculos y algunos con mapeo de trayectoria, utilizados principalmente en hogares. Estos robots no son grandes, midiendo algunos 13.45 x 13.39 x 3.65 pulgadas y con precios que pueden ir desde los 200.0 dólares americanos hasta los 1000.0 dólares americanos. Según “Markets and Markets”, se espera que la industria de los robots aspiradores crezca de 2.1 billones de dólares americanos en 2018 a 4.4 billones de dólares americanos para 2023.[11]

El término robot proviene de “robota” que significa “trabajo” en checo; introducido por el dramaturgo Karel Capek en 1921, en su obra R.U.R (Robots Universales Rossum) en donde hace alusión a una compañía que construye lo que ahora conocemos como andróides, que son creados para ayudar a la humanidad, pero que después se van contra de los humanos. Desde entonces se ha empleado no solo en el ámbito científico, también ha sido musa para la composición de música, como por ejemplo “Mr. Robot” de Styx y “Robot Rock” de Daft Punk, siento en “Mr Robot” una alusión a un robot que tiene características humanas “ I’m not a robot without emotions, I’m not what you see”. En la literatura también ha tenido impresión, tal es el caso de los reconocidos libros del escritor ruso Isaac Asimov con su “Saga de la Fundación”, “Yo, Robot” uno de los ejemplares más conocidos de dicha saga, la cual también tuvo su adaptación cinematográfica en el 2004; en dicha trilogía se dan las famosas tres leyes de la robótica:

- 1era Ley: Un robot no hará daño a un ser humano ni, por inacción, permitirá que un ser humano sufra daño.
- 2da Ley: Un robot debe cumplir las órdenes dadas por los seres humanos, a excepción de aquellas que entren en conflicto con la primera ley.

- 3era Ley: Un robot debe proteger su propia existencia en la medida en que esta protección no entre en conflicto con la primera o con la segunda ley.

En el “séptimo arte” también ha sido de mucha influencia la robótica, hay numerosos filmes en donde un robot es protagonista, antagonista o con papel secundario, pero de gran influencia en las tramas de dichos filmes. Algunos de los argumentos más comunes que se les atribuyen a los robots en las películas es la inteligencia artificial que pueden desarrollar y como esta evoluciona para ser los robots quienes controlen a los humanos y no viceversa; dando así también temas de estudio filosóficos; ¿algún día podrían tener conciencia los robots? Si bien la robótica ha influenciado como inspiración para algunas ramas del arte, en forma de precisamente de eso, de inspiración y no como algo tangible que un proyecto robótico haga en pro del arte.

Fragmento película” Yo, robot”:

- Detective: Los seres humanos tienen sueños, los perros también, pero tú no... Solo eres una máquina, una imitación de la vida. ¿Puedes componer una sinfonía? ¿Puedes convertir un lienzo en una hermosa obra de arte?
- Sonny(robot): ¿Puede usted?

Una de las áreas donde está causando mayor impacto la robótica es en el área de medicina, donde hay brazos manipuladores (controlados siempre por un doctor especialista) que pueden operar con mayor exactitud y de mínima invasión. Un ejemplo de este tipo de operaciones fue en una pancreatoduodenectomía, donde a un paciente se le hizo esta operación haciendo uso de 4 brazos robóticos con un tiempo quirúrgico de 8 horas (4:45 horas para resección y 3:15 horas para reconstrucción). Una cirugía robótica oncológica renal y suprarrenal es otro de los bastos ejemplos de cirugías que se pueden hacer con la ayuda de robots.[12] El uso de estos tipos de robots colaboradores ha ido en aumento y cada vez son dotados de más herramientas como inteligencia artificial (I.A.). El uso de robots en medicina no es de solo uso en quirófanos, sino también en laboratorios, donde estos son programados para hacer pruebas y análisis de laboratorio, automatizando así procesos estandarizados, haciendo estas tareas de una forma más eficaz. También hay exoesqueletos robóticos para tratar pacientes donde requieren fisioterapia para recobrar movimientos de extremidades. En este año, 2020, en hospitales se hizo uso de robots de compañía para mitigar soledad en pacientes de COVID-19 que tenían que estar aislados, así mismo para hacer interacción con estos pacientes sin tener una exposición directa y evitar contagios.[13] [14] [15]

En la industria del comercio también se ha hecho el uso de robots de servicio, como en cantinas, donde los robots fungen como “bartenders” y preparan las distintas bebidas que los comensales piden, dentro de su programación ya contienen los parámetros para que las cantidades de líquido vertidas sean las adecuadas. También en la preparación de alimentos participan en la elaboración de comidas rudimentarias. Aunque aún se hace investigación sobre esta relación que hay con los robots de servicio y la confianza que los clientes puedan tener al momento de interactuar con estos, especialmente los robots antropomórficos (que tienen forma humana). También hay implicaciones éticas que se deben de considerar al momento del desarrollo y la aplicación de robots en diferentes áreas de industria; implicaciones como el hecho de la reducción de empleos, aunque esto también fomenta la preparación y capacitación técnica. Otra posible complicación que pueda tener el desarrollo de robots de servicio es que podría reducir la interacción social. La producción en masa de productos “artesanales”. [16] [17]

Hoy en día la robótica va tomando más relevancia en nuestra vida cotidiana. Los robots de servicio van en aumento y estos se van desarrollando dependiendo a las necesidades que los seres humanos van necesitando, así como las actividades que no quieren hacer, no pueden hacer (cuando hay exposición a materiales radioactivos, exploración en desastres naturales) y que necesitan tener más un estándar de confianza de repetibilidad. Ante la idea que la robótica llegue a acaparar el trabajo de una persona, aún falta mucho desarrollo para que pueda haber una rentabilidad a corto y mediano plazo, ya que un robot que sea robusto para la realización de muchas tareas puede llegar a ser muy costoso tanto por fabricación como por el mantenimiento que vaya a requerir.

Por otro lado, el empleo de robots en actividades rudimentarias como mantener limpia una habitación puede dar mayor beneficio que perjudicial, ya que el tiempo empleado en esa actividad quedaría relevada y se pudiera ocupar en otra actividad de beneficio personal. La robótica debe ser desarrollada y empleada para beneficio de la sociedad, ante las posibles implicaciones negativas que la robótica pueda ocasionar, también pudieran aplicarse leyes que protejan el bienestar económico de la sociedad.

1.2. Antecedentes históricos

En 1948, R. C. Goertz del Laboratorio Nacional Argonne logró uno de los primeros progresos en robótica al crear un sistema pionero de la operación a distancia de un mecanismo. Este sistema estaba compuesto por un dispositivo maestro-esclavo mecánico, diseñado para manipular objetos radioactivos. Posteriormente, Goertz sustituyó gran parte de los elementos mecánicos por electrónicos. En los años 60, junto con la industria nuclear, la industria submarina comenzó a mostrar interés por el uso de manipuladores remotos. Este interés se unió a la industria espacial en la década de 1970. La necesidad de automatizar las líneas de producción, reemplazando a los operadores por programas informáticos que controlan los movimientos de los manipuladores, ha llevado al concepto de robótica, incitado por los requerimientos de automatizar las cadenas de manufactura [18].

En 1957 se concedió la primera patente para un mecanismo robótico, la cual fue presentada en marzo, tres años antes por el C.W. Kenward, quien era un inventor del Reino Unido; esta patente fue seguida por George S. Devol, el hombre que sentó las bases de los robots industriales modernos, a quien se le ocurrió la idea de una máquina que pudiera programarse para mover objetos. En 1956, Devol presentó su idea a Joseph F. Engelberger (1925-2015), manager de ingeniería de la división aeroespacial de la empresa Manning Maxwell y Moore en Stanford, Connecticut, y ávido lector de Asimov. En 1956, presentó la idea a Joseph F. Engelberger (1925-2015), un ávido lector de Asimov y director técnico de la división aeroespacial de Manning Maxwell y Moore en Stanford, Connecticut. Juntos, Devol y Engelberger empezaron a investigar aplicaciones industriales para sus máquinas, formando Unified Control Corporation, que luego se convirtió en Unimation. En 1961, instalaron su primer Unimate en la planta de General Motors en Trenton, Nueva Jersey, para moldear por inyección. Otras empresas importantes, como AMF, han seguido fabricando máquinas similares.

Posteriormente, por razones de negocios, se comenzó a llamar robots a estas máquinas diseñadas para mover partes de forma “flexible” o “global”, aunque su apariencia no se asemejaba al humanoide típico de los robots de los libros, historietas o el cine. Este acertado cambio de denominación, contribuyó en gran medida a la propagación e integración de los robots por parte de las empresas de manufactura, presentando un concepto de innovación y progreso tecnológico.

Kawasaki empezó la producción de robots parecidos al modelo “Unimate”, cuando Joseph Engelberger visitó Japón en 1968 para establecer acuerdos con dicha compañía. Nissan, en 1972, estableció la primera asociación enfocada a la robótica en la historia, la Asociación de Robótica Industrial de Japón (JIRA), lo hizo que el desarrollo de la robótica en Japón superara rápidamente al de los Estados Unidos. Dos años después, se fundó el Instituto de Robótica de América (RIA), que para 1984 cambiaría su título a la Asociación de Industrias Robóticas, pero mantuvo las mismas siglas (RIA) [19].

A fines de la década de 1960 e inicios de la de 1970 se sentaron las bases para la investigación en robótica en las instituciones académicas. Los robots móviles que fueron los primeros en hacerse, tenían limitados grados de libertad, se empezaron a diseñar en los departamentos de investigación de recién creación para el desarrollo de inteligencia artificial, como el robot Shakey de la Stanford University en el Stanford Research Institute. El robot PUMA (Programmable Universal Manipulation Arm), uno de los robots industriales más populares y que más se ha utilizado en entornos de investigación e industrial, diseñado y desarrollado por Victor Scheinman en el mismo centro de investigación en que se encontraba el robot Shakey, y fue el quién estableció las bases para el diseño de robots manipuladores que fueran con accionamiento eléctrico-electrónico y controlados por computadoras, y que posteriormente llevaría al

diseño, manufactura y comercialización. Por otro lado, Europa tuvo un despertar más tardío, aunque igualmente importante. El robot IRb6, primer robot construido por una compañía de Suecia llamada ASEA en 1973, funcionaba con un accionamiento completamente eléctrico, un año después construirían el IRb60. Después, ya como ABB, se volvió en una de las principales empresas a nivel global en la producción de robots antropomórficos, y Suecia una de las naciones con más robots por cada habitante. La Federación Internacional de Robótica fue creada en Estocolmo, Suecia en 1980. En Alemania, la compañía KUKA fue precursora en la producción e implementación de robots manipuladores, empresa pionera en implementar una cadena de producción para soldadura provisionada exclusivamente con máquinas robóticas. La disposición inicial de los robots se basaba en las configuraciones conocidas como esférica y antropomórfica, las cuales resultan particularmente útiles para tareas de manipulación. En la Universidad Yamanashi de Japon, en 1982, el catedrático Makino crea el término robot SCARA (Selective Compliance Assembly Robot Arm), con la finalidad de desarrollar un autómatas con un mínimo de articulaciones, variando entre tres o cuatro, un costo controlado y una estructura enfocada en el ensamblaje de piezas. Un avance significativo en la tecnología para la creación de robots es el sistema de “Direct drive” (accionamiento directo en español), inventado en 1981 por la Universidad Carnegie Mellon. Este sistema permite el desarrollo de un nuevo tipo de motores que se puedan conectar directamente a las articulaciones, sin requerir reductores de velocidad, posibilitando una mayor velocidad y precisión en los movimientos [18].

Aparte de los ambientes industriales y académicos, en la década de los setenta, los autómatas empezaron a ser utilizados de manera común en otros ámbitos, como en el sector referente al espacio o el marino, empleando primordialmente tecnologías de control remoto. Es relevante mencionar los vehículos sumergibles creados por la marina de los Estados Unidos a principios de los años sesenta y las exploraciones espaciales al satélite natural de la Tierra y al planeta Marte hacia el final de los años sesenta y al comienzo de la década de los setenta, y que culminaron en 1981 con el brazo mecánico que integro la NASA en los transbordadores de los Estados Unidos.

1.3. Definición de robótica

La robótica es una ciencia multidisciplinaria que se encarga de la investigación y desarrollo de sistemas robóticos, los cuales se diseñan para diversas aplicaciones científicas, industriales, comerciales y domésticas. La ISO 8373:2012 define a la robótica como la ciencia y prácticas que se encarga de diseñar, manufacturar y programar robots. Del mismo modo, ISO define a un robot como un mecanismo programable actuado con dos o más ejes con grados de libertad, que se puede mover dentro de su ambiente para realizar tareas previstas.

Las ramas de ingeniería en las que se apoya la robótica son electrónica, mecánica, control automático, sistemas computacionales que en conjunto forman un sistema robótico.

De las características más remarcables que podemos encontrar en los robots son versatilidad (potencialmente pueden hacer diferentes funciones), flexibilidad, autonomía (con el uso de sensores que le brinden información sobre su entorno anidada a técnicas de inteligencia artificial puede codificar trayectorias optimizadas para evadir objetos, por poner un ejemplo), precisión y repetitividad.

1.4. Clasificación de robots

Para este proyecto de tesis se clasificarán los robots en dos categorías, la primera es para Robot Manipuladores, la segunda para robots móviles. Dentro de los robots móviles a la vez hay subcategorías: terrestres, acuáticos y aéreos. Para la categoría de robots manipuladores, también llamados robots industriales, se pueden clasificar a la vez según su aplicación, la configuración antropomórfica que tengan, el número de eslabones que tengan, etc.

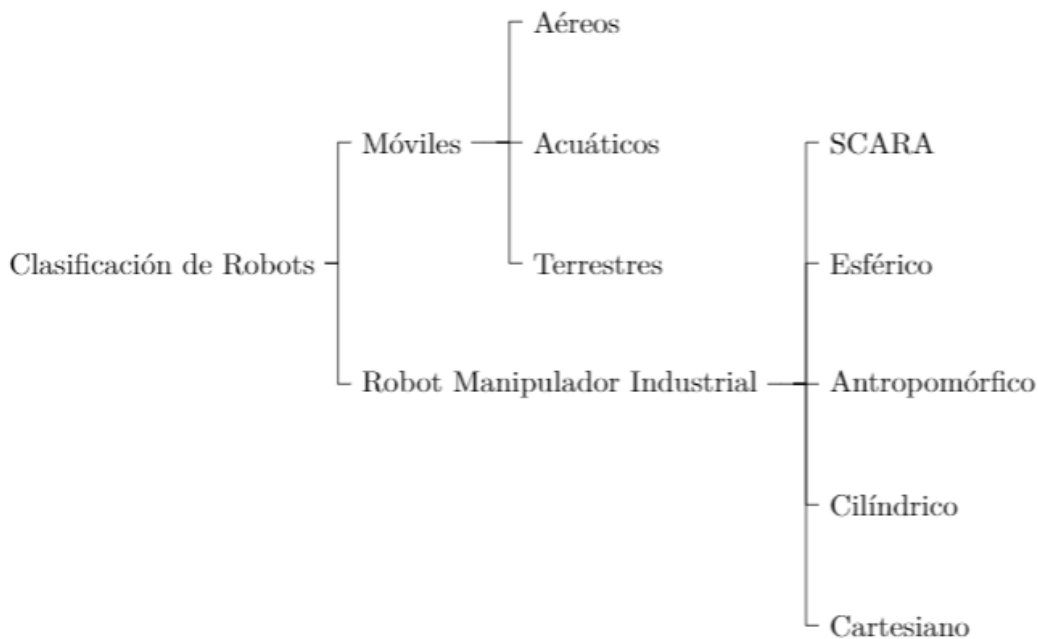


Figura 1.1: Clasificación de robots

1.5. Robots manipuladores

Hay varias definiciones para los robots industriales manipuladores, la ISO 8373, lo define a un robot manipulador como: una máquina manipuladora con varios grados de libertad controlada automáticamente, reprogramable y de múltiples usos, pudiendo estar en un lugar fijo o móvil para su empleo en aplicaciones industriales. Otra definición ampliamente usada es la de *Robotics Industries Association* : un robot industrial es un manipulador multifuncional, reprogramable, capaz de mover materias, piezas, herramientas o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas.

La idea principal de entender las definiciones de un robot manipulador en cadena cinemática abierta es que una de las características que precisamente lo pueden llegar a definir es que siguen trayectorias programadas y controladas, Y en todas las definiciones se coincide en que un robot industrial es capaz de manejar objetos y sus movimientos son controlados mediante la programación de tareas.

Algunas aplicaciones en las que los robots manipuladores son usados en la industria son: robots de soldadura, pintura, aplicación de adhesivos, corte láser, para perforar, inspección con visión por computador, ensamble de partes mecánicas, ensamble de electrónicos, atornillado cableado y fijación de componentes, para empaquetado, paletizado, clasificación de materiales, pruebas y análisis de laboratorio (CMM), entre otros.

Beneficios de los robots manipuladores

Algunos beneficios que se pueden destacar es la repetitividad y precisión que se puede obtener; debido a esto hay una alta productividad, ya que no hay paros, descansos, cambios de turno, a cómo lo habría con un operador humano. Otro factor vinculado a la alta productividad es la velocidad y calidad, cualidades que son de gran ayuda para poder hacer cálculos de OEE (Overall Equipment Effectiveness) por poner un ejemplo. Otro factor a considerar es la reducción de accidentes de trabajo por “factor humano”, es decir, en los entornos de trabajo donde hay un alto riesgo para el ser humano, se pudieran usar equipos robóticos.

1.5.1. Morfología de un robot manipulador

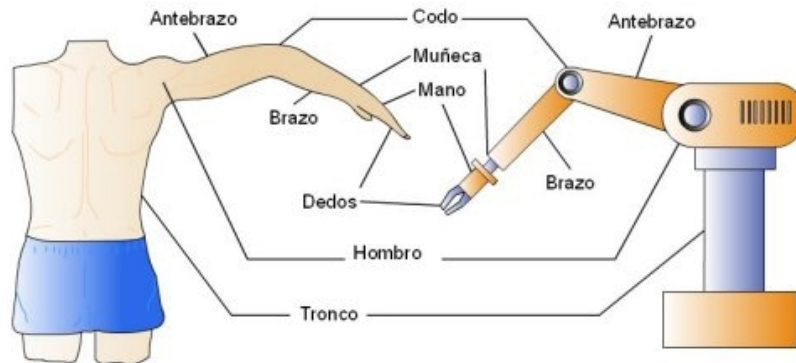


Figura 1.2: Brazo humano y brazo robótico [20]

Analógicamente, se puede hacer una comparación de un brazo de la anatomía humana a un robot manipulador industrial (como se puede observar en 1.2), o como también se le suele decir brazo robótico, esto debido a las similitudes que estos poseen en su apariencia. Un brazo robótico consta esencialmente de cuatro partes: cuerpo, brazo, muñeca y actuador final o gripper.

Esencialmente, un robot se compone por una secuencia de eslabones conectados mediante articulaciones, los cuales posibilitan el desplazamiento. El desplazamiento que se genera en las articulaciones puede tener distintas configuraciones, hay seis tipos de articulaciones: esférica, planar, cilíndrica, prismática y rotacional.

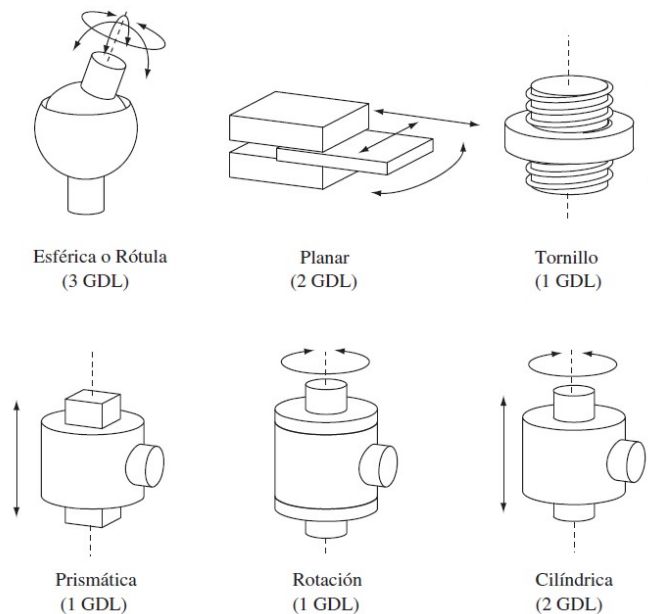


Figura 1.3: Tipos de articulaciones [18]

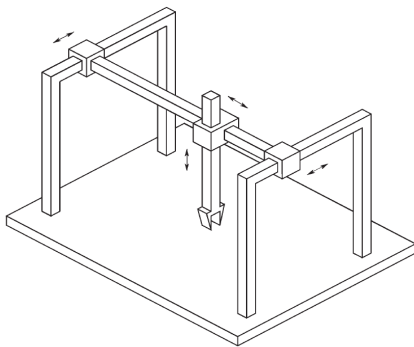
Dichas articulaciones, nos permitirán tener grados de libertad, estos grados de libertad se tienen debido a que cada movimiento de cada articulación es independiente al anterior; el número de grados de libertad de un robot generalmente es igual al número de articulaciones que lo conforman.

1.5.2. Clasificación de robots manipuladores

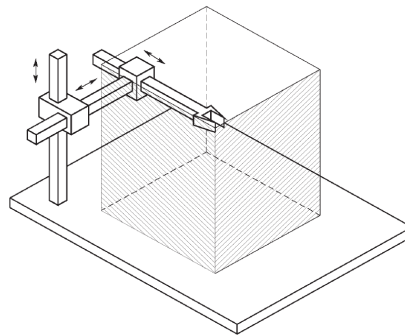
Las características antes mencionadas permiten tener diversas configuraciones de robots industriales que cumplen con diferentes funciones, sus capacidades son distintas y debido a sus características se usan para diferentes áreas de la industria.

Una de las clasificaciones más ampliamente usadas es cuando se clasifican por:

Cartesiano: tiene al menos tres articulaciones prismáticas (X, Y, Z) y las variables que producen estas articulaciones son las coordenadas cartesianas del actuador final.



(a) Robot cartesiano



(b) Robot manipulador cartesiano

Figura 1.4: Configuración cartesiana [21]

Cilíndrico: Su primera articulación es de revolución, produciendo así una rotación alrededor de su base; la segunda y tercera articulación suelen ser prismáticas.

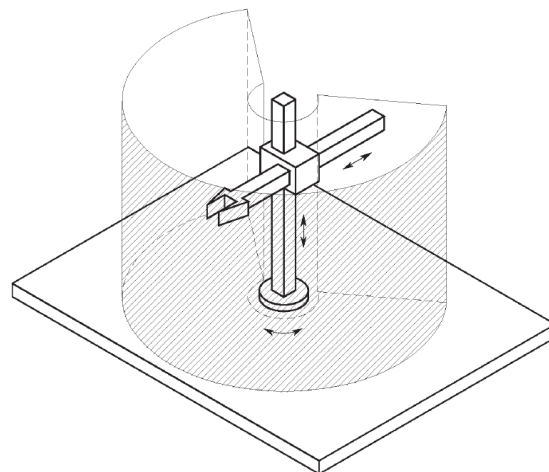


Figura 1.5: Configuración cilíndrica [18]

Angular o antropomórfico: Esta configuración tiene tres articulaciones de rotación. La forma posición del actuador final se especifica en coordenadas angulares.

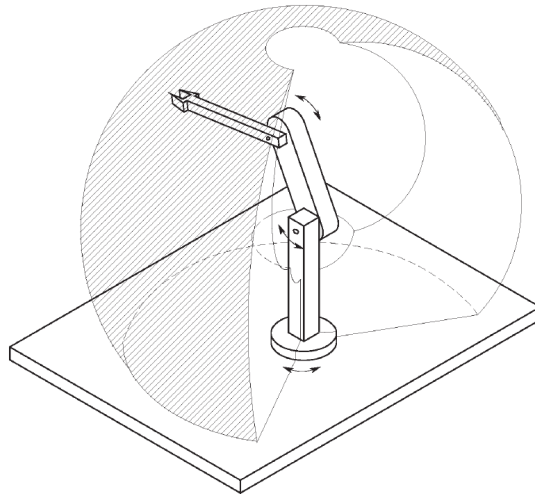


Figura 1.6: Configuración antropomórfica [21]

Polar o Esférico: Tiene dos articulaciones con movimiento rotacional y una prismática; las variables de las articulaciones expresan la posición en coordenadas polares.

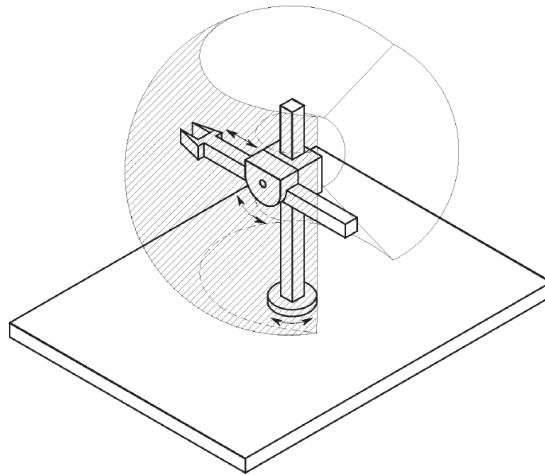


Figura 1.7: Configuración esférica [21]

SCARA (Selective Compliant Articulated Robot for Assembly): configuración diseñada principalmente para ensambles. Por lo regular, las primeras dos articulaciones son verticales de revolución moviéndose en un plano horizontal y la tercera articulación se mueve según su eje vertical.

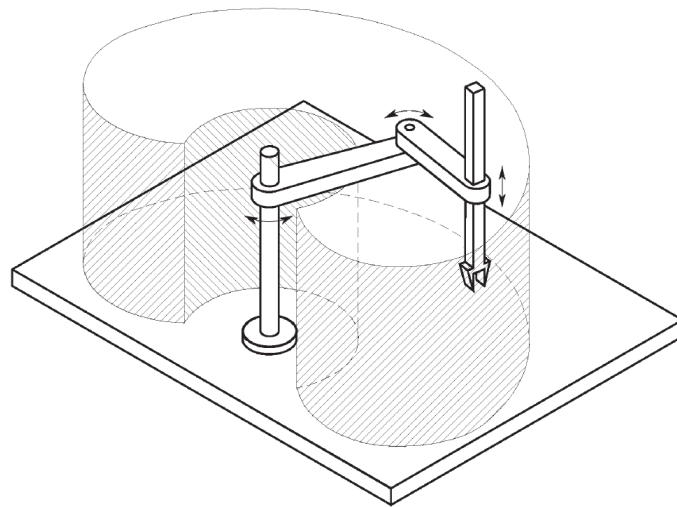


Figura 1.8: Configuración SCARA [21]

Paralelo: Consta de al menos tres brazos unidos a una base, una de sus principales funciones es para la selección y clasificación.

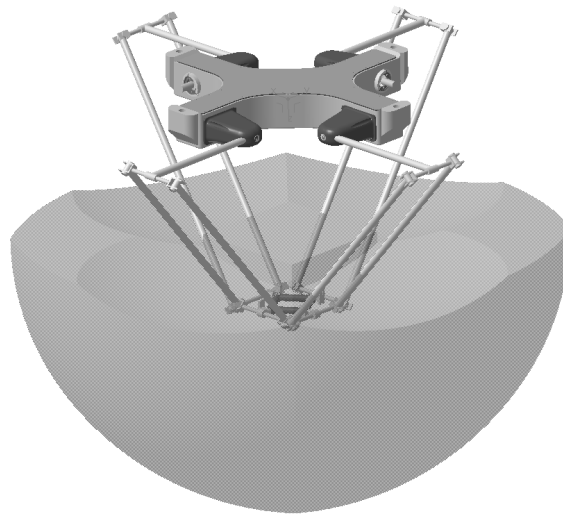


Figura 1.9: Configuración paralela [21]

1.5.3. Componentes de un robot manipulador

Un robot manipulador consta de diversos componentes necesarios para su correcto funcionamiento como lo son: motores, sensores, microcontroladores, microprocesador, armazón (que corresponde a la estructura interna y externa del robot).

Para fines prácticos de esta investigación, un sistema robótico se dividirá en 3 subsistemas: subsistema mecánico, subsistema electrónico y subsistema de control.

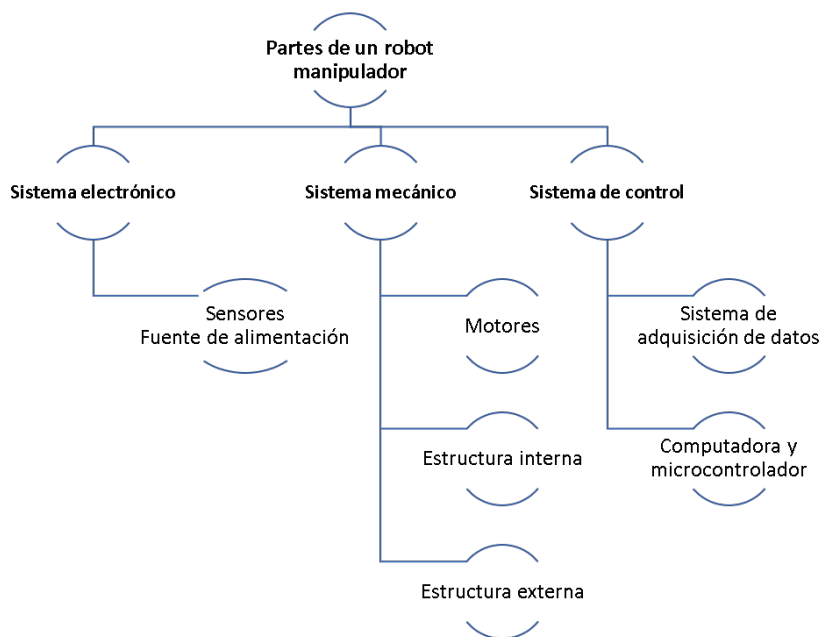


Figura 1.10: Partes de un robot manipulador

Los motores son como los músculos de los robots. Dentro de las características que deben de presentar los motores usados en brazos robóticos son: “baja inercia, alta relación de potencia-peso, posibilidad de sobrecarga y entrega de pares de torsión de impulsos, capacidad para desarrollar altas aceleraciones, amplios rangos de velocidad, alta precisión de posicionamiento, buena precisión de rastreo de trayectorias y posicionamiento, entre otros [22].”

Los motores eléctricos presentan atributos de control, simplicidad y la precisión de los accionamientos eléctricos ha hecho que sean los más empleados en los robots industriales contemporáneos.

Usualmente, se regulan mediante velocidades de referencia. Estas suelen ser monitoreadas por un circuito de retroalimentación de velocidad analógico que se completa con un componente electrónico específico. Encima de este circuito de velocidad se instala otro de ubicación, donde las referencias son producidas por la unidad de control a partir de la discrepancia entre la ubicación deseada y la actual [18].

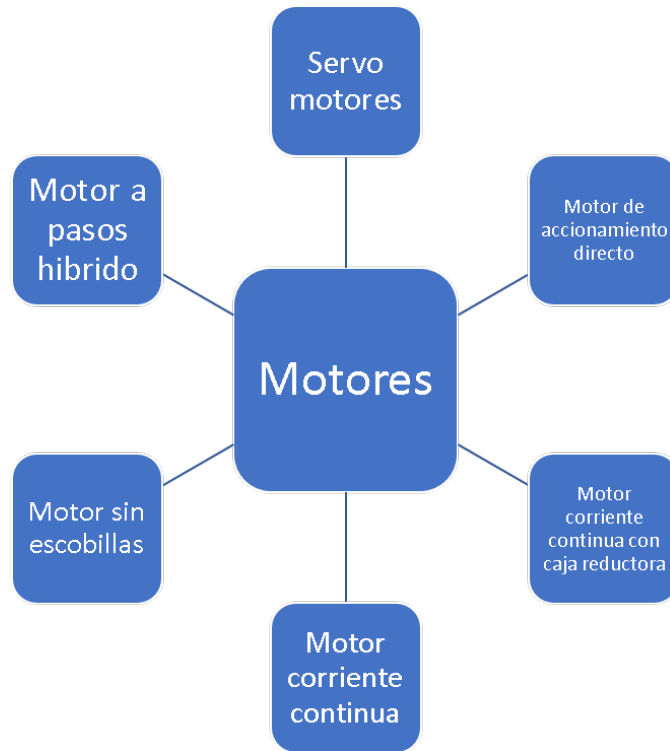


Figura 1.11: Tipos de motores

Los robots necesitan de dispositivos que le puedan indicar las características físicas de su entorno, haciendo una analogía con un ser humano que a través de órganos específicos puede percibir estímulos externos e internos del medio en el que se rodea y a lo que se le llama sentido, un robot manipulador puede tener implementados diversos sensores. La tabla 1.1 tiene algunos de los sensores que más comúnmente se pueden encontrar en los robots.

Tabla 1.1: Tipos de sensores para robots [18]

Medida	Tipo
Proximidad	Inductivo Capacitivo Sonda Hall Fotoeléctrico Ultrasónico Final de carrera
Posición Analógicos	Potenciómetros Resolver Regla magnética Desplazamiento lineal
Posición Digitales	Digitales Encoders Absolutos Encoders Incrementales Regla Óptica
Velocidad	Tacómetros

1.6. Conceptos de robótica

La teoría de control moderna se basa en el análisis en el dominio temporal de los sistemas de ecuaciones diferenciales, el cual se basa en el modelo del sistema real que se quiere controlar; mientras que la estabilidad del sistema depende del error entre el sistema real y su modelo, para esto se tiene que llevar a cero la dinámica del error, es decir, el punto de equilibrio del sistema en lazo cerrado es el origen y se tratan de llevar las trayectorias a ese punto; entonces se diseña el sistema de control definiendo en primer lugar el rango de posibles errores y después diseñando el controlador de forma que, si el error del sistema está en dicho rango, el sistema de control diseñado permanezca estable [23].

Sistemas: Es un conjunto de elementos que colaboran entre sí para lograr una meta específica. Un sistema no se restringe exclusivamente a los sistemas materiales. Este término puede ser utilizado para referirse a fenómenos abstractos y cambiantes.

Perturbación: es una señal que suele influir de forma negativa en el rendimiento de un sistema. Si dicha señal surge desde dentro del sistema, se le conoce como interno, en caso contrario, se le denomina perturbación externa.

Control retroalimentado: La retroalimentación de control se refiere a una acción que tiene como objetivo disminuir la disparidad entre la salida de un sistema y una entrada de referencia, especialmente en presencia de perturbaciones, y esto se logra considerando cuidadosamente dicha disparidad.

Modelación: obtención de un modelo matemático de la planta u objeto de control. Es encontrar una ecuación o serie de ecuaciones que describan el comportamiento de la planta.

Una planta grupo de componentes de un sistema que trabajan juntos para llevar a cabo una tarea específica.

Control: es el empleo de ciertas variables dentro de un sistema para conseguir una acción o efecto deseado.

Cadena cinemática abierta: es el conjunto o subconjunto de elementos de un sistema que están conectados entre sí. Una cadena cinemática se considera abierta cuando la última articulación no se encuentra unido a los demás.

El objetivo principal del control de robots manipuladores es la de alcanzar un punto deseado previamente establecido, para el cual se puede derivar como dos problemas: control de regulación (o también llamado como control de posición) y control de trayectoria, el cual es el tema de estudio de este trabajo de investigación. En el control de trayectoria, primero se determina una trayectoria en el espacio de trabajo y el problema consistirá (en término coloquiales) en hacer que el extremo del manipulador pase por la trayectoria establecida, lo más cerca posible, es decir, que el error sea mínimo. Para resolver dicho se puede dividir en etapas para su mejor comprensión:

- Planeación de trayectoria.
- Generación de trayectoria.

- Diseño de controlador.

Con planeación de trayectoria lo que se busca es determinar una curva en el espacio de trabajo que lleve de un punto inicial al punto final, el extremo del robot sin que toque obstáculo alguno.

La generación de trayectoria se basa en la parametrización del tiempo de cada uno de los puntos obtenidos en la planeación de la trayectoria; la trayectoria deseada es obtenida en coordenadas de espacio de trabajo para luego con cinemática inversa obtener los valores articulares deseados. Finalmente, con el diseño del controlador se busca hacer que el robot siga dicha trayectoria con mínimo error.

Diseño de controlador se refiere al desarrollo un algoritmo que permita poder controlar el sistema en cuestión por medio de una ecuación que se adapta a los requerimientos establecidos.

En este trabajo, se desarrolló un programa en el cual se pudiera implementar un modelo matemático que sería sea capaz de controlar al robot manipulador, diseñado para el problema de “control de seguimiento de trayectoria”. Dicho programa consta también de una simulación. Para lograr la simulación se necesitó del uso de planeación de itinerario y generación de trayectoria, por lo cual, todos los cálculos que se desarrollan en los siguientes capítulos, se deben de tomar en cuenta para entender y ejecutar el programa.

1.6.1. Cinemática y dinámica de robots manipuladores

Cinemática es la parte de la física que se encarga del estudio del movimiento de un cuerpo sin considerar las fuerzas que la originan, mientras que la dinámica trata del estudio de las fuerzas o pares que originan dicho movimiento[24].

La cinemática directa de robots manipuladores se refiere al estudio analítico del movimiento del robot con respecto a un sistema de referencia cartesiano fijo, $\Sigma(x, y, z)$ relacionando la dependencia que existe entre las coordenadas articulares o generalizadas $q \in \mathbb{R}^n$, sus parámetros geométricos y las coordenadas cartesianas $[x, y, z]^T \in \mathbb{R}^3$ y de orientación $[\theta, \phi, \psi]^T \in \mathbb{R}^{\neq}$ del extremo final del robot.

La cinemática inversa radica en establecer los valores que deben asumir las variables de las coordenadas de las articulaciones $q = [q_1, q_2, \dots, q_n]^T$ correspondientes a una orientación y posición específicas del efector final [25].

Denavit y Hartenberg propusieron un método sistemático para describir y representa la geometría espacial de los elementos de una cadena cinemática, y en particular de un robot, con respecto a un sistema de referencia fijo. Este método utiliza una matriz de transformación homogénea para describir la relación espacial entre dos elementos rígidos adyacentes, reduciéndose el problema cinemático directo a encontrar una matriz de transformación homogénea 4x4 que relacione la localización espacial del extremo del robot con respecto al sistema de coordenadas de su base. Hay otras formas de encontrar solución al problema de cinemática inversa, algunos son:

- Denavit-Hartenberg (Transformaciones Homogéneas)
- Método geométrico
- Cuaterniones
- Métodos iterativos

En este trabajo de tesis se optó por resolver la cinemática inversa usando el método geométrico.

Además, la cinemática del robot también se enfoca en determinar las correspondencias dadas por la velocidad de los desplazamientos de las articulaciones y las del extremo final. Dicha correspondencia es resultado del método diferencial expresado mediante la matriz jacobiana (véase capítulo 2: Preliminares matemáticos) (Más sobre el problema cinemático directo e inverso se omitirá en esta tesis, ya que no es su caso de estudio).

Cinemática diferencial

La cinemática diferencial directa es la derivada con respecto al tiempo de la cinemática directa

$$\frac{d}{dt}[x, y, z, \theta, \phi, \psi]^T = \begin{bmatrix} v \\ w \end{bmatrix} = \frac{d}{dt}f_R(q) = J(q)\dot{q} \quad (1.1)$$

Este tema se abordará más adelante en la sección de preliminares matemáticos.

1.6.2. Dinámica del robot

La dinámica nos proporcionará las ecuaciones que describen el comportamiento dinámico de un robot, lo que también llamaremos: modelo dinámico del robot, y es importante por los siguientes aspectos [22]:

1. Puede emplearse para idear estrategias de control
2. También se puede utilizar para simular por ordenador un sistema robótico, permitiendo prever su comportamiento.
3. El análisis dinámico de un robot proporciona información sobre todas las fuerzas y momentos de reacción necesarios para diseñar y determinar el tamaño de las articulaciones y actuadores.

La dinámica se centra en la interacción entre las fuerzas que influyen sobre un objeto y el desplazamiento que se produce en él. Por consiguiente, el modelo dinámico de un robot pretende comprender la conexión entre el desplazamiento del robot y las fuerzas que intervienen en él.

Esta relación se logra por medio del antes nombrado “modelo dinámico”, que establece la correspondencia matemática entre:

1. La posición del robot determinada por las variables de sus articulaciones o las coordenadas del efector final, junto con sus magnitudes derivadas: velocidad y aceleración.
2. Las fuerzas y momentos aplicados en las articulaciones (o en el efector final del robot)
3. Longitud, masa, inercia de cada articulación, es decir, de los parámetros del robot.

Deducir las ecuaciones para un mecanismo con uno o dos grados de libertad no es demasiado complejo, no obstante, a medida que el número de grados de libertad se incrementa, el enfoque y la obtención del diseño dinámico se vuelve más enredado. Por este motivo, no siempre es posible obtener un modelo dinámico expresado de una forma cerrada, esto es, mediante una serie de ecuaciones, normalmente de tipo diferencial de segundo orden, cuya integración permita conocer un movimiento determinado[18]. Por lo que el modelo ha de ser abordado iterativamente, empleando una serie de acciones numéricas para su resolución.

El desarrollo del modelo dinámico de un robot representa uno de los retos más difíciles en el campo de la robótica, por lo que a menudo se ha pasado por alto. No obstante, dicho modelo es esencial para alcanzar los siguientes objetivos:

1. Simulación del desplazamiento del robot manipulador.
2. Análisis y diseño del armazón y configuración física del robot.
3. Cálculo de las dimensiones de los motores.
4. Análisis y diseño del control dinámico del robot.

1.6.3. Beneficios de realimentación de sistemas

Los sistemas con retroalimentación presentan diversas características importantes, ya que el comportamiento total del sistema está condicionado por la interdependencia de sus componentes. La retroalimentación permite estabilizar sistemas inestables, mejorar la robustez ante variaciones del comportamiento de algunos componentes del sistema y reducir las perturbaciones externas no medibles.

Existen varias razones para introducir deliberadamente la retroalimentación en procesos que funcionan correctamente. El principal beneficio de la retroalimentación no puede deducirse sin considerar las incertidumbres en el comportamiento de la planta.

Hay dos motivos por los cuales la salida y la planta y , para una entrada u , como se aprecia en la figura (1.12), producen una trayectoria que no se puede determinar completamente de antemano. En primer lugar, la dinámica de la planta no se conoce ni se encuentra completamente especificada, por lo que no puede considerarse como una descripción precisa, y en lugar se describe más como una aproximación. En segundo lugar, las perturbaciones (que no se conocen) pueden contribuir al comportamiento del sistema. Para un caso como este, la perturbación d y la señal de control u tienen como salida a y , la cual usualmente es ambigua, ya que puede referirse a cualquier elemento en un conjunto de posibles señales de perturbación.

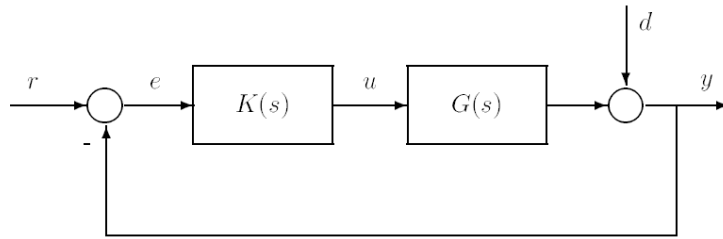


Figura 1.12: Retroalimentación de sistema

Incertidumbres de este tipo se presentan en mayor o menor grado en los desafíos de control, ya que debido al desarrollo del diseño y modelado, se debe tener en cuenta que el sistema a controlar está desconectado de su contexto. Por lo general, es complicado determinar los fenómenos que deben ser tomados en cuenta como parte de la planta o como vínculos entre la planta y su entorno y cuáles deben omitirse. En estos casos, es significativo saber que las principales características de los sistemas retroalimentados tienen una pequeña dependencia de dichas incertidumbres. La robustez frente a elementos de dinámicas no modelados y perturbaciones también puede considerarse como una propiedad estructural de los sistemas retroalimentados. Dado que la señal de control u se calcula por el controlador en función del valor actual de y , es necesario tener en cuenta el efecto de las señales de control anteriores y las perturbaciones.

La característica esencial de los sistemas de retroalimentación se ilustra mediante la comparación de las configuraciones de los sistemas de control en lazo abierto o lazo cerrado. El control en lazo abierto se basa en la precisión del modelo de la planta (y, por lo tanto, en la precisión del hardware), ya que la señal de control se calcula a partir del modelo. Las consecuencias de las dos clases de incertidumbre se muestran completamente como desviaciones de la trayectoria que resulta de la respuesta establecida. En cambio, en el control en lazo cerrado, se compara el valor actual de la salida y con la señal de referencia r , y la entrada a la planta se calcula en

función de la señal de error. Aparte de los datos proporcionados previamente sobre la planta en forma de un modelo matemático, el control retroalimentado utiliza dichos datos sobre el comportamiento actual de la planta y las perturbaciones existentes, lo cual es proporcionado implícitamente mediante la medición del valor de y .

La inclusión de la retroalimentación se debe principalmente a la falta de conocimiento completo del sistema que se desea controlar y al impacto de las perturbaciones externas. De manera más precisa, el empleo de la retroalimentación no se justifica si no hay incertidumbres en el sistema, ya que en sistemas no perturbados, el control en circuito abierto genera una respuesta similar o posiblemente mejor.

1.6.4. Espacio de operación y espacio articular

En la robótica se suele trabajar en dos tipos de referencias espaciales, el espacio operacional (o también llamado espacio cartesiano, de tarea o de trabajo) y en espacio articular (o también llamado espacio de configuración). El espacio cartesiano es más intuitivo visualmente. Este opera mediante coordenadas x , y y z , en referencia al efector final $x_e = [p_e \phi]^T$. Esto es que cuando uno traza una trayectoria, mientras que el espacio articular opera con coordenadas articulares expresadas en grados o radianes $q = [q_1 \dots q_n]^T$.

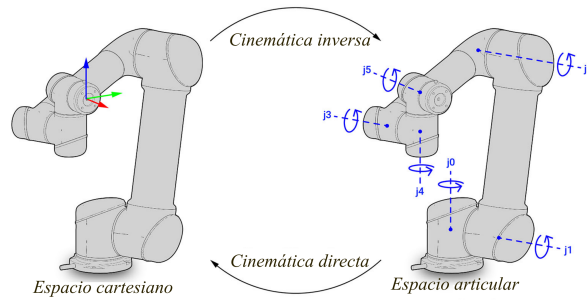


Figura 1.13: Espacio cartesiano y Espacio articular

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \in \mathbb{R}^n$$

$$x = \begin{bmatrix} x \\ y \\ z \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^n$$

Capítulo 2 | Antecedentes teóricos

2.1. Marco Teórico

En los sistemas mecánicos, particularmente en los sistemas robóticos, existe una diferencia entre las plantas reales y los modelos utilizados en el diseño. Esta discrepancia plantea la pregunta de si el controlador diseñado a partir del modelo será efectivo. Para garantizar su eficacia, se ha desarrollado la teoría de control robusto. Esta teoría parte de la premisa de que los modelos utilizados en el diseño del sistema de control tienen errores de modelado, ya sea por falta de más datos o por las características físicas del robot. El objetivo del control robusto es diseñar un controlador que pueda garantizar la estabilidad y el rendimiento de un conjunto de modelos, y que sea resistente a perturbaciones externas. En los dispositivos utilizados para el control y la retroalimentación del sistema, existen imperfecciones que generan incertidumbre, ya que no conocemos los parámetros que afectarán al sistema, y aun si los conocemos, estos pueden variar debido a la no linealidad del sistema.

En el seguimiento de trayectorias de un robot manipulador, el robot en cuestión tiene que tener una retroalimentación de la posición, velocidad y aceleración. La teoría de control que se aplica en sistemas robóticos lo que intenta hacer es que el error de seguimiento sea cero. Hay un problema para el control de robots manipuladores y es el diseño de algoritmos de control para el seguimiento de trayectorias de donde la velocidad articular debe estar disponible para retroalimentación, la cual se obtiene por medio de sensores, los cuales pueden llegar a ser costosos y el ruido en sus mediciones puede llegar a ser considerable. Entonces hay ocasiones en las que es preferible omitir la información brindada por el sensor de velocidad o simplemente remover dicho sensor y que se use solo la posición como medio de medición para el seguimiento de trayectorias.

En [26], se propone un control no lineal PD para resolver el problema de seguimiento de trayectoria para un grado de libertad usando la metodología de Euler Lagrange y encontrar estabilidad asintótica global. [27] usa un control dividido en dos partes, el primero es un control por modos deslizantes y luego un observador para estimar la velocidad, logrando estabilidad asintótica global en el sentido de Lyapunov para un sistema robótico de dos grados de libertad. Antonio Loria en 2013 [28], usa el control de Paden-Panka que tiene una estabilidad asintótica semi global, usando derivadas sucias, pero lo usa para establecer ciertas propiedades globales que servirán para poder usar una función estricta de Lyapunov; así mismo en 2016 [29] propone el uso matemático teórico de las llamadas derivadas sucias como reemplazo de las mediciones de velocidad no presentes, para poder alcanzar estabilidad asintótica global, pero sin tomar en cuenta fuerzas disipativas del sistema y no tomando en cuenta observadores en el controlador.

El uso de observadores también es ampliamente usado como en [30] donde usan un esquema híbrido de fuerza-posición para estimar la fuerza y velocidad del sistema, mediante el uso de un observador integral generalizado (GPI) con alta ganancia. En [31], mediante el uso de observadores, utiliza un control robusto para lograr estabilidad con altas ganancias. En [32], hacen una comparación de cuatro estrategias de control para hacer un seguimiento de error de dinámica del sistema y ver las equivalencias que hay entre ellos, que incluía un GPI, un controlador con filtros, ADRC y SMC, en sistemas de segundo orden, aplicaron derivada sucia

en controladores PID. En [33] presentan un control adaptativo bajo la suposición de que las entradas están delimitadas y establecen estabilidad uniforme asintótica global, que incluye una convergencia en cero en los parámetros de estimación del error, para todo esto, remueven la suposición de que el sistema tiene fricción.

En [34], hicieron experimentación uniendo dos tipos de controladores, a un control robusto le anidaron un control adaptativo para que pudiera mejorar su rendimiento, en comparación a los dos controladores que trabajaron de forma independiente. Así mismo, Víctor Santibáñez, presenta un controlador adaptativo con retroalimentación de salida, probaron que seleccionando la trayectoria tal que si la matriz de regresión está excitada persistentemente y con suficiente fricción viscosa, se puede lograr estabilidad uniforme asintótica global. Un control robusto empleado en [2] fue el uso de funciones vectoriales de Lyapunov que les pudo brindar un procedimiento explícito para las ganancias del controlador. En [35] lograron estabilidad asintótica en un robot SCARA de dos grados de libertad, sin el uso de observadores ni filtros, para esto consideran un estimador adaptativo para la fricción viscosa, el cual también solo necesita de la posición como única medición de retroalimentación.

2.2. Preliminares Matemáticos

2.2.1. Matrices

Una matriz es un arreglo de números que tiene m renglones horizontales y n columnas verticales. $\mathbb{R}^{m \times n}$ será el conjunto de matrices A de dimensión $n \times m$ formada por arreglos de números reales ordenados por m renglones y n columnas:

$$A = a_{ij} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \dots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}$$

Matriz cuadrada

Si la matriz A es cuadrada, significa que tiene el mismo número de renglones y columnas ($n=m$);

Matriz traspuesta

La matriz traspuesta se obtiene intercambiando los renglones y las columnas. La matriz traspuesta de $A = a_{ij} \in \mathbb{R}^{m \times n}$ se representa por $A^T = a_{ij} \in \mathbb{R}^{n \times m}$.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$
$$A^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix}$$

Matriz simétrica y antisimétrica

La matriz A será simétrica si $A = A^T$

$$B = \begin{bmatrix} 1 & 1 & 5 \\ 1 & 2 & 3 \\ 5 & 3 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 5 \\ 1 & 2 & 3 \\ 5 & 3 & 4 \end{bmatrix}^T$$

Mientras que una matriz será antisimétrica, si $A = -A^T \in \mathbb{R}^{n \times n}$.
Unas propiedades importantes a resaltar serán:

- $x^T Ax = 0$, para todo $x \in \mathbb{R}^n$.
- Una matriz A es igual a la suma de la matriz simétrica más la matriz antisimétrica.
 $A = A_{simetrica} + A_{antisimetrica}$
- $A_{simetrica} = \frac{A+A^T}{2}$
- $A_{antisimetrica} = \frac{A-A^T}{2}$

Matriz inversa

Sea una matriz $A \in \mathbb{R}^{n \times n}$ es invertible si su determinante es diferente de 0 y se denota de la siguiente manera:

$$A^{-1} = \frac{adj A}{|A|}$$

Donde:

$adj A$ es la matriz adjunta de A y $|A|$ es el determinante de la matriz.

2.2.2. Jacobiano

Cuando se quiere controlar un robot, es importante contar con un análisis de la dinámica del manipulador, en ellas operan la posición, velocidad y aceleración. Para ello es importante analizar los robots manipuladores desde las propiedades que están relacionadas con las velocidades del efector final y de cualquier punto de dicho manipulador. El jacobiano del robot manipulador es una herramienta matemática que permitirá establecer una relación entre las velocidades de las articulaciones con las velocidades lineales y angulares del efector final; se puede apreciar el jacobiano desde el punto de vista matemático como una transformación de espacios.

La matriz jacobiana, representada por $J \in \forall \mathbb{R}^{m \times n}$, es una matriz formada por derivadas parciales de primer orden, la cual puede escribirse de la siguiente forma:

$$J = \frac{\partial f(x)}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \dots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \quad (2.1)$$

Esto será de mucha utilidad para hacer la transformación de espacio cartesiano a espacio articular.

Si se obtiene el determinante del jacobiano y este es cero, indica que se encuentran singularidades. Las singularidades que aparecen de forma interna en el entorno de trabajo del robot, surgen cuando se produce una alineación de más de dos ejes de las articulaciones del robot; mientras que las singularidades que se presentan externamente al espacio de trabajo del robot, se refieren a cuando el efector final del robot está en un punto fuera del límite del espacio de trabajo, por lo que el robot manipulador no podrá alcanzar y/o desplazarse en la dirección deseada.

La matriz jacobiana inversa proporciona las velocidades articulares a partir de las velocidades en el espacio cartesiano.

$$\begin{bmatrix} \dot{q} \\ \vdots \\ \dot{q}_n \end{bmatrix} = J^{-1} \begin{bmatrix} \dot{x} \\ \vdots \\ \dot{\gamma} \end{bmatrix} \quad (2.2)$$

2.2.3. Modelado en espacio de estados

Se trata de un modelo matemático que representa un sistema físico a través de un grupo de entradas, salidas y variables de estado. Estas últimas están conectadas mediante ecuaciones diferenciales de orden n que se combinan para formar una ecuación diferencial matricial de orden n [23]. Para simplificar la cantidad de entradas, salidas y estados, las variables se expresan en forma de vectores y las ecuaciones algebraicas se escriben en formato matricial.

Un sistema se puede representar de la siguiente manera:

$$\dot{x}(t) = f(x, u, t) \text{ Ecuacion de entrada} \quad (2.3)$$

$$y(t) = g(x, u, t) \text{ Ecuacion de salida} \quad (2.4)$$

Donde:

- x , representa las variables (x_1, x_2, \dots, x_n)
- u , representa las entradas (u_1, u_2, \dots, u_n)
- Si las funciones vectoriales f y g involucran explícitamente el tiempo t , el sistema se denomina sistema variante con el tiempo.

Si se linealizan las ecuaciones de estado 2.3 y 2.4 alrededor del espacio de operación, se tienen las siguientes ecuaciones de estado y de salida linealizadas:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2.5)$$

$$\dot{y}(t) = Cx(t) + Du(t) \quad (2.6)$$

Donde:

- A: se denomina matriz de estado.
- B: matriz de entrada.
- C: matriz de salida.
- D: matriz de transmisión directa.

2.2.4. Valores Propios

Para cada matriz cuadrada $A \in \mathbb{R}^{n \times n}$ existen n valores propios denotados por $\lambda_1\{A\}$, $\lambda_2\{A\}, \dots, \lambda_n\{A\}$, . Los valores propios de la matriz $A \in \mathbb{R}^{n \times n}$ satisfacen:

$$\det[\lambda_i I - A] = 0, \text{ para } i = 1, 2, \dots, n$$

Donde $I \in \mathbb{R}^{n \times n}$ es la matriz identidad de dimensión n .

2.2.5. Funciones y matrices positivas definidas

A continuación se empezará a utilizar la siguiente notación para indicar una matriz definida positiva: $A > 0$.

Una matriz cuadrada $A \in \mathbb{R}^{n \times n}$, (no tiene que ser necesariamente simétrica), es definida positiva si:

$$x^T A x > 0, \text{ para todo } x \in \mathbb{R}^n, \text{ con } x \neq 0 \in \mathbb{R}^n$$

La matriz cuadrada A será definida positiva solamente si sus valores propios son positivos. Para más información sobre esta afirmación, consultar el *Teorema de Sylvester*.

Toda matriz que es simétrica y definida positiva $A = A^T > 0$ es no singular, por tanto, su inversa A^{-1} existe.

Una matriz cuadrada $A \in \mathbb{R}^{n \times n}$, la cual no necesariamente es simétrica, es semidefinida positiva si:

$$x^T A x \geq, \text{ para todo } x \in \mathbb{R}^n$$

Forma cuadrática

Se define como forma cuadrática a la función escalar

$$V(x) = x^T P x \tag{2.7}$$

donde: $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \in \mathbb{R}^n, P = P^T \in \mathbb{R}^{n \times n}$

Se dice que una función $V(x)$ es localmente positiva definida si, dada la región, $\Omega_x = \{x \in \mathbb{R}^n \mid \|x\| < r\}$ se cumple que:

1. $V(x) > 0 \forall x \in \Omega, x \neq 0$
2. $V(x) = 0 \rightarrow x = 0$
3. $V(0) = 0$

La función será definida positiva global cuando cumple con los tres requisitos antes mencionados en todo el espacio de su argumento $x \in \mathbb{R}$.

Una función es localmente positiva semidefinida si:

1. $V(x) \geq 0 \forall x \in \Omega, x \neq 0$
2. $V(0) = 0$

Toda forma cuadrática $V(x) = x^T P x; x \in \mathbb{R}^n, P = P^T \in \mathbb{R}^{n \times n}$ es una función positiva definida globalmente si

1. $\lambda_i(P) > 0, i = 1, \dots, n$
2. $m_{ii}(P) > 0$

2.3. Teoría estabilidad de Lyapunov

Un aspecto a considerar del comportamiento dinámico de los sistemas corresponde al análisis de estabilidad, en el cual un sistema debe de presentar la capacidad para mantenerse en equilibrio o regresar a su estado natural después de haber sido perturbado.

2.3.1. Teorema de Sylvester

Estable que la función $V(x) = x^T Ax > 0$ es definida positiva si y solo si la matriz $A \in \mathbb{R}^{n \times n}$ es definida positiva para cualquiera $x \in \mathbb{R}^n$

Una matriz $A > 0$ es definida positiva si cumple con los siguientes requisitos:

- La matriz A de la función cuadrática, $V(x) = x^T Ax$, debe ser una matriz simétrica $A = A^T$.
- El primer elemento de la matriz a_{11} de $a \in \mathbb{R}^{n \times n}$ debe ser positivo, es decir, $a_{11} > 0$
- Todos los determinantes menores deben ser positivos, consecuentemente el determinante de la matriz es $|A| > 0$

2.3.2. Estabilidad en el Sentido de Lyapunov

La teoría de estabilidad de Lyapunov tiene como principal objetivo estudiar el comportamiento de sistemas dinámicos descritos por ecuaciones diferenciales de la forma:

$$\dot{x}(t) = f(t, x(t)), x(0) = \in \mathbb{R}^n \forall t \geq 0 \quad (2.8)$$

donde el vector $x(t) \in \mathbb{R}^n$ se refiere al estado del sistema dinámico representado en 2.8 y $x(0) \in \mathbb{R}^n$ se denomina la condición inicial o estado inicial.

Una peculiaridad destacable de este procedimiento de estabilidad es que puede ser determinado sin necesidad de resolver la ecuación diferencial que controla el sistema. Por otro lado, el método no proporciona ninguna información sobre la respuesta transitoria o sobre el desempeño del sistema, por ejemplo, si se sobreamortigua o subamortigua o cuanto tiempo se tardará en suprimir una perturbación [22]. A grandes rasgos, para demostrar que un sistema es estable mediante el método de Lyapunov, se deben de seguir los siguientes pasos:

- Proponer una función de energía generalizada $V(x)$ que se llamará función de Lyapunov, cuyas primeras derivadas parciales sean continuas y $V(x) > 0$ para todos los x , excepto $V(0) = 0$.
- Evalúe la primera derivada respecto al tiempo de $V(x)$.
- Verifique si $\dot{V}(x) \propto 0$. Aquí, $\dot{V}(x)$ respecto al tiempo a lo largo de todas las trayectorias del sistema.

La idea es que una función que sea definida positiva, demuestre que siempre disminuye o permanece constante. Por lo tanto, el sistema es estable en el sentido que el tamaño del vector de estado se limite. Cuando $\dot{V}(x)$ es estrictamente menos que cero, el estado converge en forma asintótica hacia el vector cero. Sin embargo, cuando (\dot{x}) desaparece en algunos lugares de la

trayectoria, LaSalle con su teorema, demostró que el sistema aún puede ser asintóticamente estable en ciertas situaciones. Lo más complicado de este método, en muchas ocasiones, es la elección de la función de Lyapunov $V(x)$.

Equilibrio

Un vector constante $x_e \in \mathbb{R}^n$ es un equilibrio del sistema 2.8 si

$$f(t, x_e) = 0 \quad \forall t \geq 0$$

Si la condición inicial es cero 0, entonces satisface:

- $x(t) = x_e \quad \forall t \geq 0$
- $\dot{x}(t) = 0 \quad \forall t \geq 0$

Tradicionalmente, el origen del espacio de estado \mathbb{R}^n , esto es $x = 0 \in \mathbb{R}^n$, es un equilibrio de 2.8. En caso contrario, puede demostrarse que mediante un cambio adecuado de coordenadas, cualquier equilibrio de 2.8 puede trasladarse al origen del espacio de estado.

Estabilidad

El origen $x = 0 \in \mathbb{R}^n$ es un equilibrio estable (en el sentido de Lyapunov) de la ecuación 2.8 si para cada número $\epsilon > 0$ se puede encontrar un número $\delta > 0$, tal que:

$$\|x(0)\| < \delta \rightarrow \|x(t)\| < \epsilon \quad \forall t \geq 0$$

Lo descrito previamente quiere decir que cuando el resultado de una condición inicial acotada también se encuentra acotada, su origen se encuentra en un equilibrio estable.

Estabilidad asintótica

El $x = 0 \in \mathbb{R}^n$ es un equilibrio asintóticamente estable de (2.8) si:

1. El origen es estable
2. El origen es atractivo, es decir, $\delta' > 0$ tal que:

$$\|x(0)\| < \delta' \Rightarrow \|x(t)\| \rightarrow 0 \quad \text{cuando } t \rightarrow \infty$$

Estabilidad asintótica global

El origen $x = 0 \in \mathbb{R}^n$ es un equilibrio asintóticamente estable en forma global de 2.8 si:

1. El origen es estable
2. El origen es atractivo globalmente, es decir,

$$\|x(t)\| \rightarrow 0 \quad \text{cuando } t \rightarrow \infty, \forall x(0) \in \mathbb{R}^n$$

2.3.3. Método directo de Lyapunov

El origen $x = 0 \in \mathbb{R}^n$ es un estado de equilibrio estable si existe una función candidata a Lyapunov $V(t, x)$, tal que su derivada temporal satisfaga:

$$\dot{V}(t, x) \leq 0, \quad \forall t \geq 0$$

Estabilidad y acotamiento de las soluciones

El origen $x = 0 \in \mathbb{R}^n$ es un equilibrio estable, y las soluciones $x(t)$ están acotadas para toda condición inicial $x(0) \in \mathbb{R}^n$, si existe una función candidata de Lyapunov que sea definida positiva (globalmente), radialmente desacotada, $V(t, x)$, tal que su derivada temporal satisfaga:

$$\dot{V}(t, x) \leq 0 \forall t \geq 0 \forall x \in \mathbb{R}^n$$

Estabilidad asintótica global

El origen $x = 0 \in \mathbb{R}^n$ es un equilibrio asintóticamente estable, en forma global; si existe una función candidata de Lyapunov $V(t, x)$ definida positiva globalmente, radialmente desacotada, y menguante, tal que su derivada satisfaga:

1. $\dot{V}(t, 0) = 0 \forall t \geq 0$
2. $\dot{V}(t, x) < 0 \forall t \geq 0, \forall x \neq 0 \in \mathbb{R}^n$

El método directo establece que, si se construye una función de energía en los estados del sistema como función definida positiva continua diferenciable, la variación temporal debida a pequeñas perturbaciones en una vecindad del punto de equilibrio es siempre negativa o cero, entonces el punto de equilibrio es estable en forma global, es decir, para toda condición inicial que se encuentre dentro de la región de atracción. Sin embargo, si la razón de cambio de la función definida positiva es estrictamente negativa, entonces el punto de equilibrio es asintóticamente estable en forma global. Nótese que la razón de cambio de la función definida positiva corresponde a la potencia del sistema.

Sea el sistema lineal $\dot{x} = Ax$, y sea la función positiva definida $V(X) = x^T P x$, $P > 0$. Al calcular la derivada de $\dot{V}(x)$ se obtiene:

$$\dot{V}(x) = \frac{d}{dt}(x^T P x) = \frac{d}{dt}(x^T) P x + x^T P \frac{d}{dt}(x) = \dot{x}^T P x + x^T P \dot{x} \quad (2.9)$$

$$\dot{V}(x) = x^T (A^T P + P A) x \quad (2.10)$$

Definiendo

$$(A^T P + P A) = -Q \quad (2.11)$$

Notando que:

$$Q^T = (A^T P + P A)^T = P A + A^T P = A^T P + P A = -Q \therefore Q = Q^T \quad (2.12)$$

y $\dot{V}(x)$ es una forma cuadrática dependiente de Q

$$\dot{V}(x) = -x^T Q x \quad (2.13)$$

entonces, sí $Q > 0 \rightarrow \dot{V}(x) < 0$

sí, $V(x) > 0$ y $\dot{V}(x) < 0$

$x = 0$ Es un punto de equilibrio asintóticamente estable en el sentido de Lyapunov.

2.4. Dinámica

La dinámica comprende el estudio de las fuerzas y momentos que causan el movimiento de un sistema. En robótica, se refiere al movimiento y las fuerzas que se ejercen sobre el robot, ya sea en los actuadores o fuerzas externas que se ejerzan sobre el manipulador. Dependiendo de la configuración del manipulador, junto con el movimiento, se presentan fuerzas centrípetas debido a la rotación, así como también fuerzas de Coriolis, que se derivan por el movimiento relativo entre las articulaciones.

Conociendo los parámetros físicos del robot manipulador en cuestión: masa, longitud, longitud al centro de gravedad e inercia, todo esto para cada articulación presente en el robot. Con todos estos datos, es posible describir mediante una serie de ecuaciones el comportamiento dinámico del robot.

El modelo dinámico es necesario e indispensable para desarrollo del robot, ayudará a cumplir con los siguientes objetivos:

- Diseño del control dinámico del robot.
- Simulación del movimiento del robot.
- Elección de los motores.
- Diseño de la estructura mecánica del robot.

En la dinámica para robots manipuladores, hay dos problemas: dinámica inversa y dinámica directa. La dinámica inversa busca determinar el torque necesario para que los actuadores puedan seguir la trayectoria establecida, mientras que la dinámica directa busca determinar la trayectoria (aceleración, velocidad y posición) dado un torque establecido. Para el desarrollo de esta tesis, se utiliza la dinámica inversa, ya que es beneficioso también para controlar el robot.

2.4.1. Euler-Lagrange

Formulación de Euler Lagrange es un método basado en energía (cinética y potencial) donde el Lagrangiano es la diferencia entre la energía cinética ($K(q, \dot{q})$) y la energía potencial ($U(q)$).

$$\mathcal{L} = K(q, \dot{q}) - U(q) \quad (2.14)$$

Energía cinética se determina como la suma del producto de la mitad de su masa por el cuadrado de la rapidez del centro de masa o lo que es lo mismo, el producto de la mitad de su momento de inercia (referido al centro de masa) por la velocidad angular al cuadrado (referido al centro de masa).

$$K(q, \dot{q}) = \frac{1}{2}mv^T v + \frac{1}{2}I\dot{q}^2 \quad (2.15)$$

A comparación de la energía cinética, la energía potencial $U(q)$ no tiene una forma específica, depende de la geometría del robot, para lo que en forma general:

$$U(q) = mglh(q) \quad (2.16)$$

Siendo $h(q)$ una función que indica la altura del eslabón con respecto al origen del sistema de referencia del robot.

Las ecuaciones de movimiento de Euler Lagrange de un robot manipulador de n grados de libertad están dadas por:

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \right] - \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} = \tau - v(\dot{q}, f_e) \quad (2.17)$$

El modelo dinámico general de un robot manipulador de n grados de libertad, en su forma compacta y con la notación más ampliamente utilizada en el área de robótica, se expresa:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + f_f(\dot{q}, f_e) \quad (2.18)$$

Donde:

$M(q)$ es la matriz de inercia(masas)

$C(q, \dot{q})$ es la matriz de fuerzas centrípetas y de coriolis

$g(q)$ es el vector de fuerzas gravitacionales

$f_f(\dot{q}, f_e)$ es el vector de fricción (viscosa, Coulomb y estática)

Para el cálculo de la energía cinética y potencial, se requiere primero determinar las coordenadas generalizadas. Las coordenadas generalizadas determinan completamente la configuración (posición y orientación) de las articulaciones del sistema. Cuando conocemos las posiciones y orientaciones de todos los objetos en el sistema, decimos que conocemos completamente la configuración de un sistema mecánico. Por ejemplo, un cuerpo rígido tiene 6 grados de libertad, por lo que un sistema mecánico con m objetos en movimiento necesitaría $6m$ de coordenadas para definir completamente su configuración en el espacio cartesiano o articular. En el caso de que se trabaje con espacio articular, esto es, obtener la cinemática directa del centro de masa de cada una de las articulaciones.

Después se obtendrá la cinemática diferencial de cada articulación y subsecuentemente, deducir la rapidez lineal. Teniendo en cuenta que $v_i^T v_i = \dot{x}_i^2 + \dot{y}_i^2 + \dot{z}_i^2$

La metodología para obtener el modelo matemático de la dinámica de robots manipuladores lo simplificaremos a los siguientes pasos:

1. Cálculo de la energía cinética: $K(q, \dot{q})$.
2. Cálculo de la energía potencial: $U(q(t))$.
3. Cálculo del lagrangiano. Ecuación (2.14).
4. Desarrollo de las ecuaciones de Lagrange. Ecuación: (2.17).

2.4.2. Propiedades del modelo dinámico

El modelo dinámico del robot manipulador de n grados de libertad es una ecuación diferencial compleja, de naturaleza continua, multivariable, con dinámica fuertemente acoplada y no lineal en el vector de estados $[q^T \dot{q}^T]^T$. No obstante, tiene varias propiedades fundamentales que pueden ser explotadas para facilitar el diseño y análisis de sistemas de control.

Matriz de inercia

Representado por $M(q)\ddot{q}$

Propiedad 1. La matriz de inercia $M(q) \in \mathbb{R}^{n \times n}$ es una matriz simétrica $M(q) = M(q)^T$, definida positiva $M(q) > 0$. Por lo tanto, existe la matriz inversa $M(q)^{-1} \in \mathbb{R}^{n \times n}$ y también satisface que es una matriz simétrica $M(q)^{-1} = M(q)^{-T}$ y definida positiva $M(q)^{-1} > 0$.

Propiedad 2. La matriz de inercia $M(q)$ satisface que $\lambda_{M(q)}^{\min} I \leq M(q) \leq \lambda_{M(q)}^{\max} I$.

Propiedad 3. El fenómeno inercial de un robot manipulador satisface que:

$$\|M(q)\ddot{q}\| \leq \lambda_{M(q)}^{\max} \|\ddot{q}\| \forall q \in \mathbb{R}^n$$

; donde $\lambda_{M(q)}^{\max}$ representa el valor propio máximo de la matriz de inercia.

Propiedad 4. β del robot. Para el caso de robots provistos únicamente de articulaciones rotacionales, existe una constante $\beta > 0$ tal que:

$$\lambda_{M(q)}^{\max} \leq \beta \forall q \in \mathbb{R}^n$$

.

Propiedad 5. Para el caso de robots provistos únicamente de articulaciones rotacionales, existe una constante k_M tal que:

$$\|M(x)z - M(y)z\| \leq k_M \|x - y\| \|z\|$$

para todo vector $x, y, z \in \mathbb{R}^n$.

Matriz de fuerzas centrípetas y de Coriolis

Representado por $C(\dot{q}, q)\dot{q}$, la fuerza de coriolis representa una desviación del movimiento de traslación debido a su componente de rotación.

Propiedad 1. La matriz de fuerzas centrípetas y de coriolis $C(\dot{q}, q) \in \mathbb{R}^{n \times n}$ no es una matriz única, pero el vector $C(q, \dot{q})\dot{q}$ si lo es.

Propiedad 2. $C(q, 0) = 0$ para todo vector $q \in \mathbb{R}^n$.

Propiedad 3. Para todo vector $q, x, y, z \in \mathbb{R}^n$ y escalar α , se tiene que:

$$\begin{aligned} C(q, x)y &= C(q, x)z, \\ C(q, z + \alpha x)y &= C(q, z)y + \alpha C(q, x)y. \end{aligned}$$

Propiedad 4. El vector $C(q, x)y$ puede expresarse en la forma:

$$C(q, x)y = \begin{bmatrix} x^T C_1(q)y \\ x^T C_2(q)y \\ \vdots \\ x^T C_n(q)y \end{bmatrix}$$

donde $C_k(q)$ son matrices simétricas de dimensión n para todo $k = 1, 2, 3, \dots, n$.

Propiedad 5. Para el caso de robots provistos únicamente de articulaciones rotacionales, existe una constante $K_{c1} > 0$ tal que:

$$\|C(q, x)y\| \leq k_{c1} \|x\| \|y\|$$

para todo $q, x, y \in \mathbb{R}^n$

Propiedad 6. Para el caso de robots provistos únicamente de articulaciones rotacionales, existen constantes $k_{c1} > 0$ y $k_{c2} > 0$ tales que:

$$\|C(x, z)w - C(y, v)w\| \leq k_{c1} \|z - v\| \|w\| + k_{c2} \|z - y\| \|w\| \|z\|$$

para todo $q, x, y \in \mathbb{R}^n$

Vector gravitacional

El vector de pares gravitacionales $g(q)$ de $n \times 1$ depende solo de las posiciones articulares q . El vector $g(q)$ está acotado si q también lo está.

Propiedad 1. El vector $g(q)$ y el vector de velocidad \dot{q} pueden relacionarse mediante:

$$\int_a^b g(q)^T \dot{q} dt = \mathcal{U}(q(T)) - \mathcal{U}(q(0))$$

para todo $T \in \mathbb{R}_+$.

Propiedad 2. Para el caso de robots provistos únicamente de articulaciones rotacionales, existe una constante finita k_u tal que:

$$\int_0^T g(q)^T \dot{q} dt + \mathcal{U}(q(0)) \leq k_u$$

para todo $T \in \mathbb{R}_+$ y donde $k_u = \min_q \mathcal{U}(q)$.

Propiedad 3. Para el caso de robots provistos únicamente de articulaciones rotacionales, el vector $g(q)$ es Lipschitz, i.e., existe una constante $k_g > 0$ tal que:

$$\|g(x) - g(y)\| \leq k_g \|x - y\|$$

para todo $x, y \in \mathbb{R}^n$.

Propiedad 4. Para el caso de robots suministrados exclusivamente con articulaciones de tipo rotacionales, existen una constante k' tal que:

$$\|g(q)\| \leq k'$$

para todo $q \in \mathbb{R}^n$.

2.5. Robot de dos grados de libertad

A continuación se presenta el desarrollo de la cinemática directa e inversa, así como las ecuaciones dinámicas para un robot manipulador de dos grados de libertad con dos articulaciones rotacionales, posteriormente se desarrolla a manera de ejemplo un control de posición y un control de seguimiento de trayectoria.

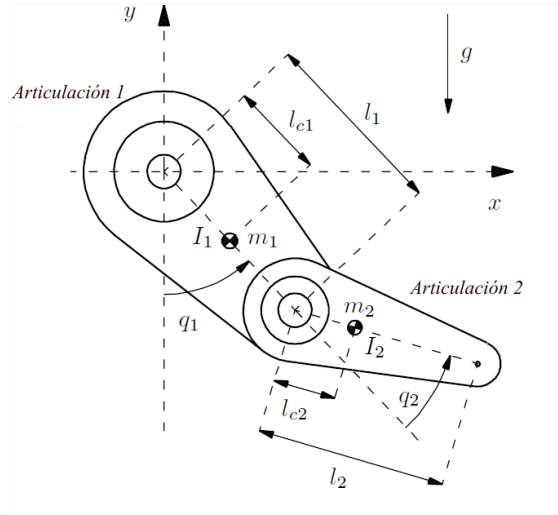


Figura 2.1: Diagrama robot de dos grados de libertad con dos articulaciones rotacionales [36]

2.5.1. Modelo cinemático directo

Cinemática directa

Debe de tomarse en cuenta que las ecuaciones presentadas a continuación son para un robot de dos grados de libertad cuya posición inicial es sobre el eje y negativo, tal y como se muestra en la figura 2.1.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_1 \text{sen}(q_1) + l_2 \text{sen}(q_1 + q_2) \\ -l_1 \text{cos}(q_1) - l_2 \text{cos}(q_1 + q_2) \end{bmatrix} \quad (2.19)$$

Cinemática diferencial

Para obtener la velocidad de la cinemática directa, se procede a obtener las derivadas, primero con respecto a q_1 y sucesivamente con respecto a q_2 , quedando de la siguiente manera:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} l_1 \text{cos}(q_1) + l_2 \text{cos}(q_1 + q_2) & l_2 \text{cos}(q_1 + q_2) \\ -l_1 \text{sen}(q_1) - l_2 \text{sen}(q_1 + q_2) & -l_2 \text{sen}(q_1 + q_2) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (2.20)$$

Jacobiano

De la ecuación 2.20 se puede obtener el jacobiano:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = J(q) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (2.21)$$

$$\begin{aligned}
J(q) &= \frac{\partial f(q)}{\partial q} \in \mathbb{R}^{2 \times 2} \\
&= \begin{bmatrix} l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \\ -l_1 \sin(q_1) - l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \end{bmatrix}
\end{aligned} \tag{2.22}$$

2.5.2. Modelo cinemático inverso

Cinemática inversa con respecto al cuarto cuadrante

$$q_2 = a \cos \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2} \tag{2.23}$$

$$q_1 = \frac{\pi}{2} + \operatorname{atan} \frac{y}{x} - \operatorname{atan} \frac{l_2 \sin(q_2)}{l_1 + l_2 \cos(q_2)} \tag{2.24}$$

Cinemática diferencial inversa con respecto al cuarto cuadrante

$$J^{-1}(q_d) = \begin{bmatrix} \frac{\sin(q_1+q_2)}{l_1 \sin q_2} & -\frac{\cos(q_1+q_2)}{l_1 \sin(q_2)} \\ \frac{-l_1 \sin(q_1) - l_2 \sin(q_1+q_2)}{l_1 l_2 \sin(q_2)} & \frac{l_1 \cos(q_1) + l_2 \cos(q_1+q_2)}{l_1 l_2 \sin(q_2)} \end{bmatrix} \tag{2.25}$$

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = J^{-1}(q) \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{2.26}$$

Aceleración inversa

$$\frac{d}{dt} [J(q_d)] = \begin{bmatrix} -l_1 \sin(q_1) \dot{q}_1 - l_2 \sin(q_1 + q_2) ((\dot{q}_1 + \dot{q}_2)) & -l_2 \sin(q_1 + q_2) (\dot{q}_1 + \dot{q}_2) \\ l_1 \cos(q_1) \dot{q}_1 + l_2 \cos(q_1 + q_2) ((\dot{q}_1 + \dot{q}_2)) & l_2 \cos(q_1 + q_2) (\dot{q}_1 + \dot{q}_2) \end{bmatrix} \tag{2.27}$$

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = -J^{-1}(q) \left[\frac{d}{dt} J(q) \right] J^{-1}(q) \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} + J^{-1}(q) \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} \tag{2.28}$$

2.5.3. Dinámica

Las coordenadas generalizadas al centro de masa de la primera articulación y de la segunda, correspondientes a la Figura 2.1, están dadas por:

$$\begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} l_{c1} \text{sen}(q_1) \\ -l_{c1} \text{cos}(q_1) \\ l_1 \text{sen}(q_1) + l_{c2} \text{sen}(q_1 + q_2) \\ -l_1 \text{cos}(q_1) - l_{c2} \text{cos}(q_1 + q_2) \end{bmatrix} \quad (2.29)$$

Por lo que sus respectivas derivadas o velocidades serían:

$$v_1 = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} l_{c1} \text{cos}(q_1) \dot{q}_1 \\ l_{c1} \text{sen}(q_1) \dot{q}_1 \end{bmatrix} \quad (2.30)$$

$$v_2 = \begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} l_1 \text{cos}(q_1) \dot{q}_1 + l_{c2} \text{cos}(q_1 + q_2) (\dot{q}_1 + \dot{q}_2) \\ l_1 \text{sen}(q_1) \dot{q}_1 + l_{c2} \text{sen}(q_1 + q_2) (\dot{q}_1 + \dot{q}_2) \end{bmatrix} \quad (2.31)$$

Para resolver la ecuación correspondiente a la energía cinética (2.15) se tiene que obtener $v^T v$.

$$v_1^T v_1 = l_{c1}^2 \dot{q}_1^2 \quad (2.32)$$

$$v_2^T v_2 = l_1^2 \dot{q}_1^2 + l_{c2}^2 [\dot{q}_1^2 + 2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2] + 2l_1 l_{c2} [\dot{q}_1^2 + \dot{q}_1 \dot{q}_2] \text{cos}(q_2) \quad (2.33)$$

La energía cinética para el primer eslabón sería:

$$k_1(q, \dot{q}) = \frac{1}{2} m_1 v_1^T v_1 + \frac{1}{2} I_1 \dot{q}_1^2 = \frac{1}{2} m_1 l_{c1}^2 \dot{q}_1^2 + \frac{1}{2} I_1 \dot{q}_1^2 \quad (2.34)$$

La energía cinética para el segundo eslabón sería:

$$\begin{aligned} k_2(q, \dot{q}) &= \frac{1}{2} m_2 v_2^T v_2 + \frac{1}{2} I_2 [\dot{q}_1 + \dot{q}_2]^2 \\ &= \frac{1}{2} m_2 l_1^2 \dot{q}_1^2 + \frac{1}{2} m_2 l_{c2}^2 [\dot{q}_1^2 + 2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2] + \\ &\quad m_2 l_1 l_{c2} [\dot{q}_1^2 + \dot{q}_1 \dot{q}_2] \text{cos}(q_2) + \frac{1}{2} I_2 [\dot{q}_1 + \dot{q}_2]^2 \end{aligned} \quad (2.35)$$

La energía potencial dada la ecuación 2.16 para los dos eslabones es:

$$U_1(q) = -m_1 l_{c1} g \text{cos}(q_1) \quad (2.36)$$

$$U_2(q) = -m_2 l_1 g \text{cos}(q_1) - m_2 l_{c2} g \text{cos}(q_1 + q_2) \quad (2.37)$$

A partir de las ecuaciones anteriores se generan el lagrangiano (2.14).

$$\begin{aligned} \mathcal{L} &= \frac{1}{2} m_1 l_{c1}^2 \dot{q}_1^2 + \frac{1}{2} I_1 \dot{q}_1^2 + \frac{1}{2} m_2 v_2^T v_2 + \\ &\quad \frac{1}{2} I_2 [\dot{q}_1 + \dot{q}_2]^2 - m_1 l_{c1} g \text{cos}(q_1) - \\ &\quad m_2 l_1 g \text{cos}(q_1) - m_2 l_{c2} g \text{cos}(q_1 + q_2) \end{aligned} \quad (2.38)$$

Y dado el lagrangiano, se puede formular las ecuaciones de movimiento de Lagrange, ecuación (2.17). Para el cual necesitamos de las siguientes expresiones:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \dot{q}_1} &= [m_1 l_{c1}^2 + m_2 l_1^2] \dot{q}_1 + m_2 l_{c2}^2 \dot{q}_2 + 2m_2 l_1 l_{c2} \cos(q_2) \dot{q}_1 + \\ &\quad m_2 l_1 l_{c2} \cos(q_2) \dot{q}_2 + I_1 \dot{q}_1 + I_2 [\dot{q}_1 + \dot{q}_2] \\ \frac{\partial \mathcal{L}}{\partial q_1} &= -[m_1 l_{c1} + m_2 l_1] g \text{sen}(q_1) - m_2 g l_{c2} \text{sen}(q_1 + q_2)\end{aligned}$$

$$\begin{aligned}\frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{q}_1} \right] &= [m_1 l_{c1}^2 + m_2 l_1^2 + 2m_2 l_1 l_{c2} \cos(q_2)] \ddot{q}_1 + [m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2)] \ddot{q}_2 - \\ &\quad 2m_2 l_1 l_{c2} \text{sen}(q_2) \dot{q}_1 \dot{q}_2 - m_2 l_1 l_{c2} \text{sen}(q_2) \dot{q}_2^2 + I_1 \ddot{q}_1 + I_2 [\ddot{q}_1 + \ddot{q}_2] \\ \frac{\partial L}{\partial \dot{q}_2} &= m_2 l_{c2}^2 \dot{q}_1 + m_2 l_{c2}^2 \dot{q}_2 + m_2 l_1 l_{c2} \cos(q_2) \dot{q}_1 + I_2 [\dot{q}_1 + \dot{q}_2]\end{aligned}$$

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}_2} \right] = m_2 l_{c2}^2 \ddot{q}_1 + m_2 l_{c2}^2 \ddot{q}_2 + m_2 l_1 l_{c2} \cos(q_2) \ddot{q}_1 - m_2 l_1 l_{c2} \text{sen}(q_2) \dot{q}_1 \dot{q}_2 + I_2 [\ddot{q}_1 + \ddot{q}_2]$$

$$\frac{\partial L}{\partial q_2} = -m_2 l_1 l_{c2} \text{sen}(q_2) [\dot{q}_1 \dot{q}_2] - m_2 g l_{c2} \text{sen}(q_1 + q_2)$$

Aplicando la ecuación (2.17), se obtienen los pares que deberían de proporcionar los motores en cada articulación, usando las expresiones antes presentadas:

$$\begin{aligned}\tau_1 &= [m_1 l_{c1}^2 + m_2 l_{c2}^2 + 2m_2 l_1 l_{c2} \cos(q_2) + I_1 + I_2] \ddot{q}_1 + \\ &\quad [m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2) + I_2] \ddot{q}_2 - \\ &\quad 2m_2 l_1 l_{c2} \text{sen}(q_2) \dot{q}_1 \dot{q}_2 - m_2 l_1 l_{c2} \text{sen}(q_2) \dot{q}_2^2 + \\ &\quad [m_1 l_{c1} + m_2 l_1] g \text{sen}(q_1) + m_2 g l_{c2} \text{sen}(q_1 + q_2)\end{aligned}\tag{2.39}$$

$$\begin{aligned}\tau_2 &= [m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2) + I_2] \ddot{q}_1 + \\ &\quad [m_2 l_{c2}^2 + I_2] \ddot{q}_2 + m_2 l_1 l_{c2} \text{sen}(q_2) \dot{q}_1^2 + \\ &\quad m_2 g l_{c2} \text{sen}(q_1 + q_2)\end{aligned}\tag{2.40}$$

A continuación se explicará brevemente otro método para calcular τ y los demás valores deseados [36].

$$k(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q}\tag{2.41}$$

Donde $M(q)$ es una matriz simétrica definida positiva de $n \times n$ denominada matriz de inercia. Como ya se mencionó anteriormente, la energía potencial $U(q)$ no tiene una forma definida como el caso de la energía cinética, pero se reconoce que esta relacionada al vector de posiciones

articulares q .

El lagrangiano $\mathcal{L}(q, \dot{q})$, para este caso, se representa de la siguiente manera:

$$\mathcal{L}(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} - U(q)$$

Siendo entonces la ecuación de movimiento de Lagrange, la siguiente:

$$\frac{d}{dt} \left[\frac{\partial}{\partial \dot{q}} \left[\frac{1}{2} \dot{q}^T M(q) \dot{q} \right] \right] - \frac{\partial}{\partial q} \left[\frac{1}{2} \dot{q}^T M(q) \dot{q} \right] + \frac{\partial U(q)}{\partial q} = \tau \quad (2.42)$$

Verificando que:

$$\begin{aligned} \frac{\partial}{\partial \dot{q}} \left[\frac{1}{2} \dot{q}^T M(q) \dot{q} \right] &= M(q) \dot{q} \\ \frac{d}{dt} \left[\frac{\partial}{\partial \dot{q}} \left[\frac{1}{2} \dot{q}^T M(q) \dot{q} \right] \right] &= M(q) \ddot{q} + \dot{M}(q) \dot{q} \end{aligned}$$

La ecuación de movimiento de Lagrange toma la siguiente estructura:

$$M(q) \ddot{q} + \dot{M}(q) \dot{q} - \frac{1}{2} \frac{\partial}{\partial q} [\dot{q}^T M(q) \dot{q}] + \frac{\partial U(q)}{\partial q} = \tau \quad (2.43)$$

Haciéndolo más compacto tendremos la ecuación (2.18).

Agrupando de forma matricial, los términos de aceleración (\ddot{q}_1, \ddot{q}_2) y velocidades (\dot{q}_1, \dot{q}_2) , tomando la siguiente forma:

$$\begin{aligned} M(q) &= \begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} \\ C(q) &= \begin{bmatrix} C_{11}(q) & C_{12}(q) \\ C_{21}(q) & C_{22}(q) \end{bmatrix} \\ g(q) &= \begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix} \\ \tau &= \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \end{aligned}$$

donde:

$$\begin{aligned}
M_{11}(q) &= m_1 l_{c1}^2 + m_2 l_1^2 + m_2 l_{c2}^2 + 2m_2 l_1 l_{c2} \cos(q_2) + I_1 + I_2 \\
M_{12}(q) &= m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2) + I_2 \\
M_{21}(q) &= m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2) + I_2 \\
M_{22}(q) &= m_2 l_{c2}^2 + I_2 \\
C_{11}(q, \dot{q}) &= -m_2 l_1 l_{c2} \operatorname{sen}(q_2) \dot{q}_2 \\
C_{12}(q, \dot{q}) &= -m_2 l_1 l_{c2} \operatorname{sen}(q_2) [\dot{q}_1 + \dot{q}_2] \\
C_{21}(q, \dot{q}) &= m_2 l_1 l_{c2} \operatorname{sen}(q_2) \dot{q}_1 \\
C_{22}(q, \dot{q}) &= 0 \\
g_1(q) &= [m_1 l_{c1} + m_2 l_1] g \operatorname{sen}(q_1) + m_2 g l_{c2} \operatorname{sen}(q_1 + q_2) \\
g_2(q) &= m_2 g l_{c2} \operatorname{sen}(q_1 + q_2)
\end{aligned}$$

Cabe resaltar que los elementos de las matrices $M(q)$, $C(q, \dot{q})$ y $g(q)$ depende de la geometría del robot que se está modelando.

La ecuación 2.18 puede escribirse en variables de estado de la siguiente manera:

$$\frac{d}{dt} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ M(q)^{-1} [\tau(t) - C(q, \dot{q}) \ddot{q} - g(q)] \end{bmatrix} \quad (2.44)$$

Capítulo 3 | Desarrollo del problema

3.1. Desarrollo

El problema de este trabajo se centra en el diseño de un controlador para seguimiento de trayectoria de la clase de sistemas lagrangianos, con un controlador con solo una parte proporcional y una parte de compensación para perturbaciones.

Considerando τ_u como la entrada de control:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F_v\dot{q} = \tau_u + \omega \quad (3.1)$$

donde $q \in \mathbb{R}^{n \times 1}$ es el vector de posiciones, $M(q)$ es la matriz de inercia, $C(q, \dot{q})$ es la matriz de fuerzas centrípetas y de coriolis, $G(q)$ es el vector de gravedad, F_v es la matriz de fricción viscosa, $\tau_u \in \mathbb{R}^n$ corresponde a la fuerza o torque, y $w \in \mathbb{R}^n$ corresponde a incertidumbres y perturbaciones.

Definiendo las variables de estado como $x_1 = q, x_2 = \dot{q}$, la representación en espacio de estados del sistema de (3.1) es

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= M^{-1}(x_1)[-C(x_1, x_2)x_2 - G(x_1) - F_v x_2 + \tau + \omega(x_1, x_2, t)] \\ y &= x_1 \end{aligned} \quad (3.2)$$

La única variable a medir será la posición x_1 .

3.1.1. Diseño del controlador

La principal característica del controlador que se presenta, es solo el uso de mediciones de posición para su retroalimentación, así como también se omitirá el uso de filtros y observadores. Tendrá también un algoritmo para perturbaciones externas constantes.

Considerando τ_u como la entrada de control dada por

$$\tau = G(x_1) + M(x_1)\ddot{x}_d + C(x_1, \dot{x}_d)\dot{x}_d + F_v\dot{x}_d - K_p z_1. \quad (3.3)$$

Donde $x_d = [x_{d1}, \dots, x_{dn}]^T \in \mathbb{R}^n$ es un vector de trayectorias deseadas, se asume que puede ser diferenciable dos veces; los errores de posición y velocidad son dados de la siguiente forma: $z_1 = x_1 - x_d$ y $z_2 = x_2 - \dot{x}_d$.

$K_p \in \mathbb{R}^{n \times n}$ es una matriz diagonal de ganancias y es definida positiva.

El sistema en lazo cerrado se obtiene reemplazando τ de (3.3) en (3.2) y considerando $-C(x_1, \dot{x}_d)\dot{x}_d + C(x_1, \dot{x})\dot{x}_d = 0$, y $w(x_1, x_2, t) = 0$.

Por lo tanto, para el caso sin perturbación:

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= -M^{-1}(x_1)[k_p Z_1 + [C(x_1, x_2) + C(x_1, \dot{x}_d)]z_2 + F_v z_2] \end{aligned} \quad (3.4)$$

El único punto de equilibrio del sistema en lazo cerrado (3.4) está dado por:

$$(\bar{z}_1, \bar{z}_2) = (0, 0) \quad (3.5)$$

3.1.2. Análisis de estabilidad

Para propósitos de estabilidad se propone la siguiente función candidata de Lyapunov:

$$V(t, z_1, z_2) = \frac{1}{2} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}^T \begin{bmatrix} K_p & 0 \\ 0 & M(z_1 + x_d) \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} > 0 \quad (3.6)$$

La derivada de (3.6) a lo largo de las soluciones de (3.4) está dado por:

$$\dot{V}(t, z_1, z_2) = -z_2^T [-C(x_1, \dot{x}_d) + F_v] z_2. \quad (3.7)$$

La propiedad $\frac{1}{2} z_2^T [M - 2C] z_2 = 0$ ha sido usada para todo $z_2 \in \mathbb{R}^n$, mostrado en la sección 4.2 de [36].

A continuación, se calcula para el límite superior $\dot{V}(t, z_1, z_2)$ por una función definida negativa en términos de sus estados z_1 y z_2 , para el cual es conveniente encontrar el límite superior de (3.7) de la siguiente manera:

$$\begin{aligned} -z_2^T [C(x_1, \dot{x}_d) + F_v] z_2 &\leq \\ (k_{c1} \|\dot{x}_d\| - \lambda_{\min} F_v) \|z_2\|^2 \end{aligned} \quad (3.8)$$

El límite (3.8) asegura que la derivada $\dot{V}(z_1, z_2)$ en (3.7) satisface el límite superior dado por

$$\dot{V}(t, z_1, z_2) \leq - \begin{bmatrix} \|z_1\| \\ \|z_2\| \end{bmatrix}^T R \begin{bmatrix} \|z_1\| \\ \|z_2\| \end{bmatrix} \quad (3.9)$$

donde

$$R = \begin{bmatrix} 0 & 0 \\ 0 & \lambda_{\min} F_v - k_{c1} \|\dot{x}_d\| \end{bmatrix} \quad (3.10)$$

La matriz R se puede mantener como positiva semidefinida cuando $\lambda_{\min} F_v > k_{c1} \|\dot{x}_d\|$. En este punto, para el control de posición se puede aplicar el teorema de LaSalle. Sin embargo, para este caso, el error del modelo dinámico está gobernado por una ecuación diferencial no autónoma. Por lo que en lugar de aplicar el teorema de LaSalle, se hará uso del teorema de Matrosov para probar estabilidad asintótica global en el origen.

3.1.3. Estimador de perturbaciones

Una compensación para la fricción viscosa adaptativa a la ley de control es de la siguiente manera:

$$\tau_u = G(x_1) + M(x_1)\ddot{x}_d + C(x_1, \dot{x}_d)\dot{x}_d + F_v\dot{x}_d - K_p z_1 - \hat{\omega}(x_1, x_2, t). \quad (3.11)$$

donde $\hat{\omega}(x_1, x_2, t) \in \mathbb{R}^n$ es un vector de estimador de perturbaciones.

La dinámica del error en lazo cerrado utilizando (3.2) y (3.11) queda de la siguiente manera

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= -M^{-1}(x_1)[k_p z_1 + [C(x_1, x_2) + C(x_1, \dot{x}_d)]z_2 + F_v z_2 - \tilde{\omega}] \end{aligned} \quad (3.12)$$

Se considera $\tilde{\omega}(\cdot) = \omega(\cdot) - \hat{\omega}(\cdot)$. Asumiendo que $\omega(\cdot)$ es un vector constante, entonces

$$\frac{d}{dt}\hat{\omega}(\cdot) = -\frac{d}{dt}\tilde{\omega}(\cdot). \quad (3.13)$$

La estabilidad y convergencia del estimador adaptativo puede ser analizado mediante la teoría de Lyapunov usando:

$$V_a(t, z_1, z_2, z_3, \hat{\omega}) = V(t, z_1, z_2, z_3) + \frac{1}{2}\tilde{\omega}^T \gamma^{-1} \hat{\omega}. \quad (3.14)$$

Tomando $V(t, z_1, z_2, z_3)$ de (3.6) pero ahora considerando el sistema en lazo cerrado (3.12) y $\gamma \in \mathbb{R}^{2 \times 2}$ como una matriz diagonal, tenemos que

$$\dot{V}_a(t, z_1, z_2, z_3, \hat{\omega}) = \dot{V}(t, z_1, z_2, z_3) + z_2^T \tilde{\omega} + \tilde{\omega}^T \gamma^{-1} \dot{\hat{\omega}}. \quad (3.15)$$

Donde la ley de control se adapta de la siguiente manera

$$\dot{\hat{\omega}} = -\gamma z_2. \quad (3.16)$$

Por lo tanto,

$$\dot{\hat{\omega}} = \gamma z_2. \quad (3.17)$$

Finalmente como $\dot{V}_a(t, z_1, z_2, z_3, \hat{\omega}) = \dot{V}(t, z_1, z_2, z_3) \leq 0$, la convergencia asintótica del sistema en espacio de estados (3.12) se comprueba mientras que $\hat{\omega}$ queda acotado para todo tiempo.

De (3.15), se ha agregado una ganancia L al error de posición

$$\dot{\hat{\omega}} = \gamma z_2 + L z_1 \quad (3.18)$$

donde $L \in \mathbb{R}^{2 \times 2}$ es una matriz diagonal.

Para evitar usar z_2 porque \dot{x} no puede ser medida, se integra $\dot{\hat{\omega}}$, quedando de este modo la siguiente expresión

$$\begin{aligned} \hat{\omega} &= \gamma z_1 + L \theta \\ \dot{\theta} &= z_1 \end{aligned} \quad (3.19)$$

Mediante el uso de (3.10) para estimar $\omega(x_1, x_2, t)$, el uso del vector de velocidad x_2 es omitido.

3.2. Metodología para simulación

3.2.1. Generación de trayectoria

Usualmente, el espacio de trabajo de los robots manipuladores suelen ser en espacio cartesiano o espacio articular, los cuales nos brindan diferentes enfoques espaciales de la posición del robot. Para el caso de espacio articular, se refiere a las coordenadas expresadas en grados o radianes de cada una de las articulaciones del robot, mientras que en espacio cartesiano se refiere a la posición en coordenadas cartesianas del efector final.

Para el problema de seguimiento de trayectoria, la trayectoria en cualquiera de los dos espacios de trabajo debe evolucionar con respecto al tiempo, en donde las posiciones en espacio articular serán $q \in \mathbb{R}^n$, mientras que en espacio cartesiano $[x, y]^T$. La cinemática directa consiste en expresar la posición del efecto final del robot manipulador con base en las coordenadas articulares. La cinemática inversa nos permitirá obtener las coordenadas articulares del robot manipulador en función de la posición del efector final.

La planificación de la trayectoria es un subconjunto del problema general que es la navegación o la planificación del movimiento. La jerarquía típica de planificación de movimiento es la siguiente:

1. Planificación de tareas. Establecer los objetivos que el robot deberá alcanzar.
2. Planificación de la ruta. Generar la ruta por las que deberán de pasar las articulaciones o el efector final.
3. Planificación de la trayectoria. Establecer la ruta con funciones dependientes del tiempo, es decir, determinar la velocidad y aceleración.
4. Seguimiento de trayectoria. Introducir la trayectoria deseada en el sistema de control que permitirá mover el robot.

Para la simulación se seguirán los siguientes pasos:

1. Definir trayectoria con ecuación paramétrica en espacio cartesiano.
2. Determinar la respectiva velocidad y aceleración.
3. Transformar las coordenadas cartesianas junto con sus respectivas derivadas a cinemática inversa
 - Usar la ecuación 2.23 para obtener las posiciones en espacio articular.
 - Usar la ecuación 2.26 para obtener la velocidad en espacio articular.
 - Usar la ecuación 2.28 para obtener las aceleraciones en espacio articular.
4. Una vez obtenidas las posiciones, velocidad y aceleraciones deseadas en espacio articular, estas podrán ser usadas dentro del sistema robótico.

A continuación, se presentarán trayectorias en los distintos espacios de trabajo, cabe resaltar que, a pesar de que son diferentes figuras, el procedimiento para la animación es diferente, en el caso de la trayectoria de espacio cartesiano se deben de transformar a espacio articular mediante la cinemática inversa, adelante se mostrará un ejemplo sencillo.

Ejemplo trayectoria espacio articular

	Articulación 1	Articulación 2
Posición	$q_1 = 0.3\sin(t)$	$q_2 = 0.3\cos(t)$
Velocidad	$\dot{q}_1 = 0.3\cos(t)$	$\dot{q}_2 = -0.3\sin(t)$
Aceleración	$\ddot{q}_1 = -0.3\sin(t)$	$\ddot{q}_2 = -0.3\cos(t)$

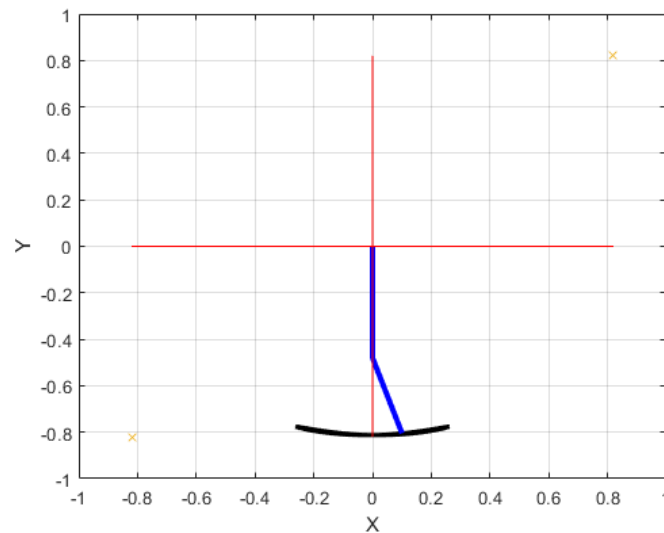


Figura 3.1: Trayectoria articular con función seno y coseno

Ejemplo trayectoria espacio cartesiano

	Articulación 1	Articulación 2
Posición	$x = 0.01(16\sin(t)^3) - 0.05$	$y = 0.01(13\cos(t) - 5\cos(2t) - 2\cos(3t) - \cos(4t)) - 0.4$
Velocidad	$\dot{x} = 0.48\sin(t)^2$	$\dot{y} = 0.01(-13\sin(t) + 10\sin(2t) + 6\sin(3t) + 4\sin(4t))$
Aceleración	$\ddot{x} = 0.01(288\cos(t)\sin(t)^2 - 3\cos(t)^3)$	$\ddot{y} = 0.01(-13\cos(t) + 20\cos(2t) + 18\cos(3t) + 16\cos(4t))$

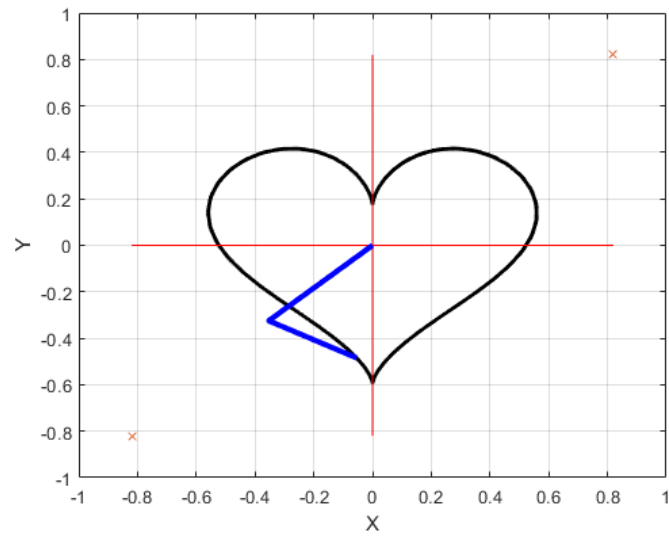


Figura 3.2: Trayectoria cartesiana: Figura en forma de corazón

3.2.2. Trayectorias cartesianas

Como se indicó en el apartado anterior, las trayectorias en espacio cartesiano son más intuitivas para el usuario. Hay varias trayectorias cartesianas que se han analizado, para ello, se parte de un tipo de ecuaciones con el que se pueda ver una evolución conforme va transcurriendo el tiempo, la cual sería una variable de la cual dependerán las otras variables con las que después podremos obtener un sistema de coordenadas que nos permitirán representar una curva en el plano.

Ecuaciones paramétricas

Si x y y están expresadas como funciones $x = f(t)$, $y = g(t)$ en un intervalo I de valores t , entonces, el conjunto de puntos $(x, y) = (f(t), g(t))$ definido por estas ecuaciones es una curva paramétrica, a las cuales llamaremos, ecuaciones paramétricas de la curva [37].

Algunas ecuaciones paramétricas famosas que se pueden aplicar al seguimiento de trayectoria y que se han estudiado en este trabajo de investigación son la hipocicloide, cardioide, epicicloide, nefroide, astroide, deltoide, curvas de Lissajous, entre otras, de las cuales se hablará brevemente a continuación.

Epitrocoide

Una epitrocoide es la curva que describe el recorrido de un punto a cierta distancia a del centro de una circunferencia que rota sin deslizarse o sin resbalar sobre otra circunferencia de referencia.

$$\begin{aligned}x &= (a + b)\cos(\theta) - h\cos\left(\frac{a + b}{b}\theta\right) \\y &= (a + b)\sin(\theta) - h\sin\left(\frac{a + b}{b}\theta\right)\end{aligned}$$

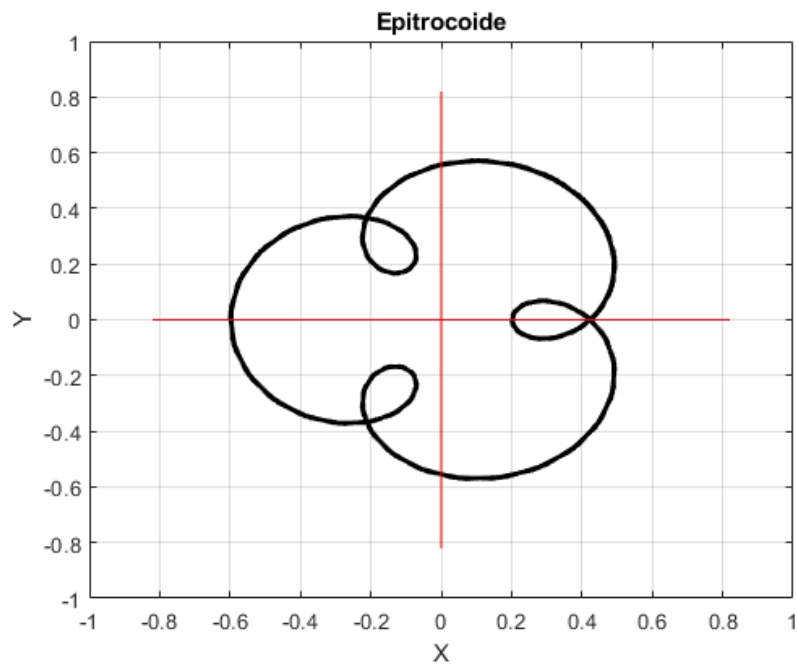


Figura 3.3: Epicicloide. $a=0.3$; $b=0.1$; $h=0.2$.

Hipotrocoide

Un hipotrocoide (figura: 3.4) es la curva plana que describe un punto vinculado a una circunferencia generatriz que rueda dentro de una circunferencia directriz, tangencialmente, sin deslizamiento.

$$x = (R - r)\cos\theta + d\cos\left(\frac{R - r}{r}\theta\right)$$

$$y = (R - r)\sin\theta - d\sin\left(\frac{R - r}{r}\theta\right)$$

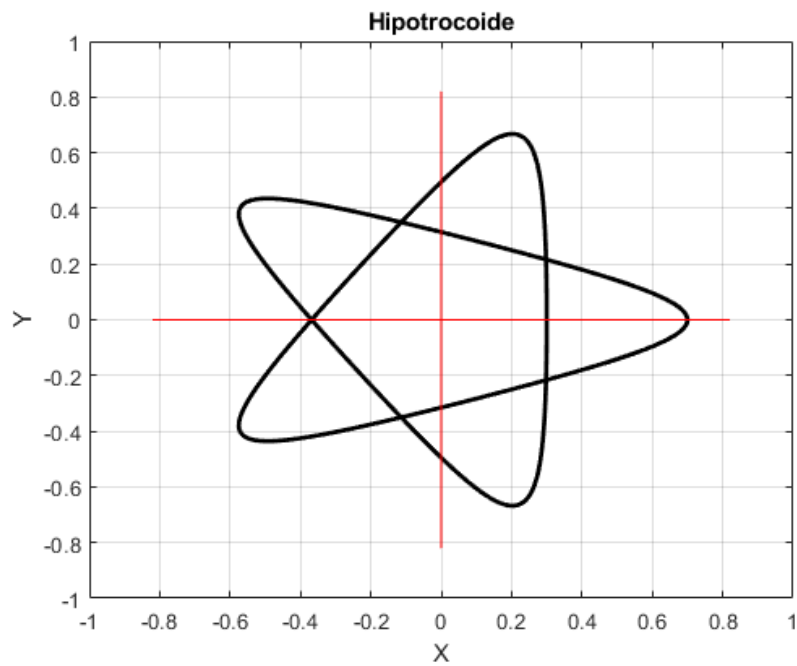


Figura 3.4: Hipotrocoide. $R=0.5$; $r=0.3$; $d=0.5$.

Epicicloide

Las epicicloides (figura: 3.5) ordinarias son curvas que se generan por un punto P de una circunferencia de radio b al rotar exteriormente y sin deslizamiento sobre otra circunferencia de radio a .

$$x = (r_1 + r_2)\sin\alpha - r_2\sin\left[\alpha\left(1 + \frac{r_1}{r_2}\right)\right]$$

$$y = (r_1 + r_2)\cos\alpha - r_2\cos\left[\alpha\left(1 + \frac{r_1}{r_2}\right)\right]$$

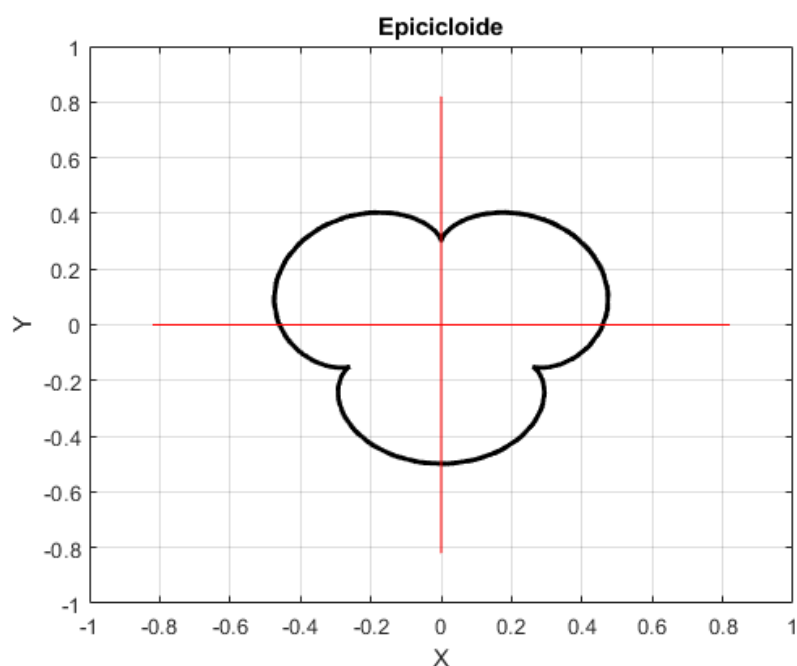


Figura 3.5: Epicicloide. $r_1=0.3$; $r_2=0.1$.

Algunos casos particulares se dan cuando

- $r_1 = r_2$, se obtiene una *cardioide*.
- $r_1 = 2r_2$, se obtiene una *nefroide*.

Nefroide

Nefroide (figura: 3.6), es un epicicloide en la que el radio de la ruleta es la mitad del de la directriz, pudiéndose ser normal, acortada o alargada, si el punto generador esta sobre, dentro o fuera de la ruleta. Su forma es parecida al contorno de un cacahuete, formada por dos arcos.

Esta curva fue estudiada por Huygens, Tschirnhausen en 1679, Jacques Bernoulli en 1692, Daniel Bernoulli en 1725 y Proctor en 1878; este último fue el que le dio el nombre de nefroide, proveniente del griego “nephros” que significa riñón.

$$x = (r_1 + r_2)\sin\alpha - r_2\sin\left[\alpha\left(1 + \frac{r_1}{r_2}\right)\right]$$

$$y = (r_1 + r_2)\cos\alpha - r_2\cos\left[\alpha\left(1 + \frac{r_1}{r_2}\right)\right]$$

Donde $r_1 = 2r_2$, por ejemplo $r_1 = 4$; $r_2 = 2$.

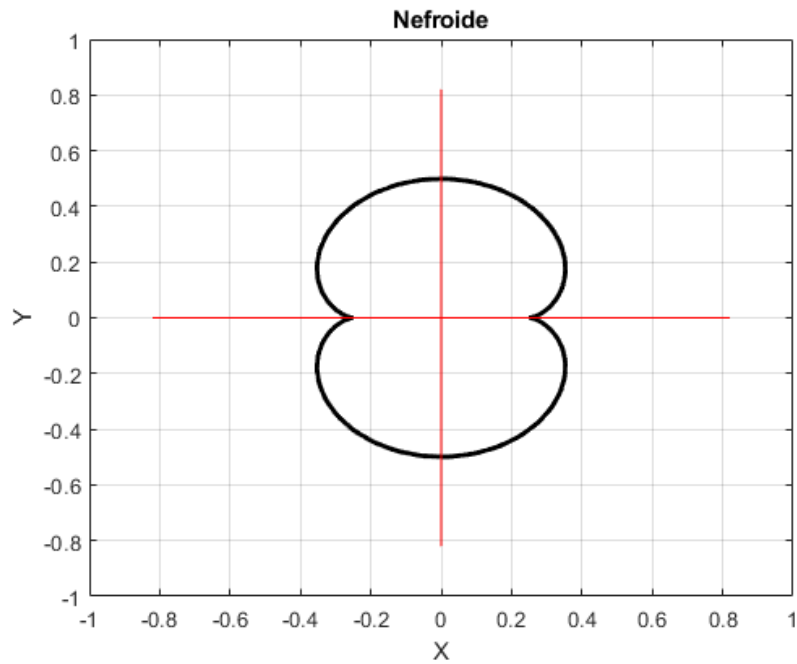


Figura 3.6: Nefroide. $r_1 = 0.25$; $r_2 = 0.125$.

Cardioide

Una cardioide (figura: 3.7) es una curva plana obtenida por una trayectoria descrita por un punto de una circunferencia que rueda sin deslizamiento tangencial por el exterior de otra circunferencia del mismo radio, donde está última está fija. Se llama cardioide porque puede tener la forma de un corazón.

$$x = (r_1 + r_2)\sin\alpha - r_2\sin\left[\alpha\left(1 + \frac{r_1}{r_2}\right)\right]$$

$$y = (r_1 + r_2)\cos\alpha - r_2\cos\left[\alpha\left(1 + \frac{r_1}{r_2}\right)\right]$$

Donde, $r_1 = r_2$.

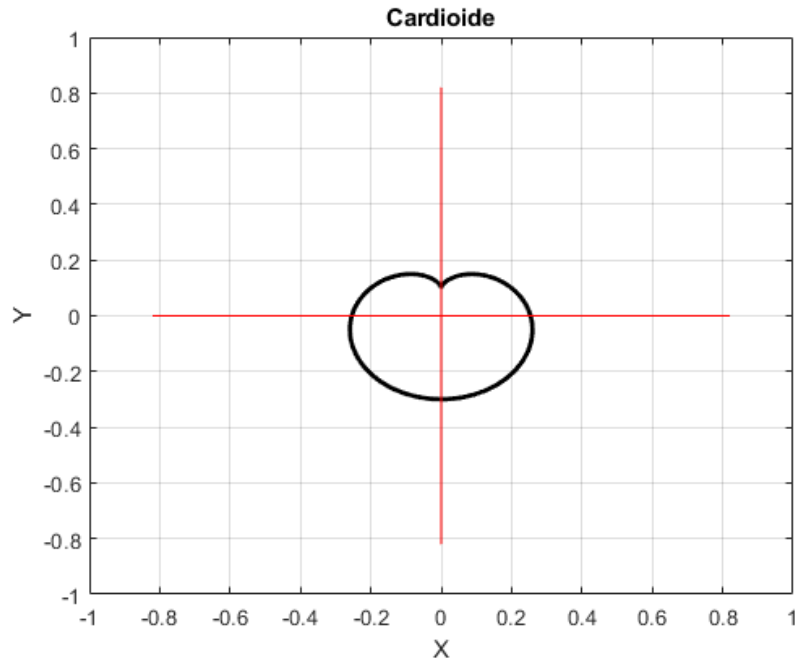


Figura 3.7: Cardioide. $r_1 = 0.1; r_2 = 0.1$

Hipocicloide: Deltoide

Las curvas hipocicloides ordinarias son curvas que se generan por un punto P de una circunferencia de radio b al rotar interiormente y sin deslizamiento sobre otra circunferencia de radio a .

$$x = (r_1 - r_2)\cos\alpha + r_2\cos\left[\alpha\left(1 - \frac{r_1}{r_2}\right)\right]$$

$$y = (r_1 - r_2)\sin\alpha + r_2\sin\left[\alpha\left(1 - \frac{r_1}{r_2}\right)\right]$$

Cuando $k = \frac{r_1}{r_2} = 3$, es decir, es una curva hipocicloide de 3 puntas, se le denomina deltoide, como la de la figura 3.8.

$$r_1 = 9; r_2 = 3; k = \frac{r_1}{r_2}$$

$$x = (r_1 - r_2)\cos\alpha + r_2\cos[\alpha(1 - k)]$$

$$y = (r_1 - r_2)\sin\alpha + r_2\sin[\alpha(1 - k)]$$

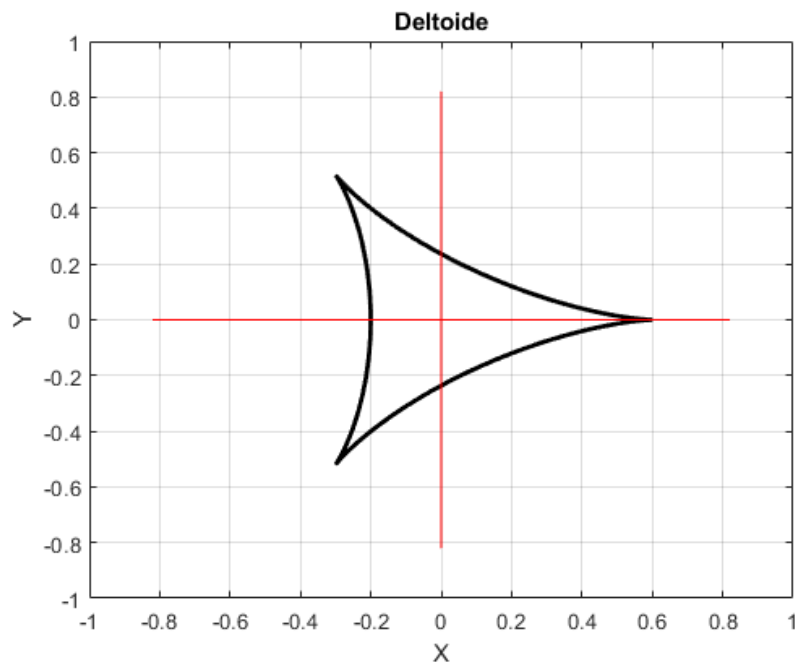


Figura 3.8: Deltoide. $r_1 = 0.6; r_2 = 0.2$

Hipocicloide: Astroide

Una curva hipocicloide de 4 puntas, se le denomina astroide (figura: 3.9). Y esta se obtiene cuando $r_1 = 4r_2$.

$$x = (r_1 - r_2)\cos\alpha + r_2\cos\left[\alpha\left(1 - \frac{r_1}{r_2}\right)\right]$$

$$y = (r_1 - r_2)\sin\alpha + r_2\sin\left[\alpha\left(1 - \frac{r_1}{r_2}\right)\right]$$

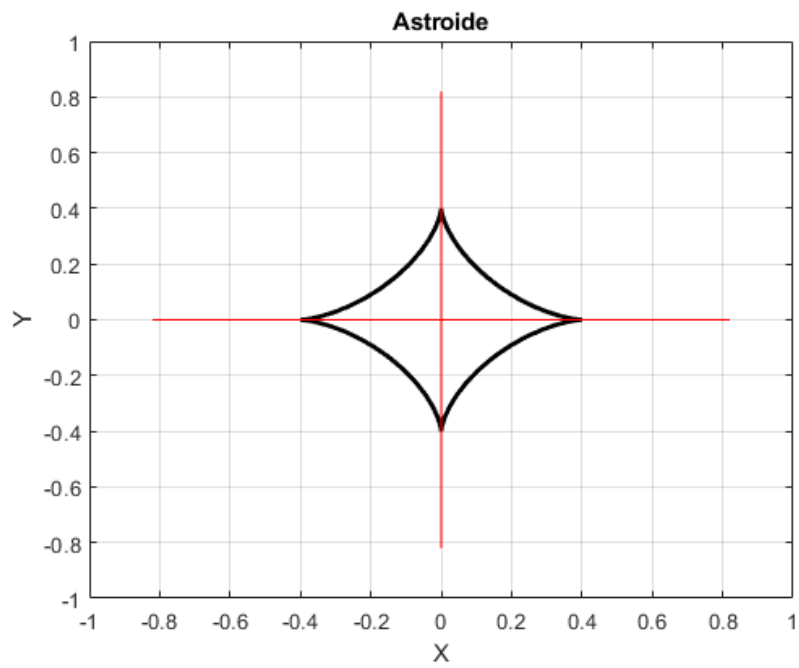


Figura 3.9: Astroide. $r_1 = 0.4; r_2 = 0.1$

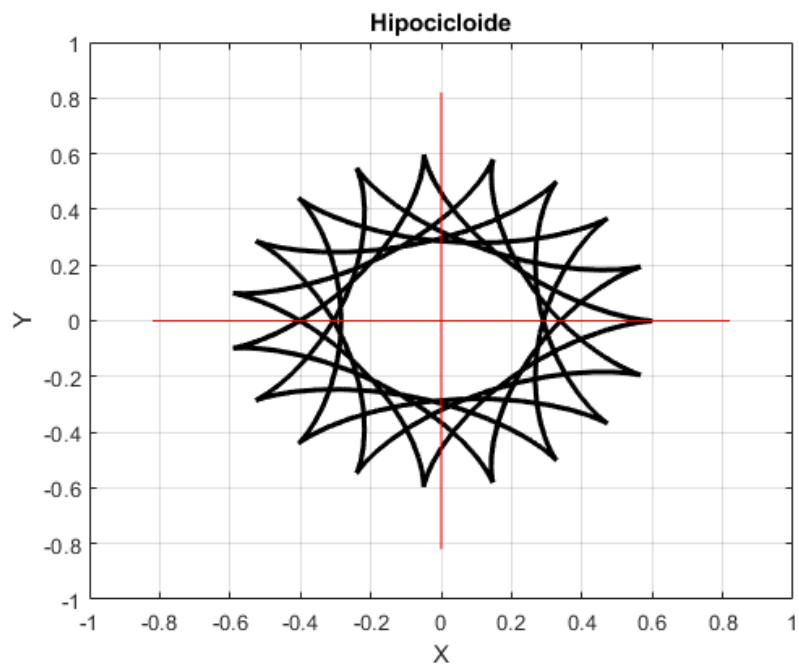


Figura 3.10: Hipocicloide, $k=3.8$. $r_1 = 0.6; r_2 = 0.15789$

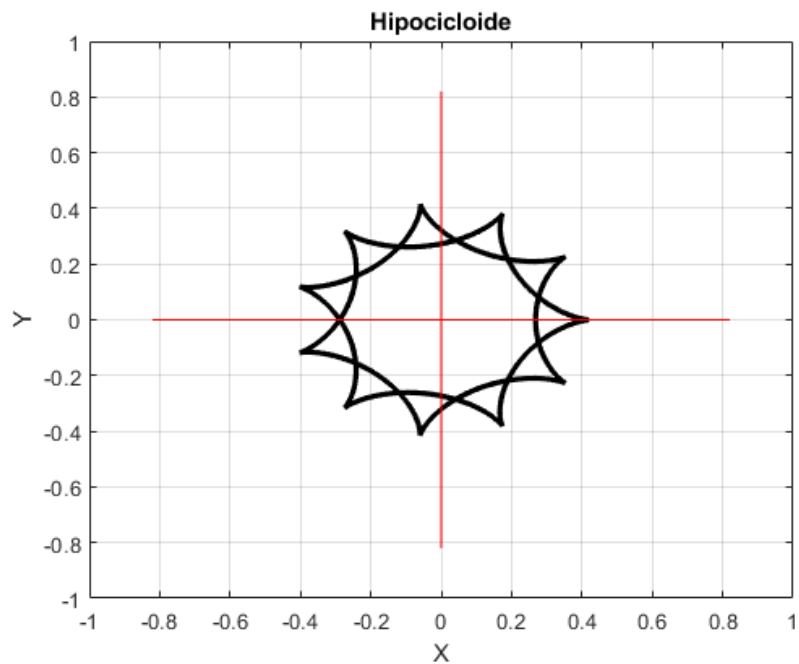


Figura 3.11: Hipocicloide, $k=5.5$. $r_1 = 0.42$; $r_2 = 0.07636$

Espiral de Arquímedes

La curva de Arquímedes recibe su denominación en honor al matemático griego Arquímedes (siglo III A.C.). La definición matemática puede describirse de la siguiente manera, desde un punto del plano situado en el extremo de una semirrecta, la curva se genera de manera uniforme a medida que se desplaza sobre ella, mientras que la semirrecta gira uniformemente en torno a uno de sus extremos, como se representa en la figura 3.12.

$$x = at\cos(t)$$

$$y = at\sin(t)$$

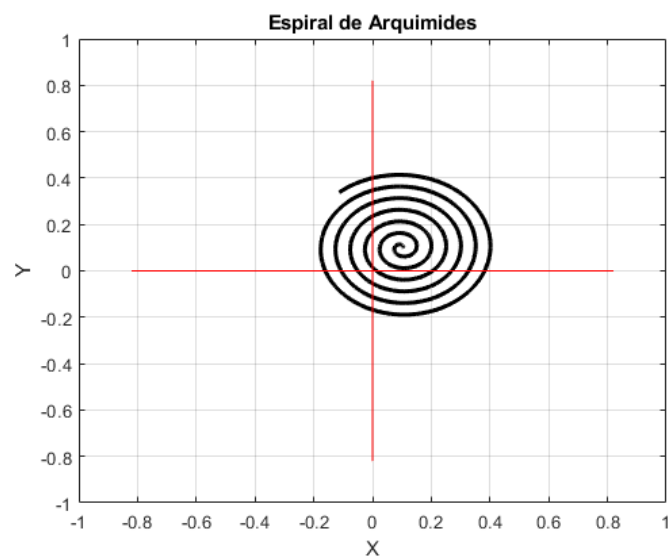


Figura 3.12: Espiral de Arquimides. $a = 0.01$

Lemniscata de Gerono

Una lemniscata (figura: 3.13) es el lugar geométrico de los puntos del plano, tales que el producto de sus distancias a dos puntos fijos llamados focos es constante.

$$x = \cos(\omega)$$
$$y = \frac{\sin(2\omega)}{2}$$

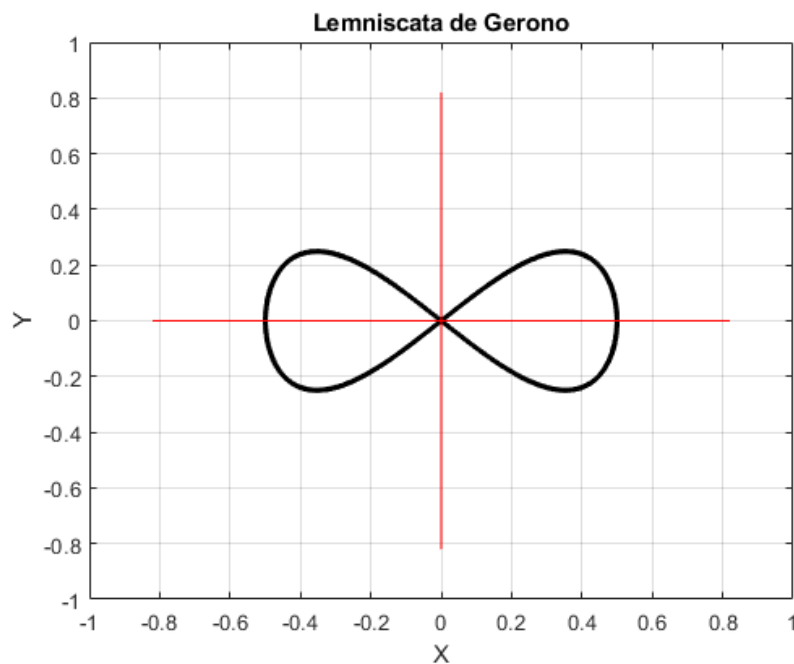


Figura 3.13: Lemniscata de Gerono.

Curvas de Lissajous

Las figuras o curvas de Lissajous fueron descubiertas por el físico francés Jules Antoine Lissajous (1822-1880), que empleaba sonidos diferentes para hacer vibrar un espejo. La luz reflejada en el espejo trazaba una curva cuya forma dependía de la frecuencia del sonido. Obteniendo de este modo lo que es la superposición de dos movimientos armónicos simples.

Las ecuaciones paramétricas más simples para obtener las figuras de Lissajous son:

$$x = \sin(a * \omega)$$
$$y = \sin(b * \omega)$$

A partir de ir modificando los valores de a y b se pueden ir creando diferentes figuras, desde figuras cerradas y figuras que no son cerradas.

Para fines de experimentación para seguimiento de trayectoria, las figuras cerradas son las de interés. Hay algunas características que se pueden notar al ir modificando los valores antes mencionados.

- Cuando $a = 1$ y b (figura: 3.14) es un número par (2,4,6,...), la curva es cerrada y alcanza b crestas. Cuando es impar, es una curva no cerrada (figura: 3.15).

- Cuando b es un número par y a es impar, o viceversa, la curva es cerrada figura 3.16 y figura 3.17. También se puede observar que alcanza b crestas en la parte superior e inferior, mientras que en los laterales tendrá a crestas.

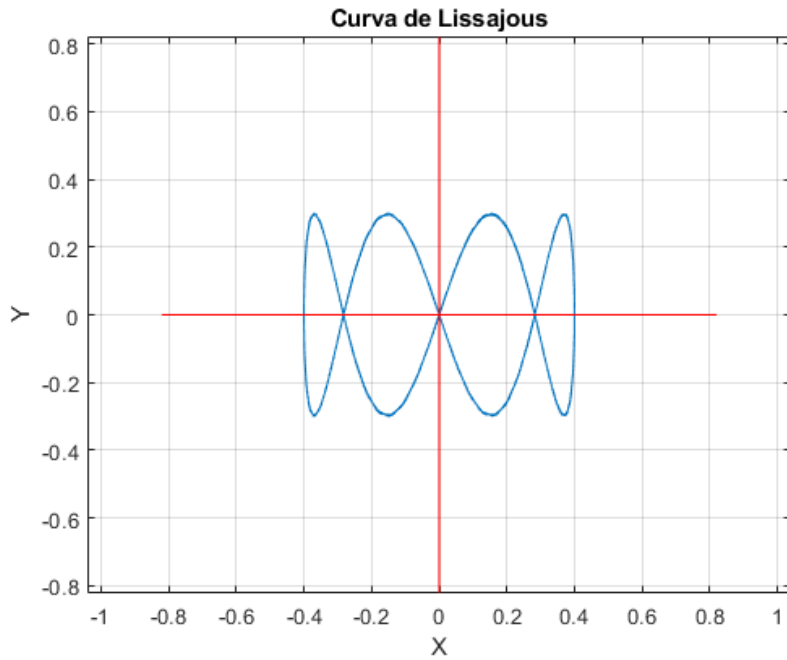


Figura 3.14: Curva de Lissajous. $a = 1; b = 4$

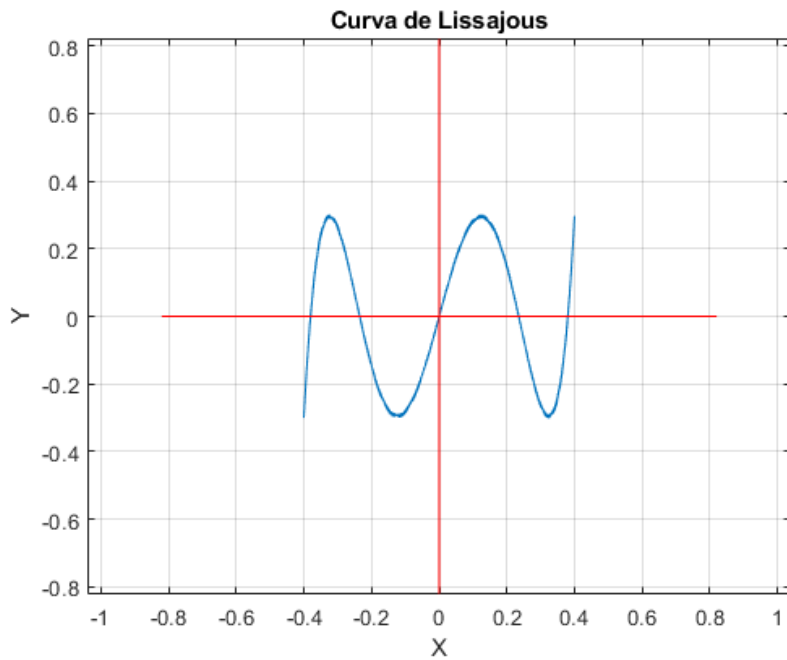


Figura 3.15: Curva de Lissajous. $a = 1; b = 5$

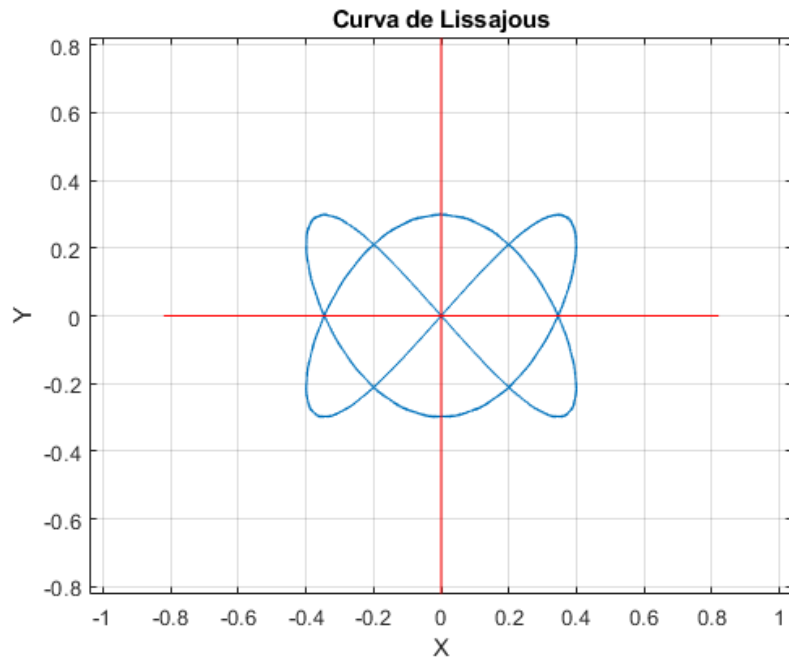


Figura 3.16: Curva de Lissajous. $a = 2; b = 3$

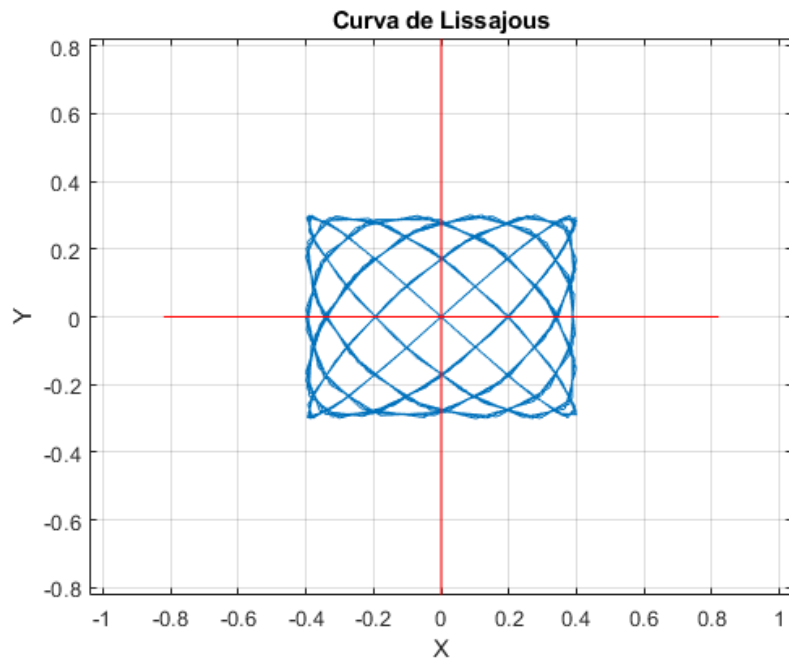


Figura 3.17: Curva de Lissajous. $a = 5; b = 6$

Como nota final para esta sección de ecuaciones paramétricas, es que no todas son candidatas debido a que puede que entren en alguna singularidad. Esto se puede corregir mediante experimentación a prueba y error, usando solamente cinemática. Habrá ocasiones en que moviendo el centro de la ecuación, también puede ser que se necesite que la figura este muy pequeña o que solo se realice en algún cuadrante del plano, por ejemplo el cuadrante IV.

3.2.3. Programa MATLAB

Gran parte de los esfuerzos del desarrollo de este proyecto, se centró en el desarrollo de un programa en MATLAB en el cuál se pudieran visualizar distintas gráficas y una animación con la cual uno pudiera observar el comportamiento del sistema, en específico, un brazo robótico con dos articulaciones rotacionales que se moverían sobre el plano cartesiano X-Y.

El programa desarrollado consta de 5 scripts que contienen información necesaria para obtener las gráficas y la animación. Dicho programa se desarrolló en el software MATLAB.

MATLAB es una plataforma de programación y cálculo numérico para analizar datos, desarrollar algoritmos y crear modelos matemáticos. El tipo de archivos con los que trabaja MATLAB es m-files, que son archivos de texto donde se guarda la secuencia de comandos, su extensión es “.m” y puede haber dos tipos: Scripts y Funciones. Los “scripts” serán de utilidad cuando se busque automatizar un bloque de comandos, estos operan sobre datos existentes y crean nuevos datos a partir de las operaciones contenidas, estos nos aceptan argumentos de entrada ni devuelven argumentos de salida, solo operan sobre datos en su área de trabajo. Las “funciones” ejecutan tareas específicas, aceptan y devuelven argumentos de salida, las variables internas son locales para su función. La sintaxis para una función es:

```
1      function [output1, output2, ...] = nombre_funcion(input1, input2, ...)
2          %comentario
3          comandos
4      end
```

MATLAB nos permitirá calcular la solución a diversas ecuaciones diferenciales ordinarias (EDO) usando sus solvers interno, sin embargo, estas deben de ser EDO de primer orden. Cuando se necesita resolver EDO de orden superior, se necesita hacer un cambio de variables manualmente antes de introducirlo en MATLAB, dicho cambio debe tener la siguiente forma:

$$\frac{dV}{dt} = f(t, V)$$

donde V es un vector con las funciones del sistema. En otras palabras, el sistema debe estar en un formato como

$$\frac{dy1}{dt} = \dots; \frac{dy2}{dt} = \dots, etc$$

El método numérico ode45 ha sido considerado para usarse dentro de este programa. Su sintaxis es la siguiente:

$$[t, x] = ode45(odefun, tspan, ci, options)$$

donde:

- x, es una matriz donde cada columna corresponde a las variables dependientes y t es el vector de tiempo.
- Odefun: es el nombre de la función.
- tspan: especifica el intervalo de tiempo. tspan=[ti,tf].
- ci: es un vector que contiene las condiciones iniciales.

- options: corresponde a una estructura que se crea con la función odeset.

En general, una de las características más importantes de esta función es que permite detener la integración numérica cuando se cumple una o varias condiciones. También se usa para determinar el nivel de error en la integración y así será para el caso de este programa.

Trayectoria en espacio cartesiano

A continuación se presentarán algunas trayectorias en espacio cartesiano que se han usado en esta investigación.

```

1      % CIRCULO
2      r=.3;           %Radio del circulo
3      xc =-.1; yc = 0; %Centro del circulo
4      x = r*cos(T) + xc;
5      y = r*sin(T) + yc;

```

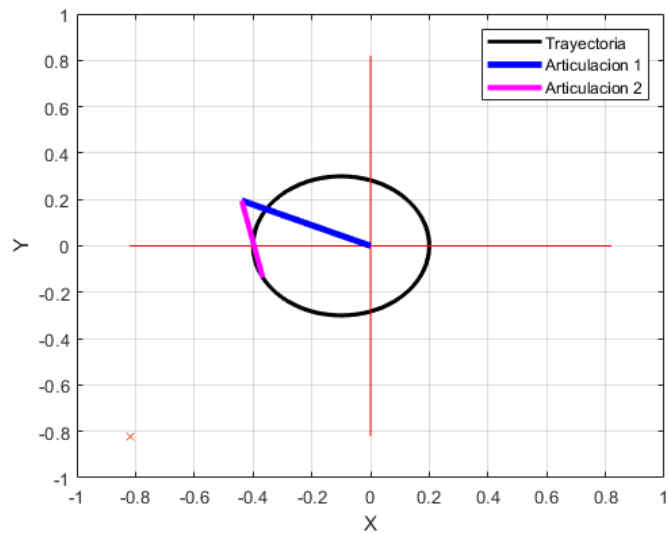


Figura 3.18: Trayectoria cartesiana: Figura en forma de circulo

```

1      % ESPIRAL
2      r=.001.*T.*T;
3      b=.14;           %0.14 es el valor minimo
4      x = (b+r) .*cos(T);
5      y = (b+r) .*sin(T);

```

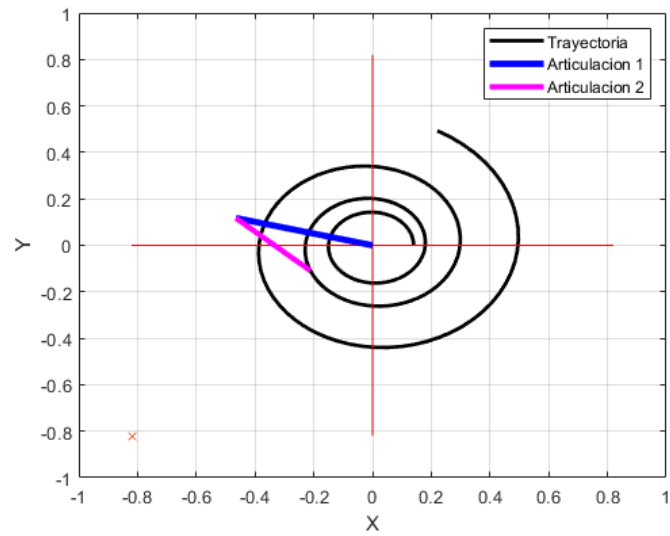


Figura 3.19: Trayectoria cartesiana: Figura en forma de espiral

```

1      % HIPOCICLOIDE
2      r1= 11; r2=12;
3      a=1-(r1/r2);
4      x=(r1-r2)*cos(T)+r2*cos(T*(a));
5      y=(r1-r2)*sin(T)+r2*sin(T*(a));

```

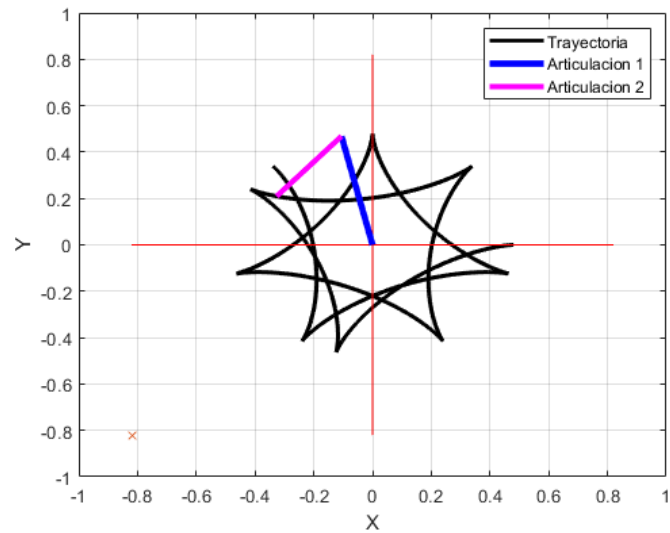


Figura 3.20: Trayectoria cartesiana: Figura hipocicloide

```

1      % HIPOCICLOIDE ESTRELLA
2      a=.6;
3      b=9;                                     % # de picos +1
4      x=(9/10*a*cos(T)+a/10*cos(b*T));
5      y=(9/10*a*sin(T)-a/10*sin(b*T));

```

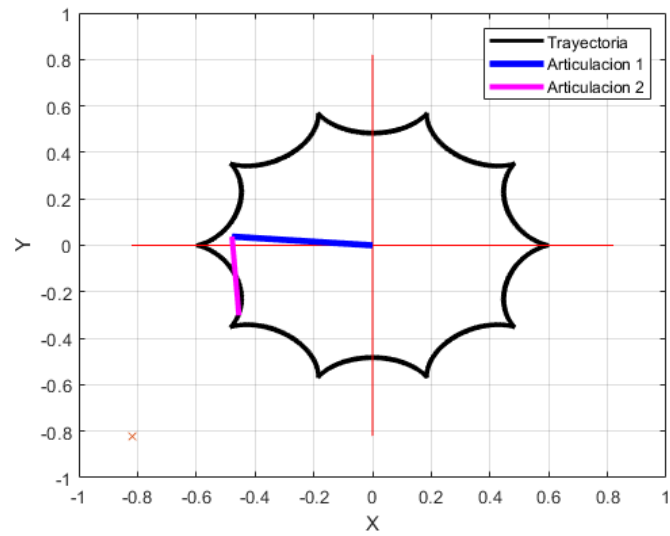


Figura 3.21: Trayectoria cartesiana: Figura hipocicloide en forma de estrella

```

1      %Flor
2      a=5;
3      x=.3*(.5*cos(T)-cos(a*T));
4      y=.3*(.5*sin(T)-sin(a*T));

```

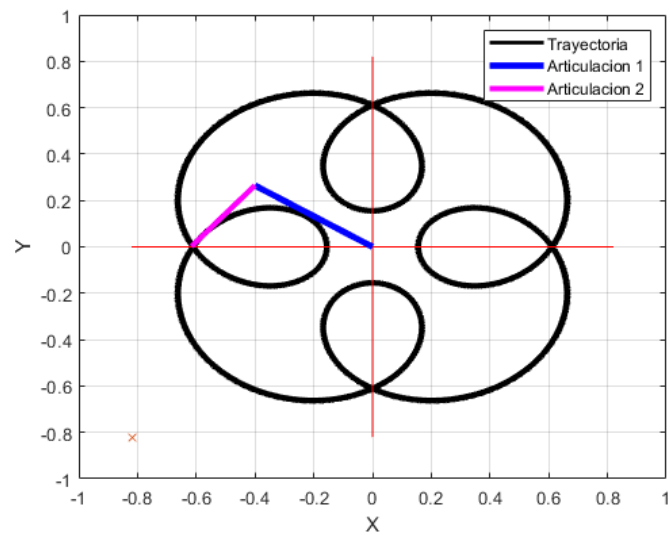


Figura 3.22: Trayectoria cartesiana: Figura en forma de flor

```

1      %Flor 2
2      a=8; b=4; %variar a&b
3      x=.1*(b*cos(T)-cos(a*T));
4      y=.1*(b*sin(T)-sin(a*T));

```

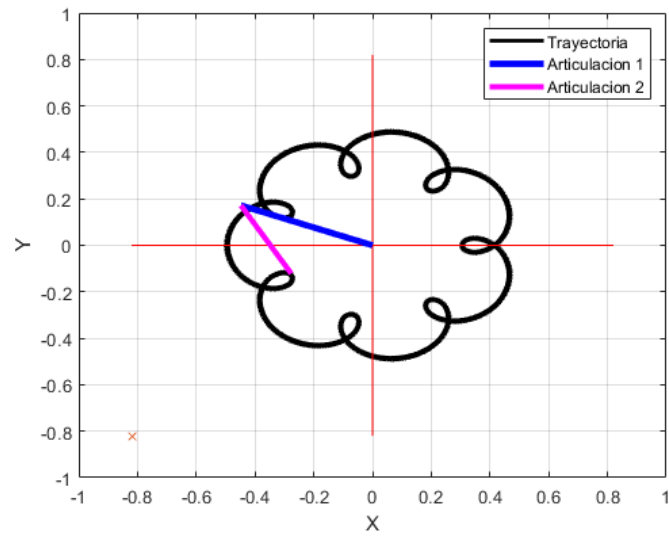


Figura 3.23: Trayectoria cartesiana: Figura en forma de flor

Cabe mencionar que en las figuras pasadas, se puede apreciar que hay dos líneas (azul y magenta) que representan a las articulaciones de un robot manipulador con articulaciones rotacionales.

3.2.4. Animación

Una animación es una larga secuencia de imágenes que difieren muy poco entre dos consecutivas y que se muestran con suficiente rapidez para producir la ilusión de un movimiento continuo. Hay varias maneras de crear animaciones en MATLAB usando funciones como: “drawnow”, “movie”, “animated line”, “comet”. La técnica usada en este programa consta de usar un ciclo for donde el tamaño de la iteración será la longitud del vector de tiempo y usando “clear figure (clf)” se borrara la figura y se agregara la nueva figura correspondiente a la iteración en curso.

A continuación, un ejemplo para una animación de una función seno:

```
1      t = 0:0.1:2*pi;           %vector de tiempo que va desde cero ...
      hasta 2pi en un intervalo de 0.1
2      y=sin(t);                % Funcion seno
3
4      for k=1:length(t)
5          plot(t(k),y(k),'x')   %Grafica "x" de la iteracion en ...
          curso de k
6          hold on
7          plot(t(1:k),y(1:k))  %Grafica todos los puntos desde 1 ...
          hasta la iteracion en curso k
8          grid on
9          axis([0 2*pi -1.5 1.5]) %Los valores en los ejes
10         xlabel('t'); ylabel('y') %Nombre de los ejes
11         pause(0.1)           %Pausa el programa 0.1 segundos
12         if k ≠ length(x)     %Borra la figura a excepcion de la ...
            clf                 ultima iteracion.
13
14         end
15 end
```

Animación: Trayectoria cartesiana

Para la animación de la trayectoria cartesiana, lo primero que se hace es determinar la trayectoria que se desea, luego se tienen que definir las ecuaciones paramétricas correspondientes a la trayectoria. Dichas ecuaciones describen el movimiento que el efector final debe de seguir. Posteriormente, se obtiene mediante la cinemática inversa el valor del ángulo de las articulaciones para cada instante de tiempo definido, finalmente se genera la animación.

```
1      %% ANIMACION Trayectoria Cartesiana
2
3      clear; close all; clc;
4
5      l1 = 0.48; l2 = 0.34;    %Largo de las artuclaciones
6      tam = l1+l2;
7
8      T = 0:0.1:20;
9
10     %Ecuacion parametrica
11     r = 0.3;                 %Radio del circulo
12     x = r*cos(T);
13     y = r*sin(T);
```

```

14
15 disp("Circulo")
16
17 %Cinematica Inversa
18 for i=1:length(T)
19     n=(x(i).^2)+(y(i).^2)-(l1^2)-(l2^2);
20     d=2*l1*l2;
21     D=n/d; a=sqrt(1-(D^2));
22     qd_2=atan2(a,D);
23     n2 = l2*sin(qd_2);
24     d2 = l1+(l2*cos(qd_2));
25     qd_1=(pi/2)+ atan2(y(i),x(i)) - atan2(n2,d2);
26     qd1(i)=qd_1;      qd2(i)=qd_2;
27 end
28
29 %Animacion
30 for k=1:length(qd1)
31     %Posicion inicial de la articulacion
32     x0=0;                      y0=0;
33     %Articulacion 1
34     x1= l1*sin(qd1(k));        y1 = -l1*cos(qd1(k));
35
36     %Articulacion 2
37     x2= l1*sin(qd1(k))+l2*sin(qd1(k)+qd2(k));    y2 = ...
38         -l1*cos(qd1(k))-l2*cos(qd1(k)+qd2(k));
39
40 plot(x,y,'k'); hold on;          %Figura
41 plot([x0,x1],[y0,y1],'b','Linewidth',3.5); hold on; %Articulacion 1
42 plot([x1,x2],[y1,y2],'g','Linewidth',3) ; hold on; %Articulacion 2
43
44 plot([-tam,tam],[0,0],'r');plot([0,0],[-tam,tam],'r');    hold on; ...
45     %Lineas Rojas
46
47 % Propiedades de la grafica
48 grid on
49 legend('Figura','Articulacion 1','Articulacion 2')
50 title('Animacion X Y')
51
52 pause(0.1)
53 if k ≠ length(qd1)
54     clf
55 end
56 end

```

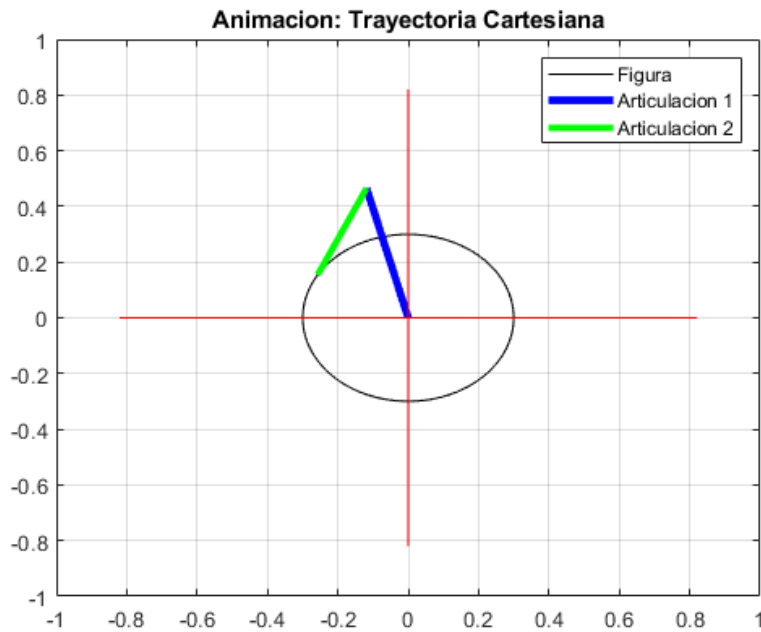


Figura 3.24: Animación generada a partir de una trayectoria cartesiana

Animación: Trayectoria articular

Para la animación de la trayectoria articular, lo primero que se hace es determinar la trayectoria que se desea, como ya se tienen los ángulos de la base de cada articulación, se procede a obtener la cinemática directa que corresponde a la posición del efector final para cada instante de tiempo, esto nos servirá para dibujar la figura que sigue el efector final (3.25). Finalmente, se procede a hacer la simulación.

```

1           %% Animacion Trayectoria Articular
2
3  clear;clc; close all;
4
5  l1 = 0.48; l2 = 0.34;
6  T = 0:.1:20;
7
8  tam = l1+l2;
9
10         %%Ecuaciones en espacio articular
11         r = 2; e = 0.5; w = 0.01;
12         qd1 = (e*T)+r;           %Articulacion 1
13         qd2 = (w*cos(T))+r;     %Articulacion 2
14
15         %Cinematica Directa
16         x = l1*sin(qd1)+l2*sin(qd1+qd2);
17         y = -l1*cos(qd1)-l2*cos(qd1+qd2);
18         %Animacion
19  for k=1:length(qd1)
20         %Posicion inicial de la articulacion
21         x0 = 0;                   y0 = 0;
22         %Articulacion 1
23         x1 = l1*sin(qd1(k));      y1 = -l1*cos(qd1(k));
24         %Articulacion 2

```

```

25 x2 = l1*sin(qd1(k))+l2*sin(qd1(k)+qd2(k));      y2 = ...
    -l1*cos(qd1(k))-l2*cos(qd1(k)+qd2(k));
26
27 plot(x,y,'k'); hold on;                          %figura
28 plot([x0,x1],[y0,y1],'b','Linewidth',3.5); hold on; %Articulacion 1
29 plot([x1,x2],[y1,y2],'g','Linewidth',3); hold on; %Articulacion 2
30
31 plot([-tam,tam],[0,0],'r');plot([0,0],[-tam,tam],'r'); hold on; ...
    %LineasRojas
32
33 % Propiedades de la grafica
34 grid on
35 legend('Figura','Articulacion 1','Articulacion 2')
36 title('Animacion: Trayectoria Articular')
37
38 pause(0.1)
39     if k ≠ length(qd1)
40         clf
41     end
42 end

```

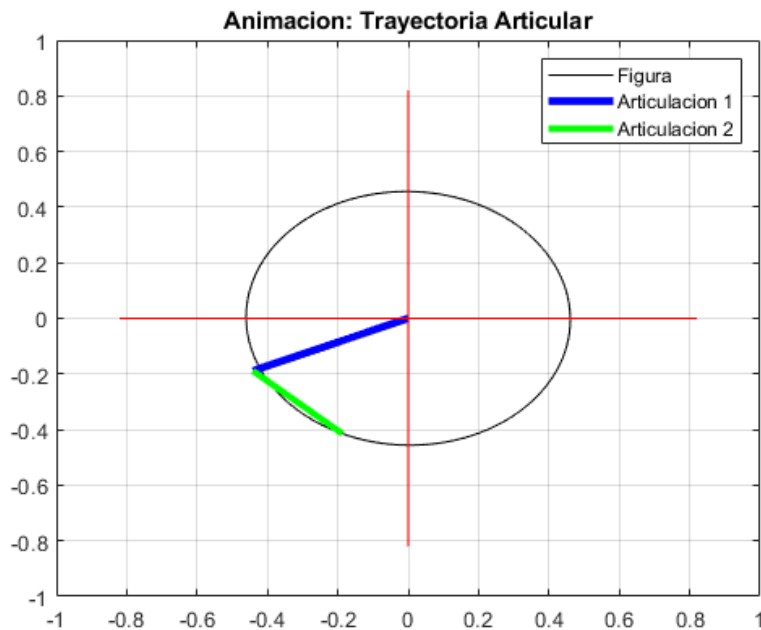


Figura 3.25: Animación generada a partir de una trayectoria articular

3.3. Programa final

A continuación se presentará el programa que se desarrolló a lo largo de este trabajo de investigación; las características principales de este programa son las siguientes:

1. Visualización de comportamiento del robot manipulador con dos articulaciones rotacionales.
2. Comparación visual de la trayectoria deseada y de la trayectoria real.
3. Límites físicos del robot manipulador que se encuentra en laboratorio de mecatrónica de la Universidad Autónoma de Baja California(Figura: 3.26).
4. Gráficas de posiciones y velocidades articulares.
5. Gráficas de los torques, los errores de posición y velocidad, y de las perturbaciones.
6. Diagramas de fase.
7. Gráfica de energía del sistema.

El programa consta de un *script* y 5 *funciones*.



Figura 3.26: Robot SCARA de UABC

3.3.1. Funciones del programa

Parámetros_Robot

Esta función contiene los parámetros del robot manipulador tipo SCARA del laboratorio de mecatrónica de la Universidad Autónoma de Baja California.

```
1 function [l1,l2,m1,m2,lc1,lc2,I1,I2,g]=Parametros_Robot ()
2 %Art1           %Art2
3 %Masa de cada eslabon
4 m1=4;          m2=3.5;
5
6 %Longitud de cada eslabon
7 l1=0.48;       l2=0.34;
8
9 %Longitud al centro de gravedad
10 lc1=0.22;     lc2=0.17;
11
12 %Inercia
13 I1=0.5;       I2=0.5;
14
15 %gravedad
16 g=9.8065;
17
18 end
```

Trayectoria _articular

Esta función recibe el tiempo T y regresa las posiciones, velocidades y aceleraciones **deseadas**.

```
1 function [qd1,qd2,dqd1,dqd2,ddqd1,ddqd2]= Trayectoria_Articular(T)
2
3 % Senos & Cosenos
4 % 1er ARTICULACION      2da ARTICULACION
5
6     qd1 = 0.3*sin(T);      qd2 = 0.3*cos(T);      %Posicion_Deseada
7     dqd1 = 0.3*cos(T);    dqd2 = -0.3*sin(T);    %Velocidad_Deseada
8     ddqd1 = -0.3*sin(T);  ddqd2 = -0.3*cos(T);    %Aceleracion_Deseada
9
10 % -----
11 % Circulo
12 %     r = 2; e = 0.5; w = 0.01;
13 % 1er ARTICULACION      2da ARTICULACION
14 %     qd1 = (e*T)+r;      qd2 = (w*cos(T))+r;    %Posicion_Deseada
15 %     dqd1 = e;           dqd2 = -w*sin(T);      %Velocidad_Deseada
16 %     ddqd1 = 0 ;        ddqd2 = -w*cos(T);    %Aceleracion_Deseada
17 end
```

Como se podrá notar, hay una sección que está comentada, es porque en esta función esta puesta una segunda trayectoria con la cual también se puede experimentar.

Para “comentar” y “descomentar” rápidamente una serie de líneas de código, se recomienda usar los accesos rápidos *Ctrl+r* y *Ctrl+t*.

Trayectoria _cartesiana

```
1 function [qd1,qd2,dqd1,dqd2,ddqd1,ddqd2]= Trayectoria_Cartesiana(T)
2 [l1,l2]=Parametros_Robot() ;
3
4 %Corazon
5 %Ecuacion Parametrica
6 xc=0.2; yc=-0.4;
7 s=0.01; % con esta variable se escala el corazon
8 x=s*(16*sin(T).^3)+xc;
9 y=s*(13*cos(T)-5*cos(2*T)-2*cos(3*T)-cos(4*T)) + yc;
10
11 % % %disp("Corazon")
12 dx = s*16*3*sin(T).^2.*cos(T);
13
14 dy = s*(-13*sin(T)+10*sin(2*T)+6*sin(3*T)+4*sin(4*T));
15
16 ddx = s*(16*3*6*cos(x).*sin(x).^2 - 3*cos(x).^3);
17
18 ddy = s*(-13*cos(T)+20*cos(2*T)+18*cos(3*T)+16*cos(4*T));
19 % -----
20 %% CINEMATICA INVERSA
21 n=(x.^2)+(y.^2)-(l1^2)-(l2^2);
22 d=2*l1*l2;
23 D=n/d; a=sqrt(1-(D.^2));
24 qd2=atan2(a,D);
25
26 n2 = l2*sin(qd2);
27 d2 = l1+(l2*cos(qd2));
28 qd1=(pi/2)+ atan2(y,x) - atan2(n2,d2);
29
30 %% VELOCIDAD ARTICULAR
31 dqd1 = ((cos(qd1+qd2))/(l1*sin(qd2)))*dx + ((sin(qd1+qd2))/(l1*sin(qd2)))*dy;
32 dqd2 = - ((l1*cos(qd1)+l2*cos(qd1+qd2))/(l1*l2*sin(qd2)))*dx - ...
33         ((l1*sin(qd1)+l2*sin(qd1+qd2))/(l1*l2*sin(qd2)))*dy;
34
35 %% ACELERACION ARTICULAR
36 ddqd1 = ((-sin(qd1+qd2).*sin(qd2).*(dqd1+dqd2) -...
37         cos(qd1+qd2).*cos(qd2).*dqd2)/(l1*sin(qd2).^2))*dx + ...
38         ((cos(qd1+qd2).*sin(qd2).*(dqd1+dqd2) - ...
39         sin(qd1+qd2).*cos(qd2).*dqd2)/((l1*sin(qd2).^2)))*dy + ...
40         ((l2*cos(qd1+qd2))/(l1*l2*sin(qd2).^2))*ddx + ...
41         ((l2*sin(qd1+qd2))/(l1*l2*sin(qd2)))*ddy;
42
43 ddqd2 = (((l1*sin(qd1).*dqd1 + l2*sin(qd1+qd2).*(dqd1+dqd2)).*sin(qd2) +...
44         (l1*cos(qd1) + ...
45         l2*cos(qd1+qd2)).*cos(qd2).*dqd2)/(l1*l2*sin(qd2).^2)).*dx + ...
46         ((- (l1*cos(qd1).*dqd1+l2*cos(qd1+qd2).*(dqd1+dqd2)).*sin(qd2) +...
47         (l1*sin(qd1)+l2*sin(qd1+qd2)).*cos(qd2).*dqd2)/ ...
48         (l1*l2*sin(qd2).^2))*dy - ((l1*cos(qd1) + ...
49         l2*cos(qd1+qd2))/(l1*l2*sin(qd2)))*ddx - ...
50         ((l1*sin(qd1) + l2*sin(qd1+qd2))/(l1*l2*sin(qd2)))*ddy;
51 end
```

RR _controlador _Pplus

```
1 function [tau,z1,z2,ew1,ew2]=RR_controlador_Pplus(x,q,dq,T)
2   q1=q(1);      q2=q(2);
3   %q=[q1; q2];%vector de posiciones
4   dq1=dq(1);   dq2=dq(2);
5   %dq=[dq1; dq2];%vector de velocidades
6
7   theta1=x(5);  theta2=x(6);
8
9   %%   Matrices y parametros
10  [l1,l2,m1,m2,lc1,lc2,I1,I2]=Parametros_Robot(); %Parametros del Robot
11
12      %Parametros Robot RR
13      %Friccion Viscosa
14      f1=l1;  f2=8;
15      Fv = [f1 0;0 f2];
16      %Matriz de inercia
17      m11 = m1*lc1^2+m2*l1^2+2*m2*l1*lc2*cos(q2)+I1+I2;
18      m12 = m2*lc2^2+m2*l1*lc2*cos(q2)+I2;
19      m21 = m2*lc2^2+m2*l1*lc2*cos(q2)+I2;
20      m22 = m2*lc2^2+I2;
21      M = [m11 m12;m21 m22];
22
23      %GANANCIAS
24      kp1=1000; kp2=400;
25      Kp=[kp1 0; 0 kp2]; %ganancial proporcional
26      kv1=0.20*kp1; kv2=0.20*kp2;
27      Kv=[kv1 0;0 kv2]; %ganancia derivativa
28
29      %% Trayectoria Deseada Robot RR
30
31      [qd1,qd2,dqd1,dqd2,ddqd1,ddqd2] = Trayectoria_Articular(T); ...
32      %   %Trayectorias_articulares
33      %   [qd1,qd2,dqd1,dqd2,ddqd1,ddqd2] = Trayectoria_Cartesiana(T); ...
34      %   %Trayectorias_cartesianas
35
36      %Matrices de Trayectorias deseadas
37      qd = [qd1;qd2];
38      Dqd = [dq1;dq2];
39      Ddqd = [ddqd1;ddqd2];
40
41      %Matriz de fuerzas centrifugas y de coriolis con velocidad DESEADA
42      c11d = -m2*l1*lc2*sin(q2)*(dq2);
43      c12d = -m2*l1*lc2*sin(q2)*(dq1+dq2);
44      c21d = m2*l1*lc2*sin(q2)*(dq1);
45      c22d = 0;
46      Cd = [c11d c12d;c21d c22d];
47
48      %Vector de errores
49      z1=q1-qd1;
50      z2=q2-qd2;
51      Z1=[q1-qd1;q2-qd2];
52      Z=[z1;z2];
53
54      %% Estimador de Perturbaciones
55      %Disturbance estimator gain
```

```

54 gamma1 = 100;
55 gamma2 = 100;
56     %Ganancias basadas en el observador de Luenberger
57 L1 = 2;
58 L2 = 1;
59
60     ew1 = gamma1*z1+L1*theta1;
61     ew2 = gamma2*z2+L2*theta2;
62 estw = [ew1;ew2];
63
64 %% %*CONTROLADOR*
65 %tau = Kp*Z-Kv*Dqd ; % Control PD
66 %tau = M*Ddqd+Cd*Dqd+Fv*Dqd-Kp*Z; %ControlP+ sin perturbacion
67 tau = M*Ddqd+Cd*Dqd+Fv*Dqd-Kp*Z1-estw; %ControlP+ con perturbacion
68 end

```

RR _robot2gdl _dinámica

```
1     function [dx] = RR_robot2gdl_dinamica(T,x)
2     q1=x(1);      q2=x(2);          %posicion
3     dq1=x(3);    dq2=x(4);        %velocidades
4     theta1=x(5);
5     theta2=x(6);
6
7     %vector de posiciones
8     q=[q1;q2];
9     %vector de velocidades
10    dq=[dq1;dq2];
11
12    %% Parametros R_Scara
13    [l1,l2,m1,m2,lc1,lc2,I1,I2]=Parametros_Robot();
14
15    %Matriz de Inercia
16    m11 = m1*lc1^2+m2*l1^2+2*m2*l1*lc2*cos(q2)+I1+I2;
17    m12 = m2*lc2^2+m2*l1*lc2*cos(q2)+I2;
18    m21= m2*lc2^2+m2*l1*lc2*cos(q2)+I2;
19    m22= m2*lc2^2+I2;
20    M= [m11 m12;m21 m22];
21    %Matriz de fuerzas centrifugas y de coriolis
22    c11 = -m2*l1*lc2*sin(q2)*(dq2);
23    c12 = -m2*l1*lc2*sin(q2)*(dq1+dq2);
24    c21 = m2*l1*lc2*sin(q2)*(dq1);
25    c22 = 0;
26    C=[c11 c12; c21 c22];
27    %vector de fricciones
28    f1=11; f2=8;
29    Fv=[f1 0;0 f2];
30
31    %% Controlador
32
33    [tau,z1,z2,ew1,ew2]=RR_controlador_Pplus(x,q,dq,T);
34
35    %Perturbacion
36    W=[1;1];
37
38    %% SISTEMA
39    ddq=M\(-C*dq-Fv*dq+tau+W); %RR_Robot
40
41    dx = zeros(6,1);
42    dx(1) = dq1;      %posicion 1
43    dx(2) = dq2;      %posicion 2
44    dx(3) = ddq(1);  %vel 1
45    dx(4) = ddq(2);  %vel 2
46
47    dx(5) = z1;
48    dx(6) = z2;
```

Script principal

```
1      clear; clc; close all;
2
3      options = odeset('RelTol', 1e-4, 'AbsTol', [1e-4 1e-4 1e-4 1e-4 1e-4 1e-4]);
4      t =20 ;
5      %Condiciones Inicales en radianes
6      %art1      art2
7      Ang1=-pi/20;      Ang2=pi/6;      % (-0.065)&(1.047)
8      vell=0;      vel2=0;
9
10     [T,X] = ode23(@RR_robot2gdl_dinamica,[0 t],[Ang1 Ang2 vell vel2 0 ...
11         0],options);
12     %%estados del sistema
13     q1=X(:,1);      q2=X(:,2);
14     dq1=X(:,3);      dq2=X(:,4);
15     theta1=X(:,5);
16     theta2=X(:,6);
17
18     [l1,l2,m1,m2,lc1,lc2,I1,I2,g]=Parametros_Robot(); %Parametros del Robot
19
20     %% Trayectoria Deseada Robot RR
21     [qd1,qd2,dqd1,dqd2,ddqd1,ddqd2]= Trayectoria_Articular(T);
22     % [qd1,qd2,dqd1,dqd2,ddqd1,ddqd2]= Trayectoria_Cartesiana(T);
23
24     num=length(q1);
25     tam=0.9;
26     [n,m]=size(T);
27     %% Calculos varios
28     %Errores
29     e1=q1-qd1;
30     e2=q2-qd2;
31     De1=dq1-dqd1;
32     De2=dq2-dqd2;
33
34     %CINEMATICA
35     xc=l1*sin(q1)+l2*sin(q1+q2); %para asterisco
36     yc=-l1*cos(q1)-l2*cos(q1+q2); %
37     xdc=l1*sin(qd1)+l2*sin(qd1+qd2); %Deseado
38     ydc=-l1*cos(qd1)-l2*cos(qd1+qd2);
39
40
41     %% ANIMACIONES
42     %
43     fig=figure(1);
44     for d=1:length(q1)
45         set(gcf,'position',[50 350 1000 600])
46
47     %      CALCULOS
48     % Sistema Real
49         %1er articulacion      %2da articulacion
50     x0=0;      y0=0;
51     x1 = l1*sin(q1(d));      x2 = x1 + l2*sin(q1(d)+q2(d));
52     y1 = -l1*cos(q1(d));      y2 = y1 - l2*cos(q1(d)+q2(d));
53     %Valores DESEADOS
```

```

54     xd0=0;                                yd0=0;
55     xd1 = l1*sin(qd1(d));                xd2 = xd1 + l2*sin(qd1(d)+qd2(d));
56     yd1 = -l1*cos(qd1(d));                yd2 = yd1 - l2*cos(qd1(d)+qd2(d));
57     %-----
58     %posicion especifica de las articulaciones
59     xe1=l1*cos(q1(1));                    xe2=xe1+l2*cos(q1(1)+q2(1));
60     ye1=l1*sin(q1(1));                    ye2=ye1+l2*sin(q1(1)+q2(1));
61
62     limx1= (.7)*sin(2.4);  limx2= (.7)*sin(3.85);  % limite art1
63     limy1=-(.7)*cos(2.4);  limy2=-(.7)*cos(3.85);
64
65     limx3= (l1)*sin(q1(d))+l2*sin(q1(d)-.2);      limx4 = ...
66     (l1)*sin(q1(d))+l2*sin(q1(d)+2.5);  %limite art2
67     limy3=- (l1)*cos(q1(d))-l2*cos(q1(d)-.2);    limy4 ...
68     =-(l1)*cos(q1(d))-l2*cos(q1(d)+2.5);
69     %-----
70     % ANIMACIONES
71     plot([-tam,tam],[-tam,tam],'x'); hold on;
72     plot([-tam,tam],[0,0],'r');plot([0,0],[-tam,tam],'r');    ...
73     hold on;          %linea roja en x&y
74
75     plot([x0 x1 x2] , [y0,y1,y2] , 'k.-','linewidth',3); ...
76     hold on;          %Sistema Real
77     plot([xd0 xd1 xd2], [yd0 yd1 yd2], 'r.-','linewidth',3); ...
78     hold on;          %Sistema Deseado
79     %legend('Posicion Deseada','Poscion Actual');
80     plot(xdc(d),ydc(d), '*r'); hold on;          %posicion deseada ...
81     asterisco
82     plot(xc(d),yc(d), '*k');          hold on;          %posicion real ...
83     asterisco
84
85     plot(xdc(1:d),ydc(1:d), '-r');          hold on;          ...
86     %posicion deseada linea
87     plot(xc(1:d),yc(1:d), 'k');          hold on;
88     % lgd=legend; %('Posicion Deseada','Poscion Actual');
89     %-----
90     %posicion especifica de las articulaciones
91     %plot([y0,ye1,ye2],[-x0,-xe1,-xe2] , 'b--','linewidth',2 );
92
93     %plot([x0,x1,x2] , [y0,y1,y2], 'b--','linewidth',2 ); hold on;
94     plot([x0,limx1], [y0,limy1], 'g--','linewidth',3); hold on;
95     plot([x0,limx2], [y0,limy2], 'g--','linewidth',3);hold on;
96     plot([x1,limx3], [y1,limy3], 'g--','linewidth',3); hold on;
97     plot([x1,limx4], [y1,limy4], 'g--','linewidth',3);hold on;
98     %-----
99     % Graph properties
100
101     xlim([-tam tam]);          ylim([-tam tam]);
102     xticks(-tam:0.1:tam);          yticks(-tam:0.1:tam);
103     grid on;
104     title('Animacion X Y');
105
106     % LETREROS
107
108     % Tiempo
109     posX = -(tam-0.05); posY = (tam-0.05);
110     txtTime = text(posX, posY, 's', 'fontsize', 8);
111     timeSimulation = (t*d)/length(q1);
112     set(txtTime , 'string', ['Tiempo:' ...
113     num2str(timeSimulation) 's']);

```

```

103         %Angulos
104         pos2X = -(tam-0.05); pos2Y = (tam-.1);
105         txtQ1=text(pos2X, pos2Y, 's', 'fontsize',8);
106         set(txtQ1, 'string', ['Angulos Q1:' num2str(q1(d))]);
107         pos2X2 = -(tam-0.05); pos2Y2 = (tam-.15);
108         txtQ2=text(pos2X2, pos2Y2, 's', 'fontsize',8);
109         set(txtQ2, 'string', ['Angulos Q2:' num2str(q2(d))]);
110         grid on;
111     %
112     time_pause=0.001;
113     pause(time_pause);
114
115     F(d) = getframe(gcf);
116
117     frame=getframe(fig);
118     im{d} = frame2im(frame);
119
120
121     if d ≠ length(q1)
122         clf
123     end
124 end
125
126 %% GRAFICAS POSICIONES Y VELOCIDADES ARTICULARES
127 figure(2)
128 set(gcf, 'position', [1050 50 800 400])
129 subplot(2,2,1)
130 plot(T,q1,T,qd1, '--r', 'LineWidth',2);
131 legend('Posicion Actual', 'Posicion Deseada', 'Location', 'best')
132 xlabel('T [s]'); ylabel('Posicion en Radianes')
133 xlim([-1 t+2]); ylim('auto'); grid on;
134 title('Posicion del eslabon 1');
135 subplot(2,2,2)
136 plot(T,dq1,T,dqd1, '--r', 'LineWidth',2);
137 xlabel('T [s]'); ylabel('Velocidad')
138 legend('Velocidad Actual', 'Velocidad Deseada', 'Location', 'best')
139 xlim([-1 t+2]); ylim('auto'); grid on;
140 title('Velocidad del eslabon 1');
141 subplot(2,2,3)
142 plot(T,q2,T,qd2, '--r', 'LineWidth',2);
143 legend('Posicion Actual', 'Posicion Deseada', 'Location', 'best')
144 xlabel('T [s]'); ylabel('Posicion en Radianes')
145 xlim([-1 t+2]); ylim('auto'); grid on;
146 title('Posicion del eslabon 2');
147 subplot(2,2,4)
148 plot(T,dq2,T,dqd2, '--r', 'LineWidth',2);
149 xlabel('T [s]'); ylabel('Velocidad');
150 legend('Velocidad Actual', 'Velocidad Deseada', 'Location', 'best');
151 xlim([-1 t+2]); ylim('auto'); grid on;
152 title('Velocidad del eslabon 2');
153
154 %% GRAFICAS TAU, ERROR, PERTURBACIONES
155
156 tau1=zeros(n,1); tau2=zeros(n,1);
157 z1=zeros(n,1); z2=zeros(n,1);
158 ew_1=zeros(n,1); ew_2=zeros(n,1);
159 w1=1.*ones(1,n); w2=1.*ones(n,1);
160

```

```

161     for h=1:num
162         q=[q1(h);q2(h)];
163         dq=[dq1(h);dq2(h)];
164         X_x=X(h,:);
165         T_t=T(h);
166     [tau,z1,z2,ew1,ew2] = RR_controlador_Pplus(X_x,q,dq,T_t);
167
168     tau1(h)=tau(1);      tau2(h)=tau(2);
169     z1(h)=tau(1);      z2(h)=tau(1);
170     ew_1(h)=ew1(1);    ew_2(h)=ew2(1);
171
172     end
173
174     figure(3)
175     set(gcf,'position',[50 50 1000 600])
176     subplot(2,3,1)
177     plot(T,tau1,'r','LineWidth',2);
178         legend('Torque 1','Location','best')
179         xlabel('T [s]'); ylabel('N*m');
180         xlim([-1 t+2]); ylim('auto'); grid on;% axis padded;
181         title('Torque 1');
182     subplot(2,3,2)
183     plot(T,e1,'m','LineWidth',2);
184         xlabel('T [s]'); ylabel('Error de posicion')
185         legend('Error 1','Location','best')
186         xlim([-1 t+2]); ylim('auto'); grid on;%axis padded
187         title('Error eslabon 1 q-qd');
188     subplot(2,3,3)
189     plot(T,ew_1,'b',T,w1,'--r','LineWidth',2);
190         legend('Perturbaci n estimada','Perturbaci n real')
191         xlabel('T [s]'); ylabel('N.m')
192         xlim([-1 t+2]); ylim 'auto' ; grid on;%axis padded
193         title('Perturbaciones en el eslab n 1');
194     subplot(2,3,4)
195     plot(T,tau2,'r','LineWidth',2);
196         legend('Torque 2','Location','best')
197         xlabel('T [s]'); ylabel('N*m')
198         xlim([-1 t+2]); ylim 'auto'; grid on;%axis padded%
199         title('Torque 2');
200     subplot(2,3,5)
201     plot(T,e2,'m','LineWidth',2);
202         xlabel('T [s]'); ylabel('Error');
203         legend('Error2','Location','best');
204         xlim([-1 t+2]); ylim ('auto'); grid on; %axis padded%
205         title('Error eslabon2 q-qd');
206     subplot(2,3,6)
207     plot(T,ew_2,'b',T,w2,'--r','LineWidth',2);
208         legend('Perturbaci n estimada','Perturbaci n real')
209         xlabel('T [s]'); ylabel('N.m')
210         xlim([-1 t+2]); ylim('auto') ; grid on;%axis padded%
211         title('Perturbaciones en el eslab n 2');
212
213     figure(10)
214     subplot(2,1,1)
215     plot(T,ew_1,'b',T,w1,'--r','LineWidth',2);
216         legend('Perturbaci n estimada','Perturbaci n real')
217         xlabel('T [s]'); ylabel('N.m')
218         xlim([-1 t+2]); ylim('auto') ; grid on;%axis padded%

```

```

219     title('Perturbaciones en el eslabon 1');
220     subplot(2,1,2)
221     plot(T,ew_2,'b',T,w2,'--r','LineWidth',2);
222     legend('Perturbaci n estimada','Perturbaci n real')
223     xlabel('T [s]'); ylabel('N.m')
224     xlim([-1 t+2]); ylim('auto') ; grid on; %axis padded%
225     title('Perturbaciones en el eslabon 2');
226
227     %% Diagramas de fase
228     max1 =1.25*( max(e1, [], 'all')); max2 = 1.25*(max(De1, [], 'all'));
229     min1 = 1.25*(min(e1, [], 'all')); min2 = 1.25*(min(De1, [], 'all'));
230
231     max3 =1.25*( max(e2, [], 'all')); max4 = 1.25*(max(De2, [], 'all'));
232     min3 = 1.25*(min(e2, [], 'all')); min4 = 1.25*(min(De2, [], 'all'));
233     figure(4)
234     %subplot(2,1,1)
235     plot(e1,De1,'--k','LineWidth',2);
236     %legend('Perturbaci n estimada','Perturbaci n real')
237     xlabel('Error de posicion'); ylabel('Error de velocidad')
238     xlim([min1 max1]); ylim([min2 max2]); grid on;
239     title('Phase Portraits:Eslabon 1');
240     %subplot(2,1,2)
241     figure(5)
242     plot(e2,De2,'--k','LineWidth',2);
243     xlabel('Error de posicion'); ylabel('Error de velocidad')
244     xlim([min3 max3]); ylim([min4 max4]); grid on;
245     title('Phase Portraits:Eslabon 2');
246
247
248     %% GRAFICAS ENERGIA CINETICA,POTENCIAL
249     k1=zeros(num,1); k2=zeros(num,1); K=zeros(num,1);
250     u1=zeros(num,1); u2=zeros(num,1); U=zeros(num,1);
251     L=zeros(num,1); H=zeros(num,1);
252
253     for h=1:num
254         k1(h)=(1/2)*m1*lc1^2*dq1(h)^2+(1/2)*I1*dq1(h)^2; %resumida para ler ...
255         k2(h)=(1/2)*m2*(I1^2)*(dq1(h)^2)+...
256             (1/2)*m2*(lc2^2)*((dq1(h)^2)+(2*dq1(h))+(2*dq1(h)*dq2(h))+...
257             (q2(h)^2))+m2*l1*lc2*((dq1(h)^2)+(dq1(h)*dq2(h)))*cos(q2(h))+...
258             (1/2)*I2*((dq1(h)+dq2(h))^2);
259
260         u1(h)=-m1*lc1*g*cos(q1(h));
261         u2(h)=-m2*l1*g*cos(q1(h))-m2*lc2*g*cos(q1(h)+q2(h));
262         K(h)=k1(h)+k2(h); %Energia Cinetica
263         U(h)=u1(h)+u2(h); %Energia Potencial
264         L(h)=K(h)-U(h); %Lagraniano
265         H(h)=K(h)+U(h); %Hamiltoniano
266     end
267
268     figure(5)
269     set(gcf,'position',[1200 350 800 600])
270     subplot(4,1,1)
271     plot(T,K)
272     legend('Energia cinetica')
273     xlabel('T [s]'); ylabel('Joules');
274     max1 =1.25*( max(K, [], 'all'));min1 =1.25*( min(K, [], 'all'));
275     xlim([-1 t+1]); ylim([min1 max1]); grid on; xticks(-1:1:(t+2));

```

```

276     title('Energia Cinetica (K)');
277 subplot(4,1,2)
278 plot(T,U,'m')
279     legend('Energia Potencial')
280     max1 =1.25*( max(U, [], 'all'));min1 =1.25*( min(U, [], 'all'));
281     xlim([-1 t+1]); ylim([min1 max1]); grid on; xticks(-1:1:(t+2));
282     title('Energia Potencial (U)');
283 subplot(4,1,3)
284 plot(T,K,T,U,'m')
285     legend('Energia cinetica','Energia Potencial')
286     xlabel('T [s]'); ylabel('Joules')
287     max1 =1.25*( max(K, [], 'all'));min1 =1.25*( min(K, [], ...
288         'all'));
289     xlim([-1 t+2]); ylim([min1 max1]); grid on; ...
290     xticks(-1:1:(t+2));;
291     title('Energias del Sistema');
292 subplot(4,1,4)
293 plot(T,L,T,H,'k','LineWidth',3)
294     legend('Lagraniano','Hamiltoniano')
295     xlabel('T [s]'); ylabel('Joules')
296     xlim([-1 t+1]); ylim('auto'); grid on; xticks(-1:1:(t+2));
297     title('Lagraniano L=K-U');

```

Graficas

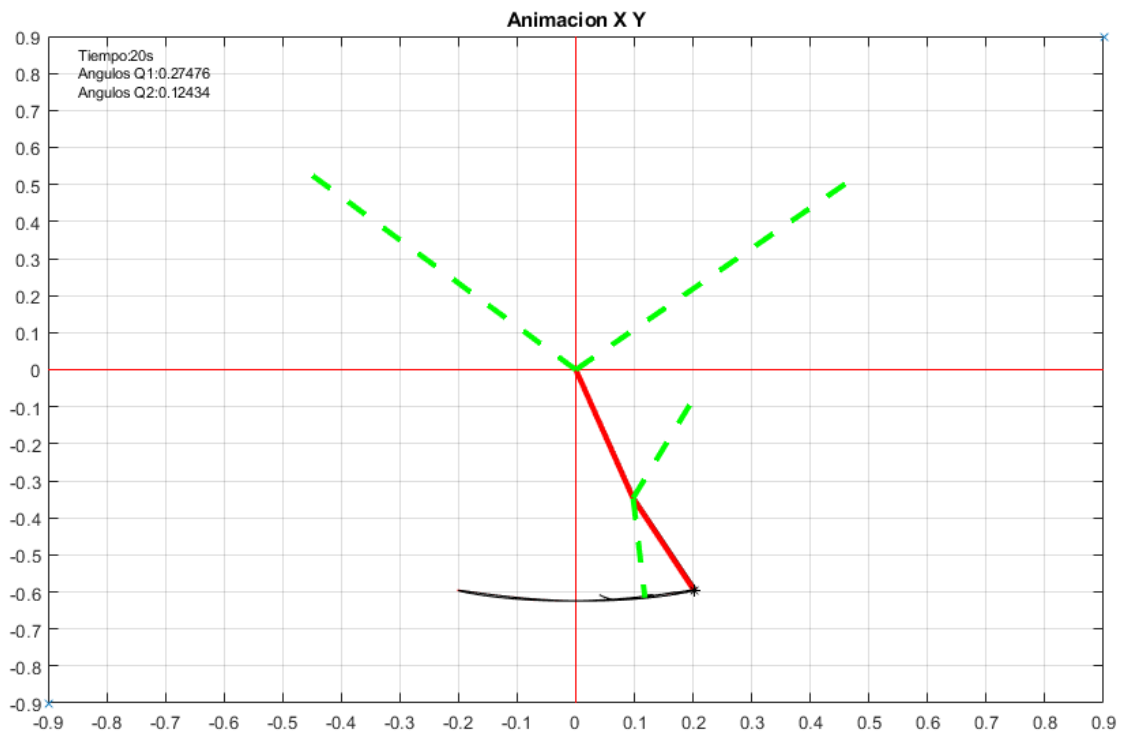


Figura 3.27: Animación vista en plano X-Y

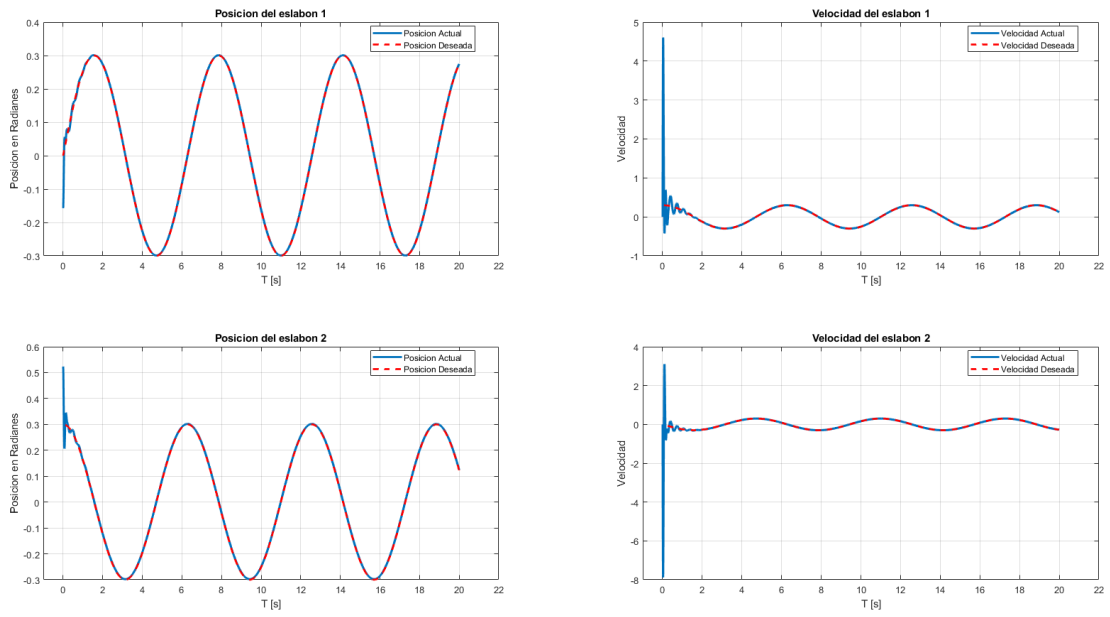


Figura 3.28: Gráfica de posiciones y velocidades articulares

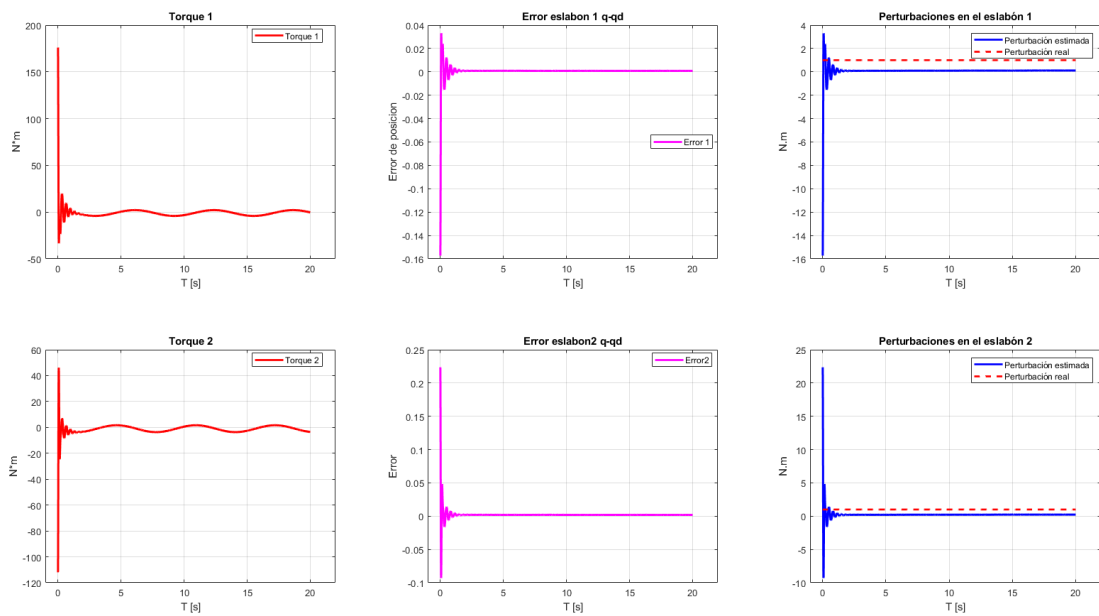


Figura 3.29: Gráficas de torque, errores y perturbaciones

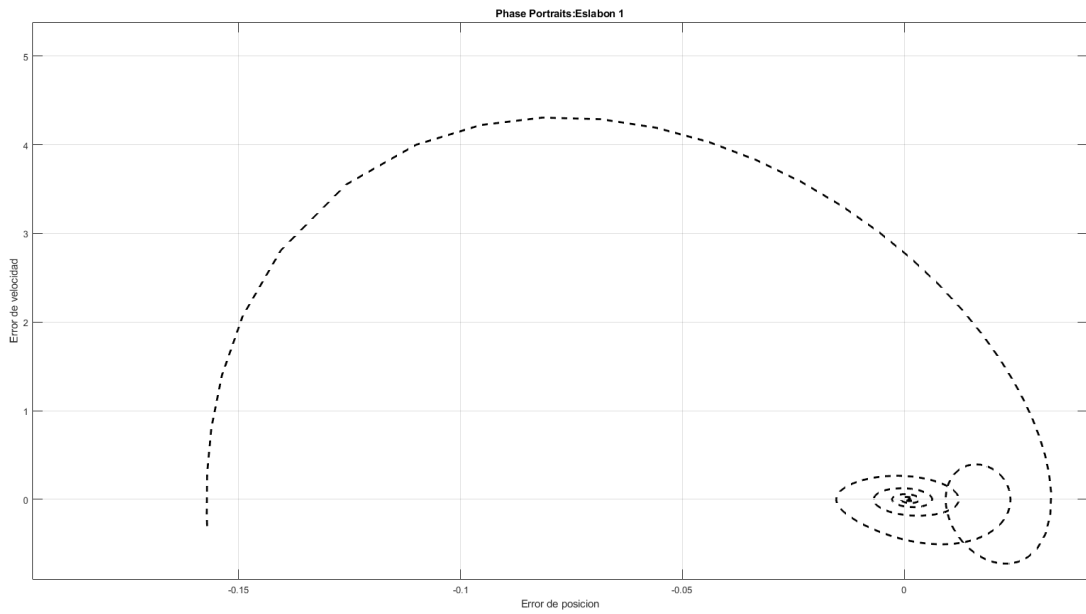


Figura 3.30: Gráfica de plano fase

Retrato de fase

Las trayectorias en el tiempo alrededor de un punto de equilibrio nos permiten identificar el tipo de equilibrio presente en el sistema. Los diagramas ilustran los comportamientos cualitativos que corresponden a cada punto de equilibrio de la ecuación; depende de la variable independiente. Es un método cualitativo gráfico que nos permite analizar una ecuación diferencial ordinaria.

Capítulo 4 | Análisis de Resultados

4.1. Simulaciones en MATLAB

En el programa desarrollado en MATLAB presentado en la sección anterior se hicieron diferentes simulaciones usando diferentes trayectorias en tanto en espacio cartesiano, así como en espacio articular.

Para todas las simulaciones, los parámetros del robot manipulador son las siguientes:

Parámetros del robot manipulador tipo SCARA				
	Descripción	Notación	Valor	Unidades
1era Articulación	Longitud	l_1	0.48	m
	Masa	m_1	4	kg
	Distancia al centro de masa	l_{c1}	0.22	m
	Fricción	fv_1	11	kg m ² /s
	Inercia	I_1	0.5	kg m ²
2da Articulación	Longitud	l_2	0.34	m
	Masa	m_2	3.5	kg
	Distancia al centro de masa	l_{c2}	0.17	m
	Fricción	fv_2	8	kg m ² /s
	Inercia	I_2	0.5	kg m ²

Tabla 4.1: Parámetros robot manipulador SCARA de UABC

A continuación se presentará una serie de trayectorias en espacio articular junto con los resultados arrojados en el simulador.

Cabe señalar que la dinámica es la que corresponde a la ecuación (3.2) y el control implementado es (3.11).

Las ganancias implementadas en el controlador son las siguientes:

Ganancias del controlador		
	Art 1	Art 2
Proporcional	1000	400
Gamma	100	100
L	2	1

Tabla 4.2: Ganancias del controlador

4.1.1. Trayectoria articular: senos y cosenos

Trayectoria articular		
	Art 1	Art 2
Trayectoria	$0.3\sin(T)$	$0.3\cos(T)$

Tabla 4.3: Parámetros de la trayectoria articular simple.

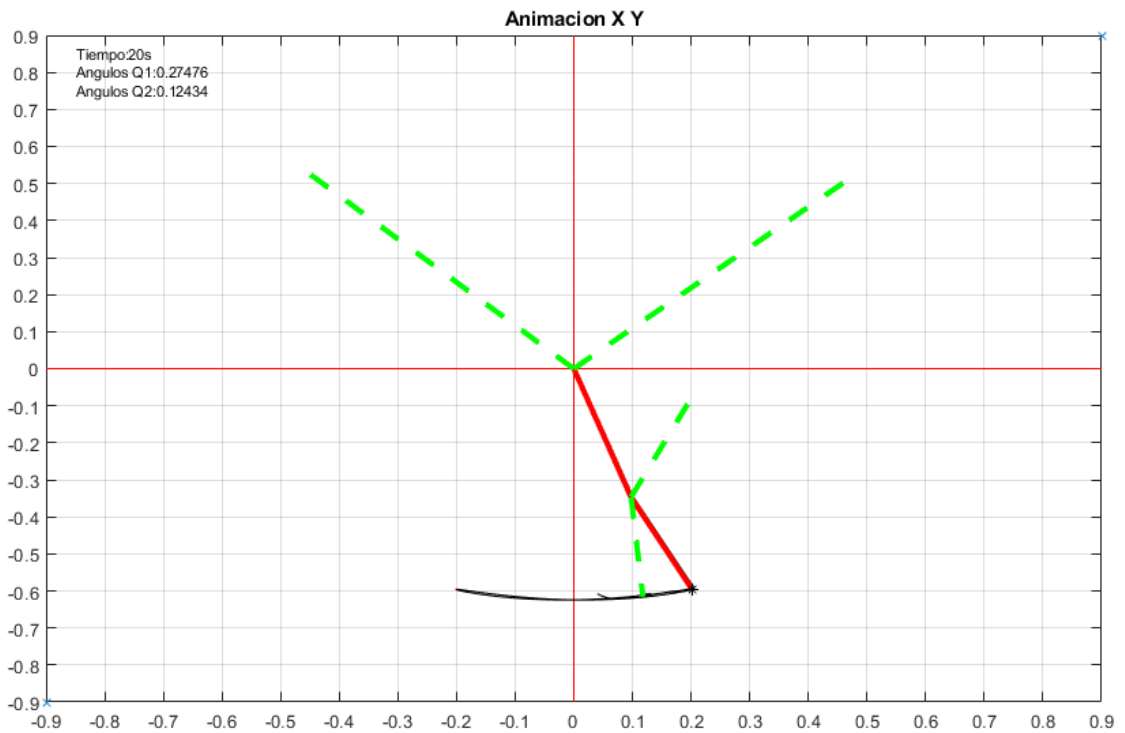


Figura 4.1: Animación vista en plano X-Y

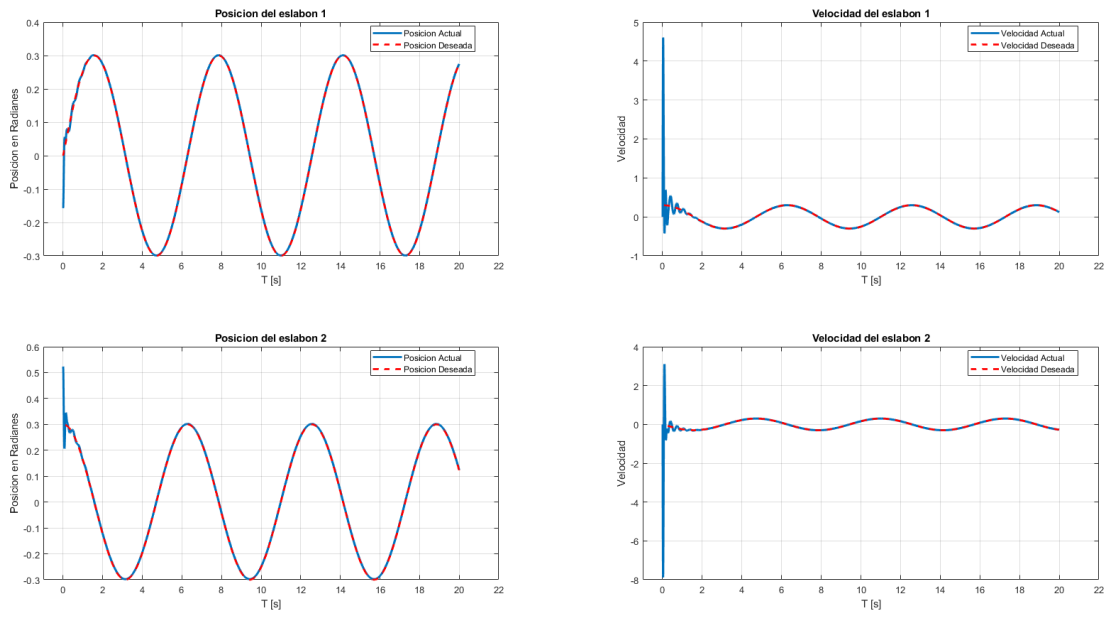


Figura 4.2: Gráfica de posiciones y velocidades articulares

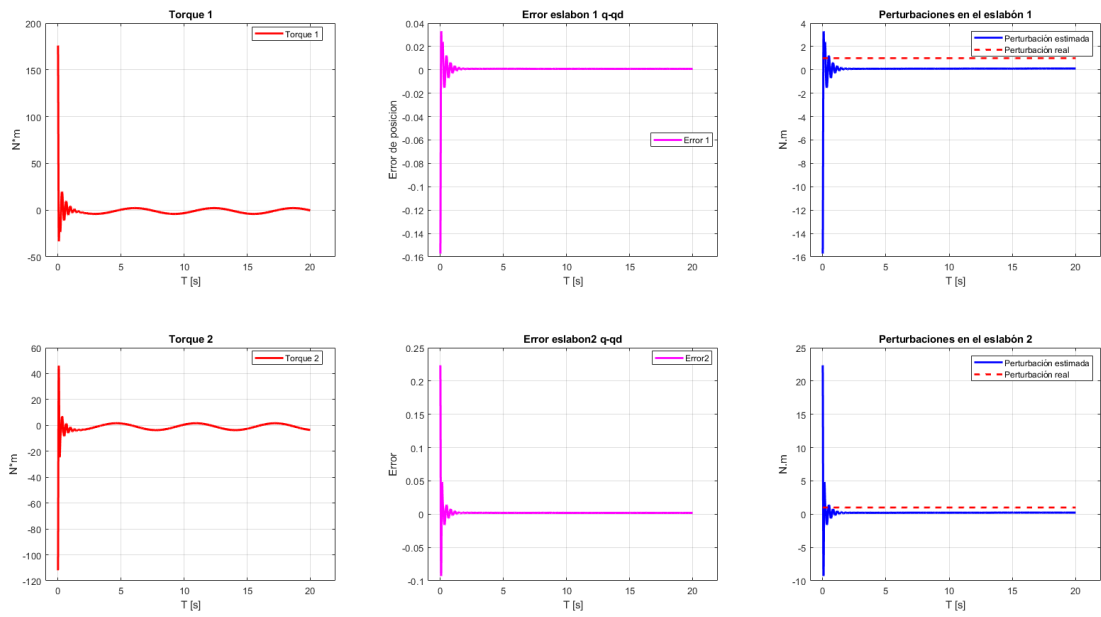


Figura 4.3: Gráficas de torque, errores y perturbaciones

4.1.2. Trayectoria articular: infinito-orejona

Trayectoria articular		
	Art 1	Art 2
Ajustes	$r_1=0.5 ; r_2=0.001; r_3=2.375$	
Trayectoria	$r_2+r_3\cos(r_1T)$	$r_2+r_3\sin(r_1T)$

Tabla 4.4: Parámetros de la trayectoria articular: infinito-modificado.

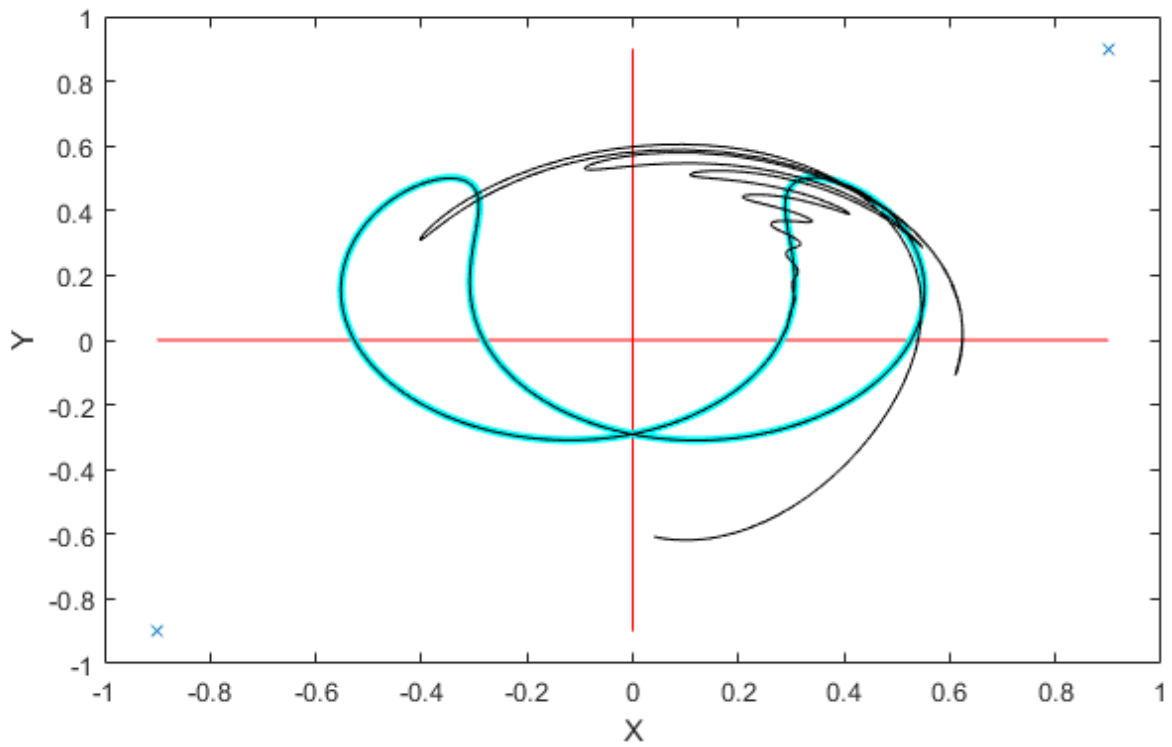


Figura 4.4: Trayectoria articular: infinito-orejona

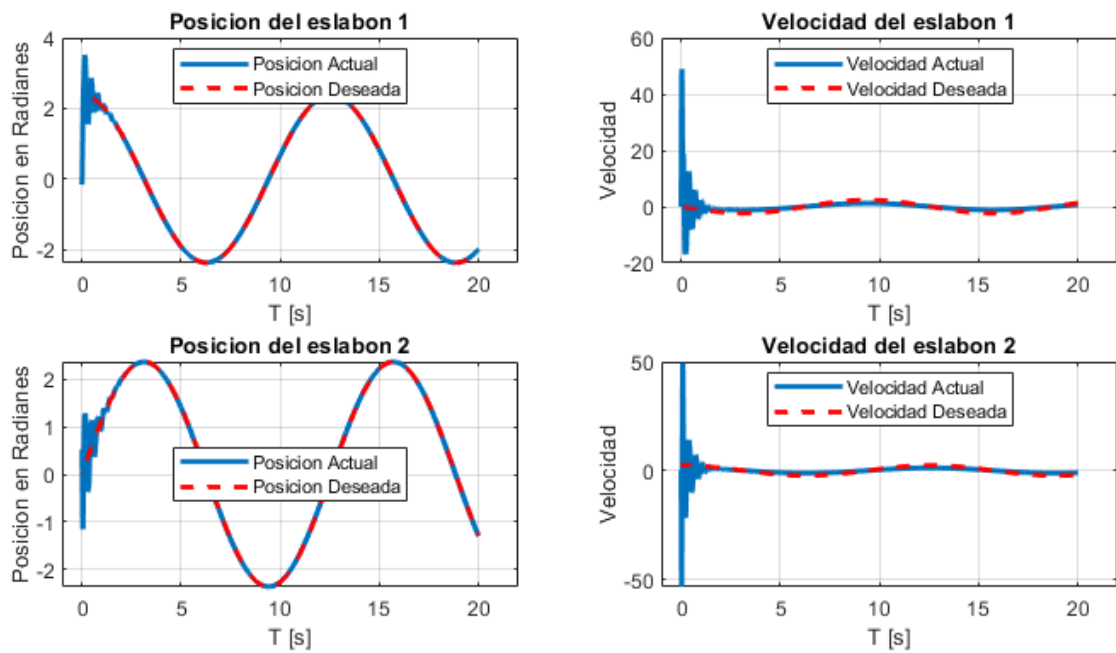


Figura 4.5: Trayectoria articular: infinito-orejona, posición y velocidad

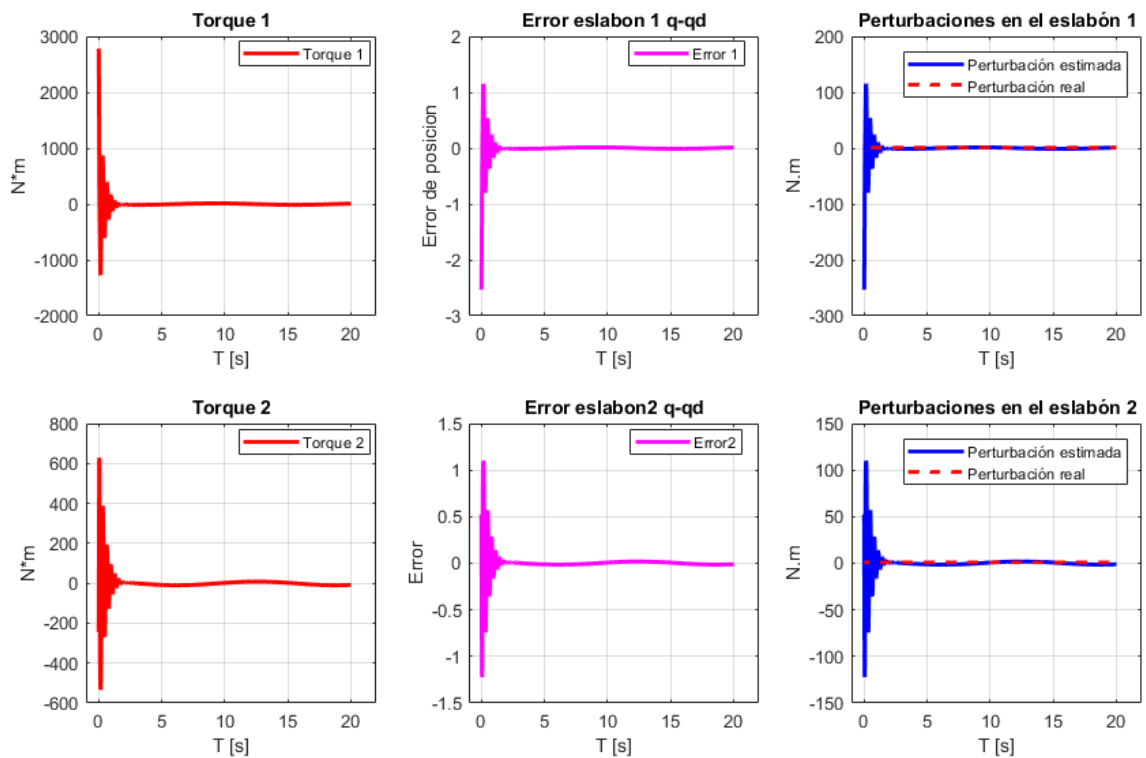


Figura 4.6: Trayectoria articular: infinito-orejona, torque, error y perturbacion

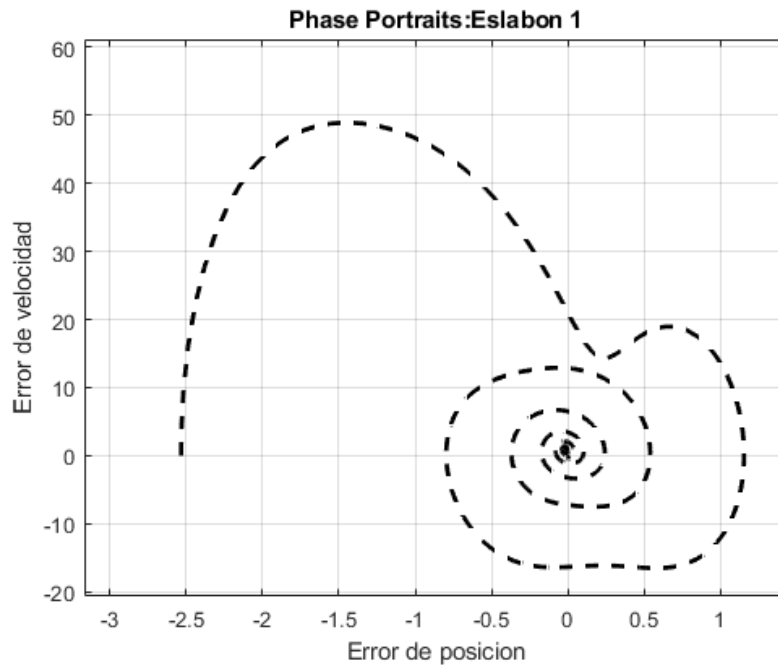


Figura 4.7: Trayectoria articular: infinito-orejona, plano fase articulación 1

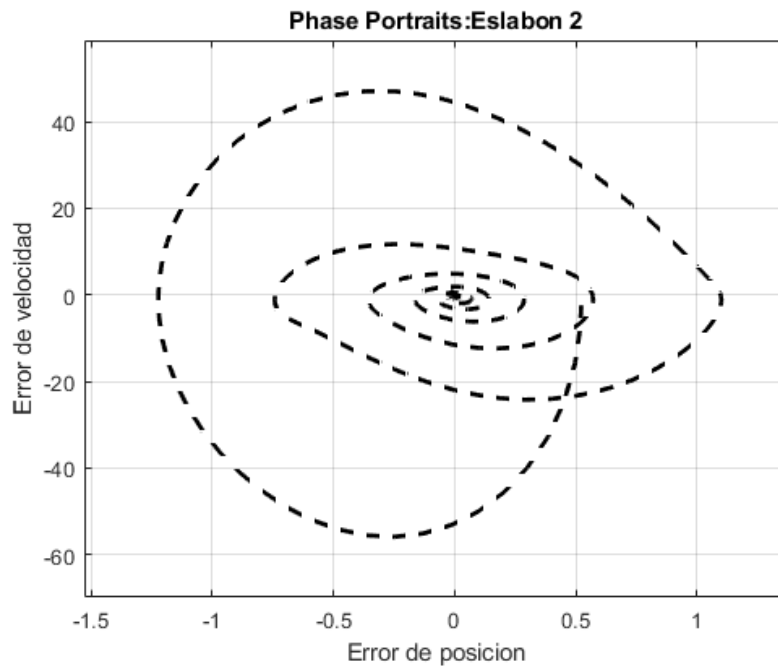


Figura 4.8: Trayectoria articular: infinito-orejona, plano fase articulación 2

Análisis trayectoria articular: infinito-orejona

Con altas ganancias, se logró que la posición actual convergiera a la posición deseada en un tiempo menor a 3 segundos.

En la figura 4.16 podemos observar que el algoritmo para perturbaciones está funcionando y que la señal de la perturbación estimada converge a la real. Cabe señalar que el momento inicial (torque, figura 4.16), es relativamente muy grande a comparación del esfuerzo requerido

para cuando ya está estable el sistema. Tal esfuerzo se ve reflejado en la velocidad con la que sintoniza y que se puede observar en las gráficas de velocidad de ambas articulaciones del robot (figura 4.15)

4.1.3. Trayectoria cartesiana: corazón

Trayectoria cartesiana: corazón, caso global

Ecuación paramétrica	
Centro de la ecuación	$x_c = -.05 ; y_c=0.05;$
Escala	$s=0.03;$
x	$x=s(16\sin(T)^3)+x_c;$
y	$y=s(13\cos(T)-5\cos(2T)-2\cos(3T)-\cos(4T)) + y_c;$

Tabla 4.5: Parámetros de trayectoria: corazón.

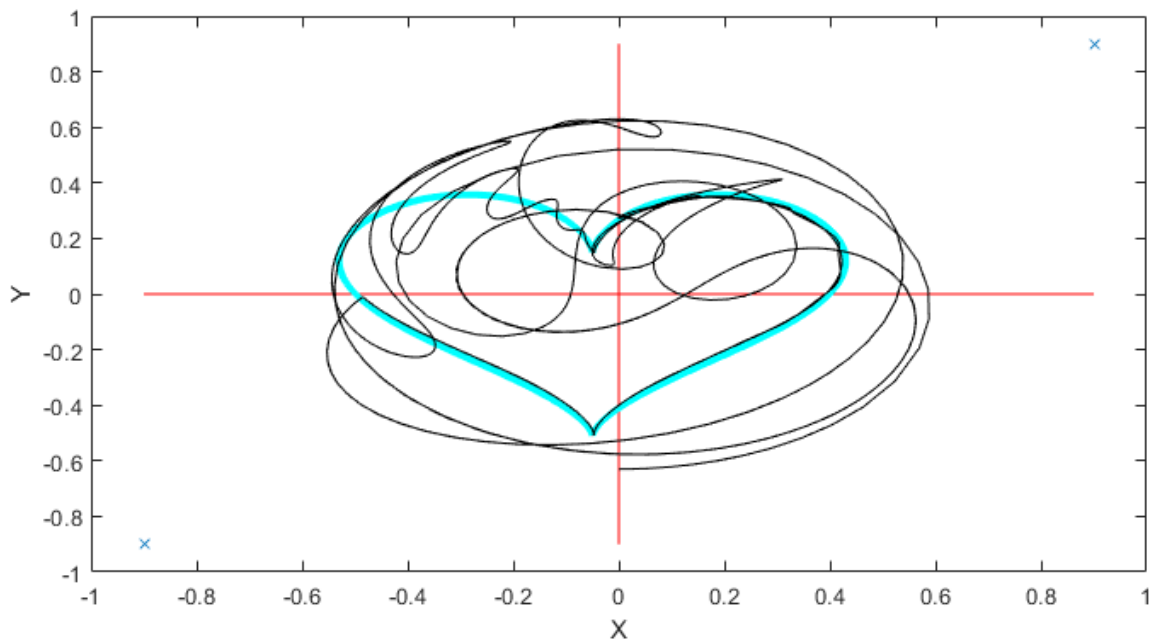


Figura 4.9: Trayectoria cartesiana: Corazón caso global

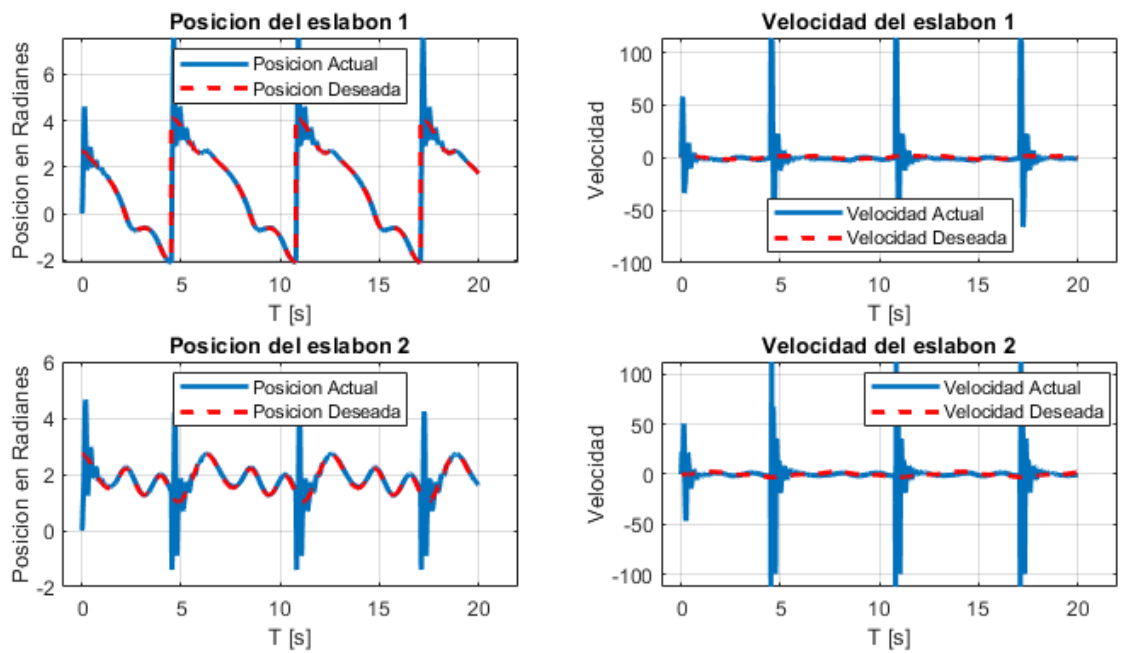


Figura 4.10: Trayectoria cartesiana: Corazón, caso global, posición y velocidad

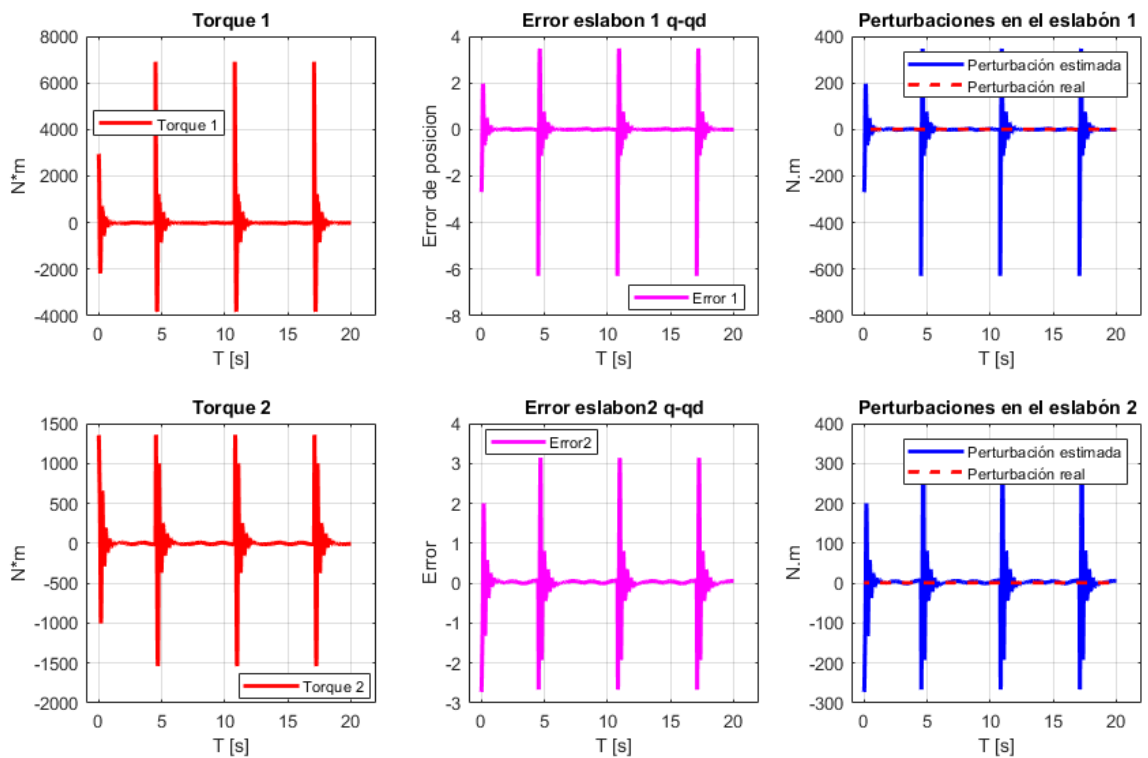


Figura 4.11: Trayectoria cartesiana: Corazón, caso global, torque, error y perturbacion

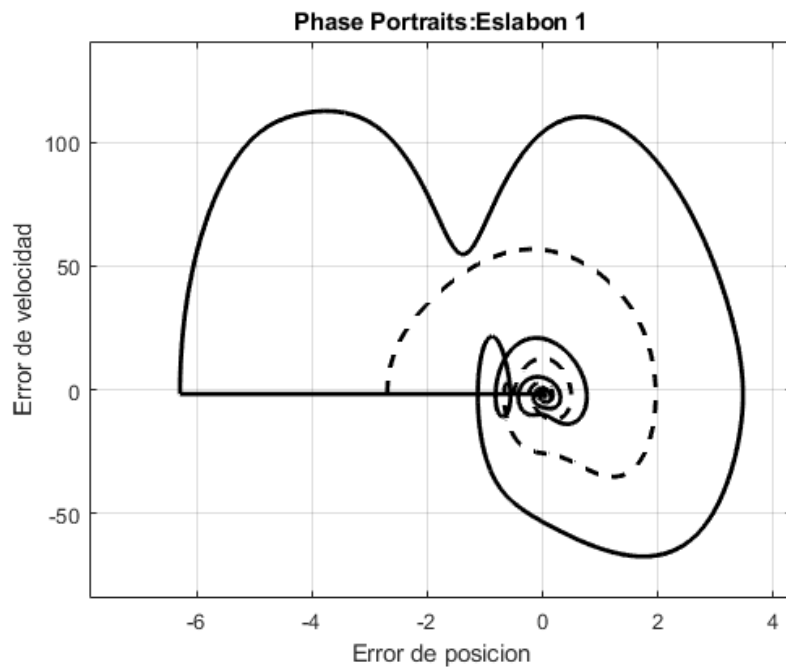


Figura 4.12: Trayectoria cartesiana: Corazón, caso global, plano fase articulación 1

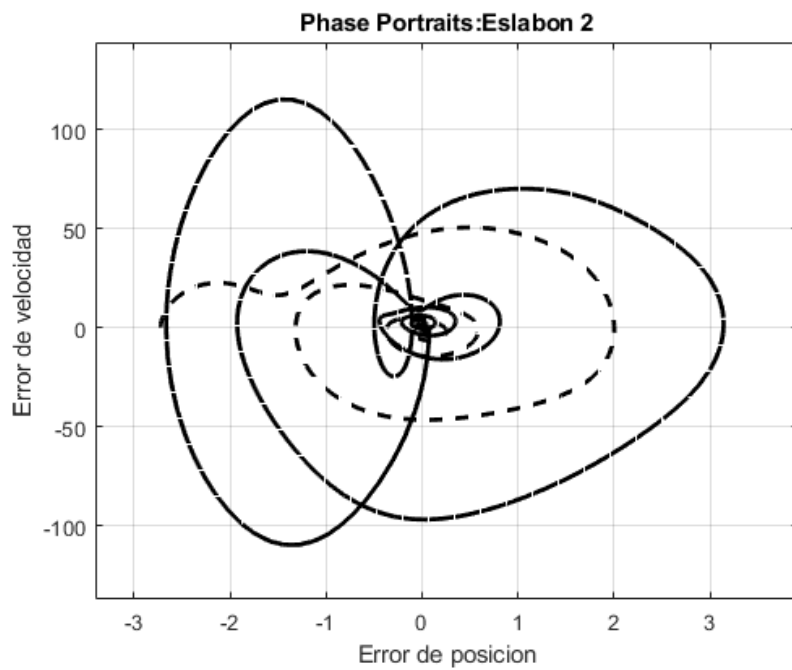


Figura 4.13: Trayectoria cartesiana: Corazón, caso global, plano fase articulación 2

Trayectoria cartesiana: corazón (caso local)

Ecuación paramétrica	
Centro de la ecuación	$x_c = -.2 ; y_c = -0.4;$
Escala	$s = 0.02;$
x	$x = s(16\sin(T)^3) + x_c;$
y	$y = s(13\cos(T) - 5\cos(2T) - 2\cos(3T) - \cos(4T)) + y_c;$

Tabla 4.6: Parámetros de trayectoria: corazón en 4 cuadrante

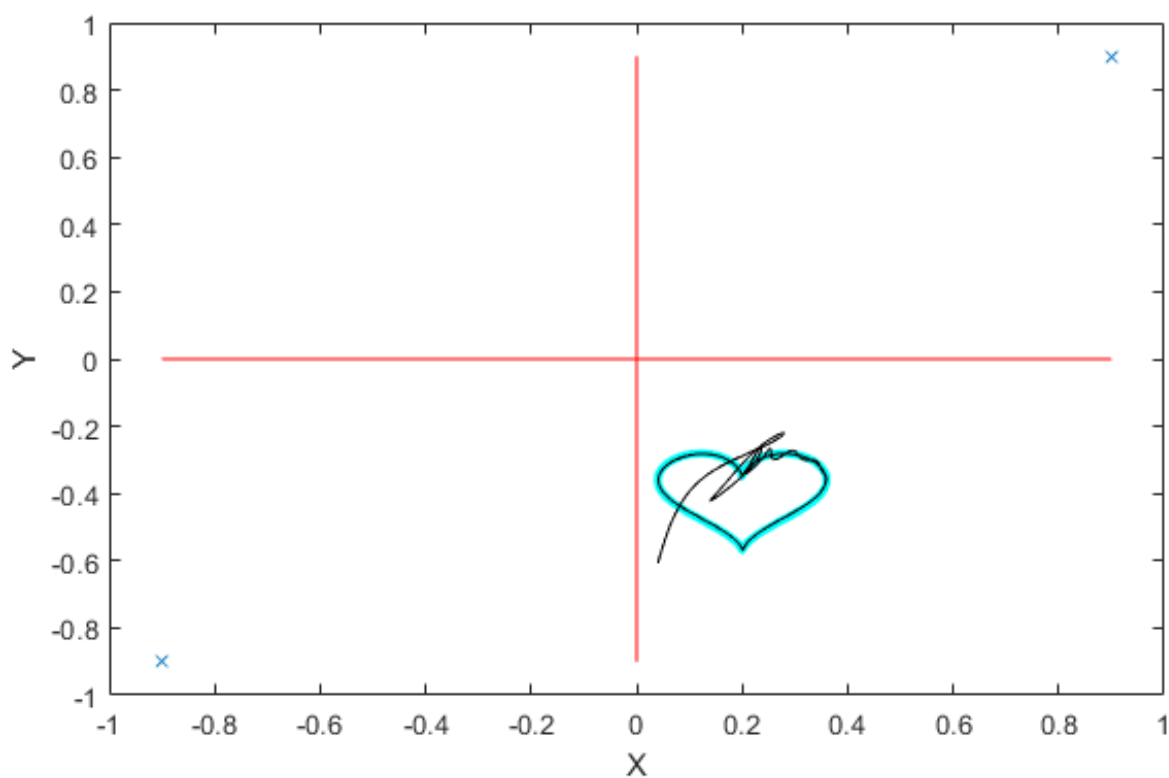


Figura 4.14: Trayectoria cartesiana: Corazón caso local

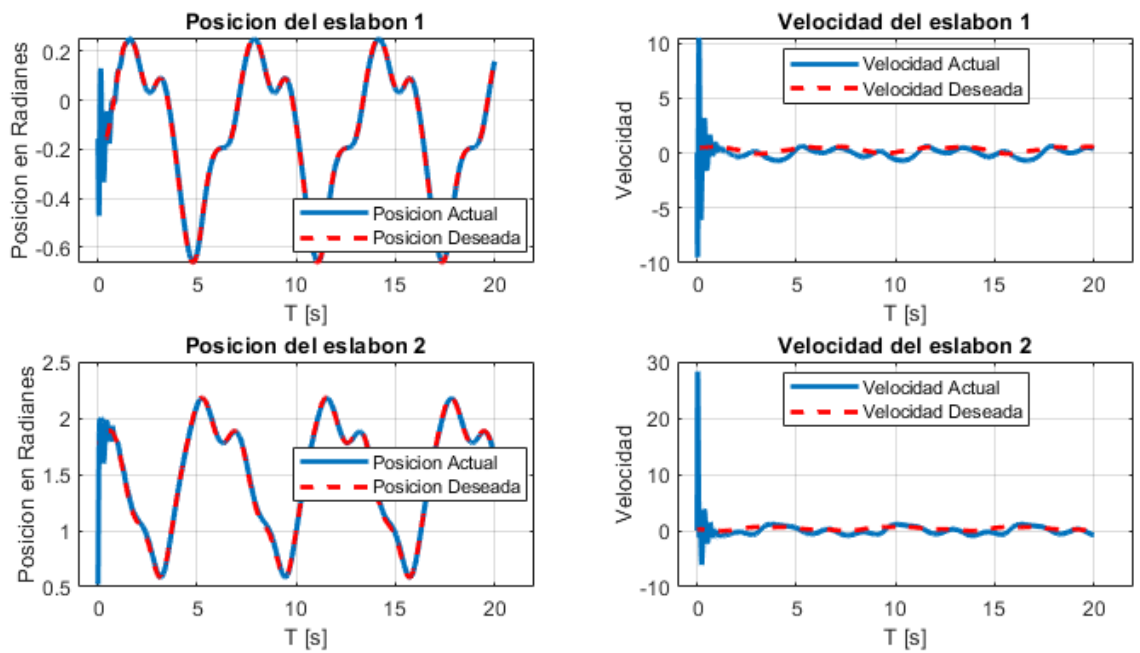


Figura 4.15: Trayectoria cartesiana: Corazón, caso local, posición y velocidad

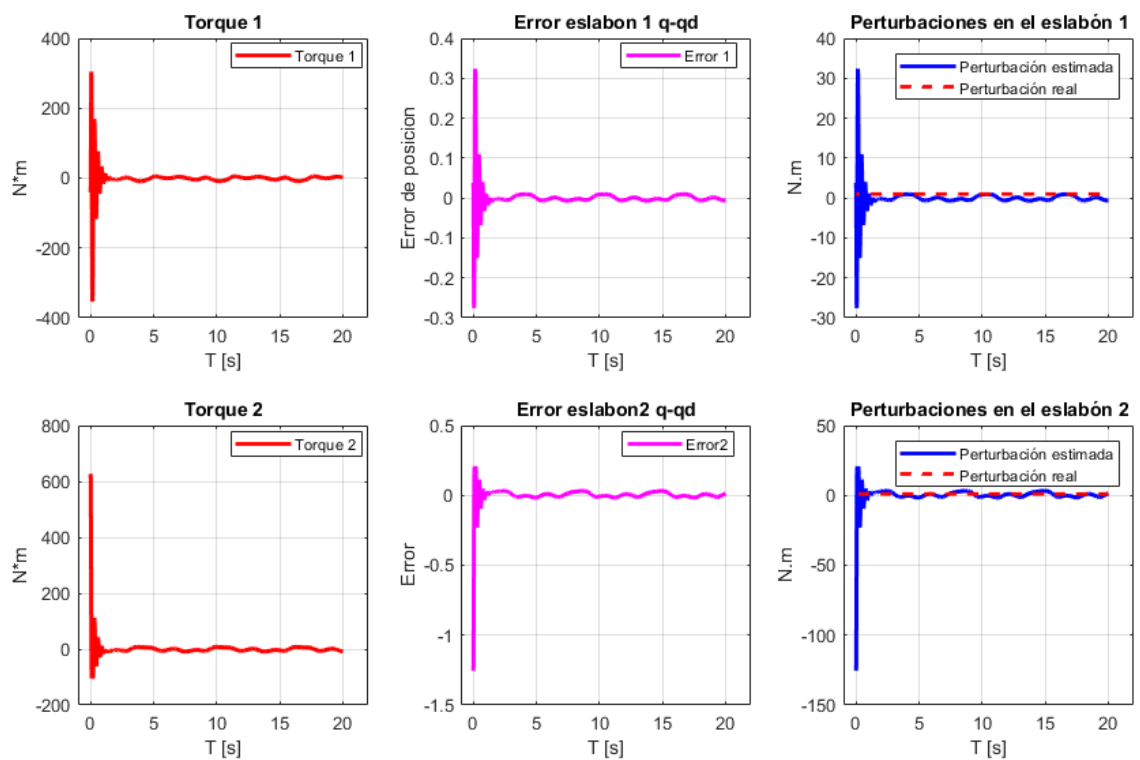


Figura 4.16: Trayectoria cartesiana: Corazón, caso local, torque, error y perturbación

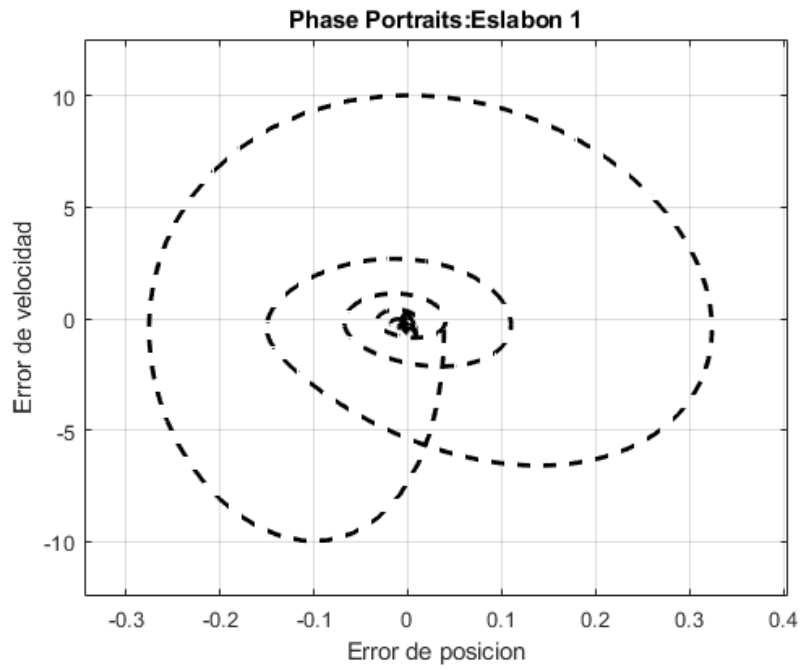


Figura 4.17: Trayectoria cartesiana: Corazón, caso local, plano fase articulación 1

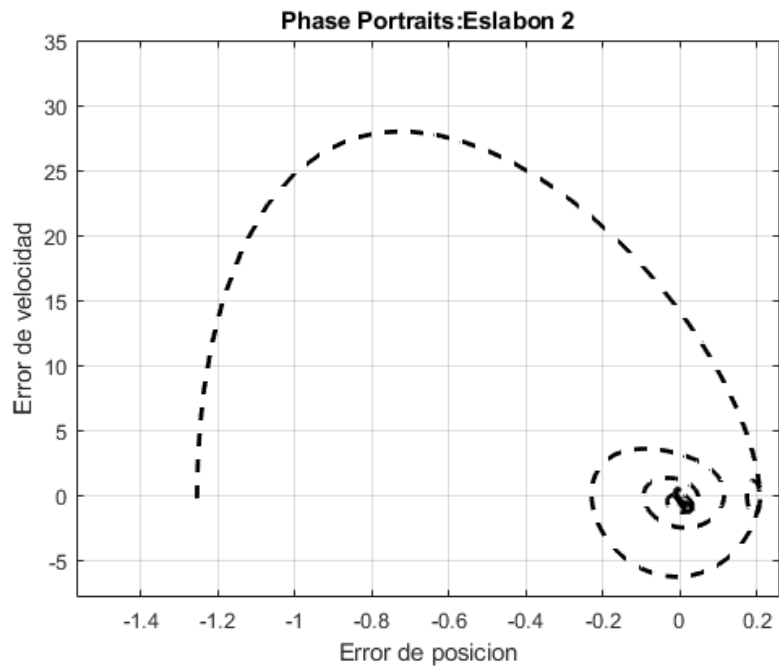
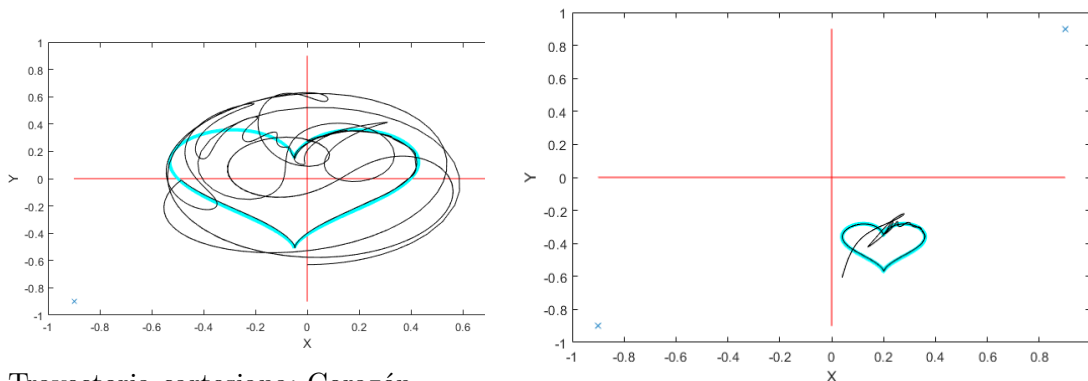


Figura 4.18: Trayectoria cartesiana: Corazón, caso local, plano fase articulación 2

4.1.4. Análisis trayectorias cartesianas: corazón



(a) Trayectoria cartesiana: Corazón caso global.

(b) Trayectoria cartesiana: Corazón caso local.

Figura 4.19: Comparación de trayectorias cartesianas: corazón.

Como se puede observar, en la tabla 4.5 y la tabla 4.6, la ecuación paramétrica que dicta la forma de la figura es la misma. Para el caso local, se cambió el centro de la ecuación al cuarto cuadrante y se escaló, es decir, se hizo más pequeña la figura para que no saliera de dicho cuadrante. Sin embargo, el caso global (figura 4.19a) presenta un movimiento que le causa inestabilidad, cuando se encuentra en el segundo cuadrante. Esto es debido a que entra en una singularidad, la cual matemáticamente se presenta cuando se hace la transformación de cinemática directa a cinemática inversa.

4.1.5. Trayectoria cartesiana: Hipocicloide

Trayectoria cartesiana: Hipocicloide, caso global

Ecuación paramétrica	
Variables	$r_1=0.42; r_2=.07636; k=r_1 / r_2;$
Escala	$s=1;$
x	$x = (r_1-r_2)\cos(T) + r_2\cos(T(1-(k)));$
y	$y = (r_1-r_2)\sin(T) + r_2\sin(T(1-(k)));$

Tabla 4.7: Parámetros de trayectoria: hipocicloide

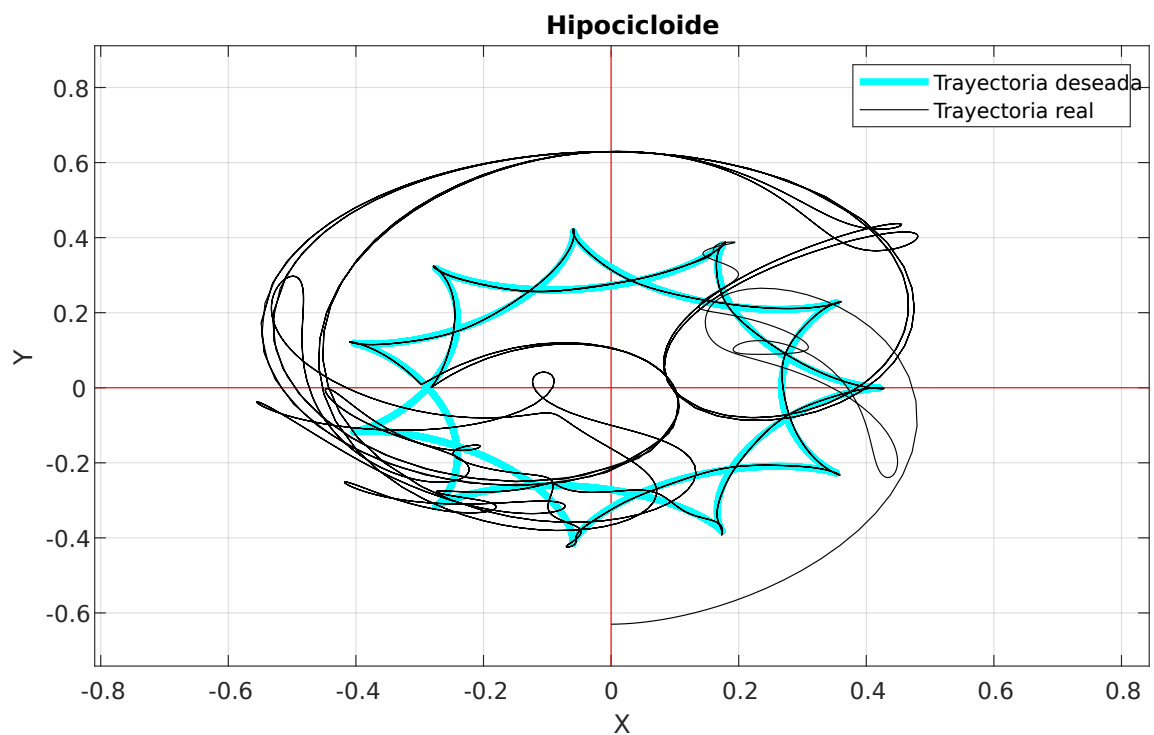


Figura 4.20: Trayectoria cartesiana: Hipocicloide.

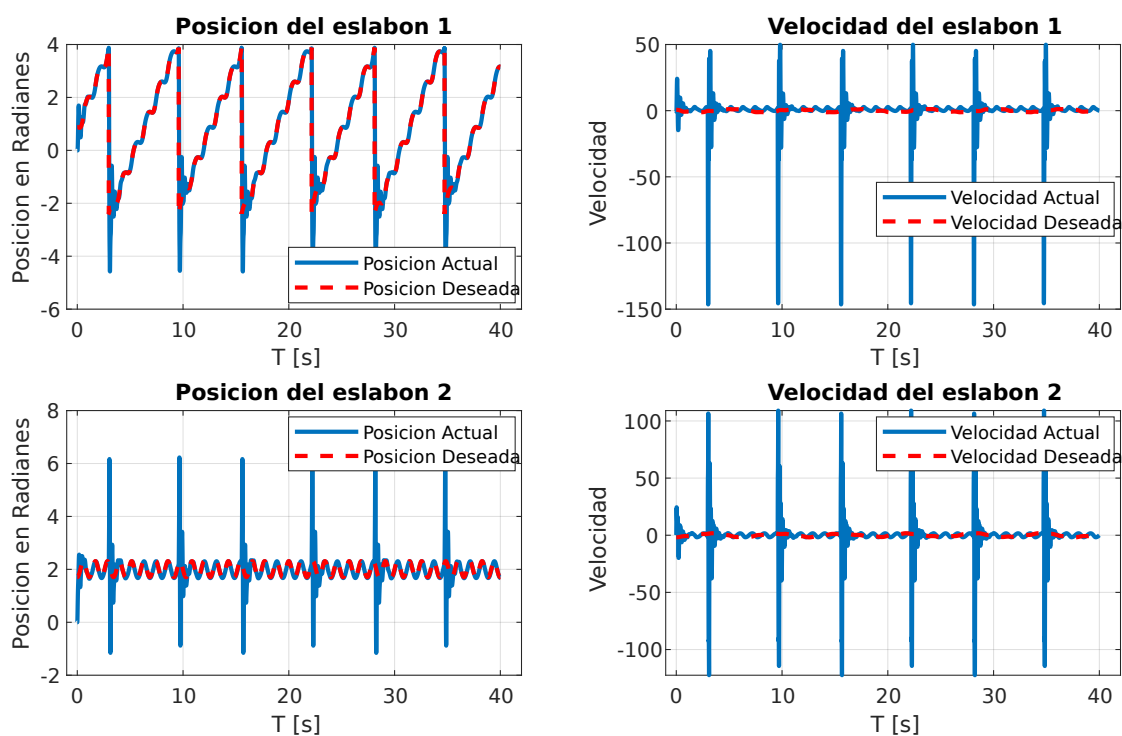


Figura 4.21: Trayectoria cartesiana: Hipocicloide, posiciones y velocidades

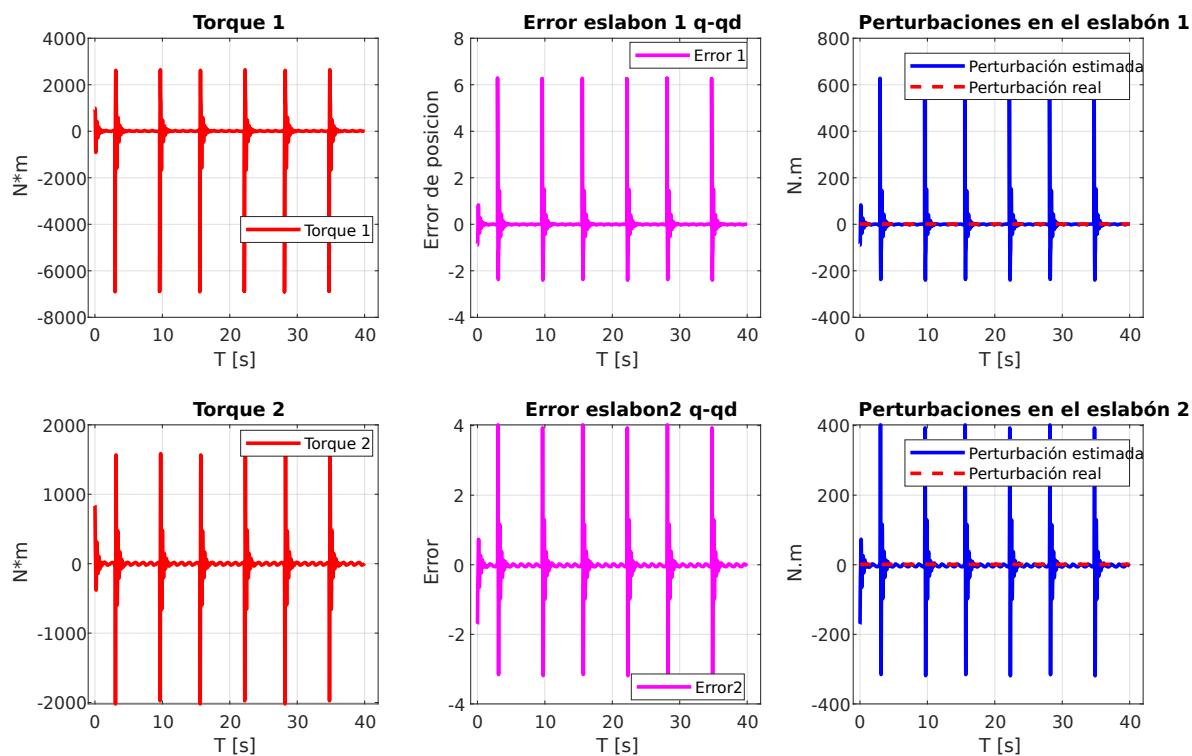


Figura 4.22: Trayectoria cartesiana: Hipocicloide, torque, errores y perturbaciones

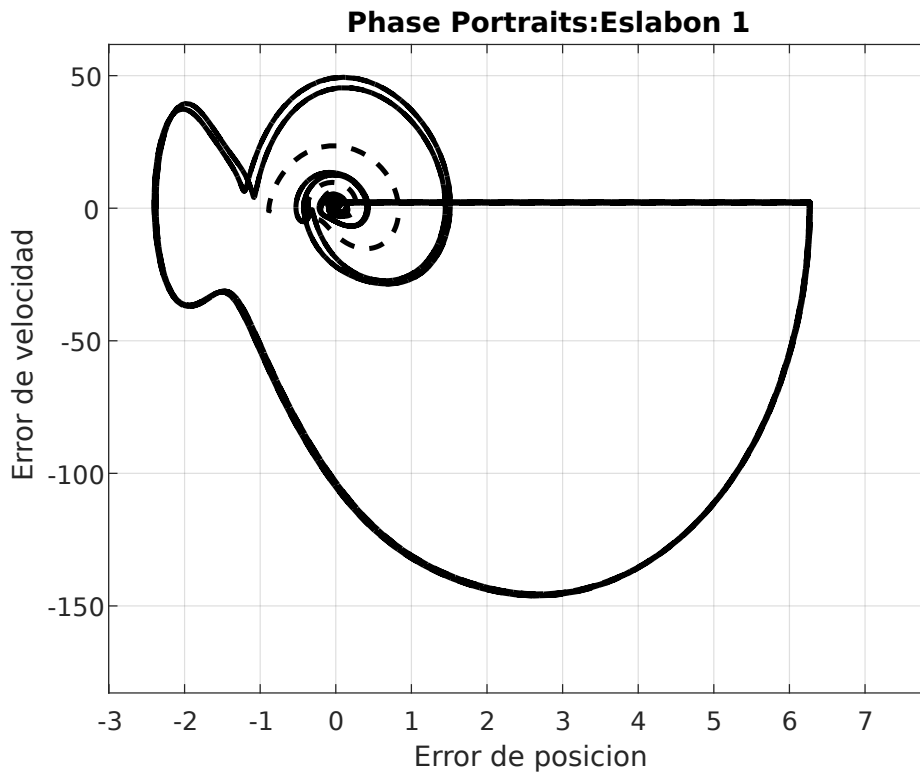


Figura 4.23: Trayectoria cartesiana: Hipocicloide, plano fase articulación 1

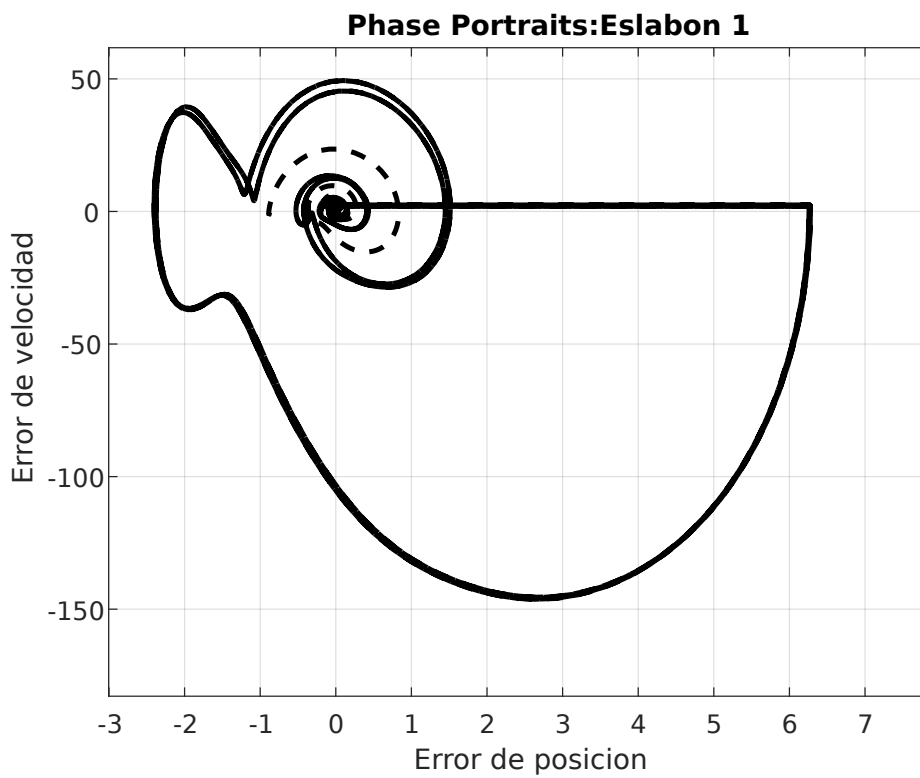


Figura 4.24: Trayectoria cartesiana: Hipocicloide, plano fase articulación 2

Trayectoria cartesiana: Hipocicloide (caso local)

Ecuación paramétrica	
Variables	$r_1=0.42; r_2=.07636; k=r_1 / r_2; xc=0.3; yc=-0.25;$
Escala	$s=0.5;$
x	$x=s((r_1-r_2)\cos(T)+r_2\cos(T(1-k)))+ xc;$
y	$y = s((r_1-r_2)\sin(T)+r_2\sin(T*(1-k)))+ yc;$

Tabla 4.8: Parámetros de trayectoria: hipocicloide cuarto cuadrante.

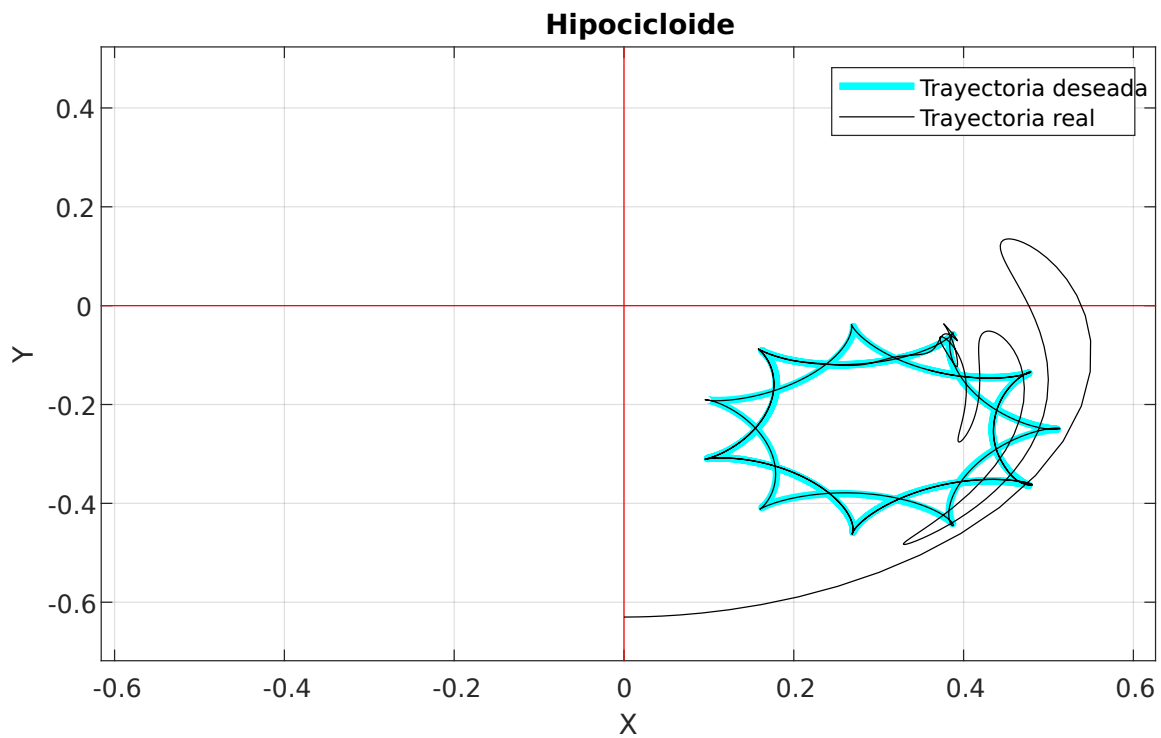


Figura 4.25: Trayectoria cartesiana: Hipocicloide caso local.

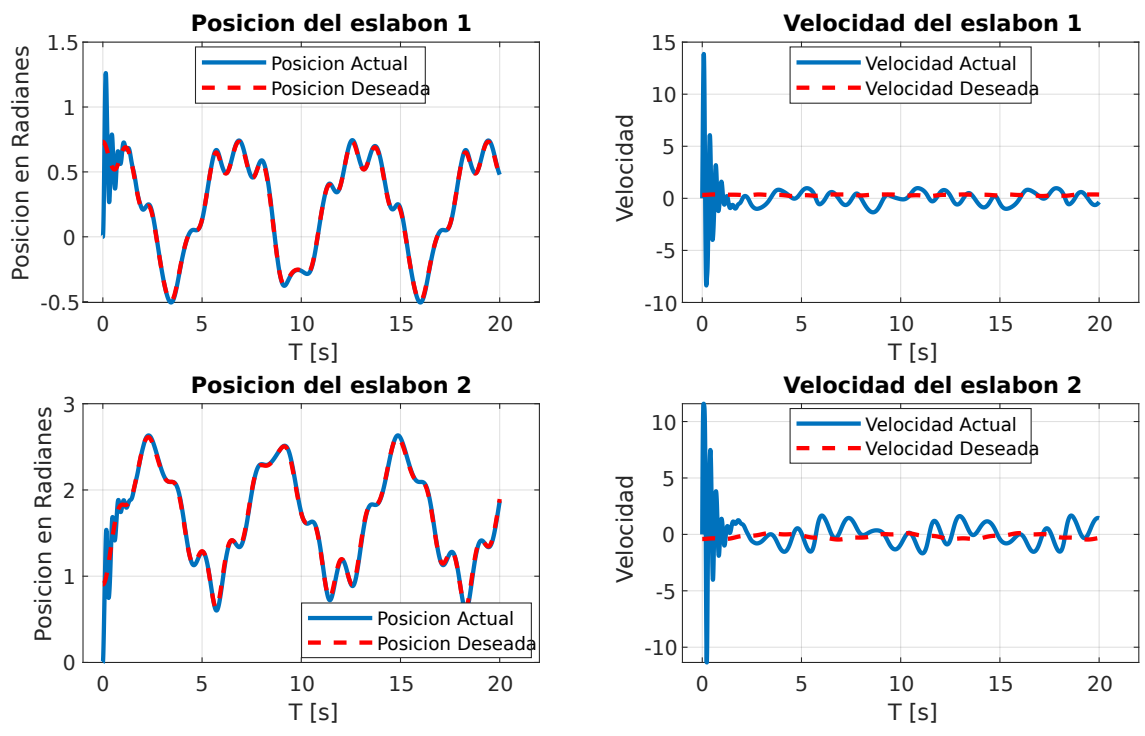


Figura 4.26: Trayectoria cartesiana: Hipocicloide, caso local, posiciones y velocidades.

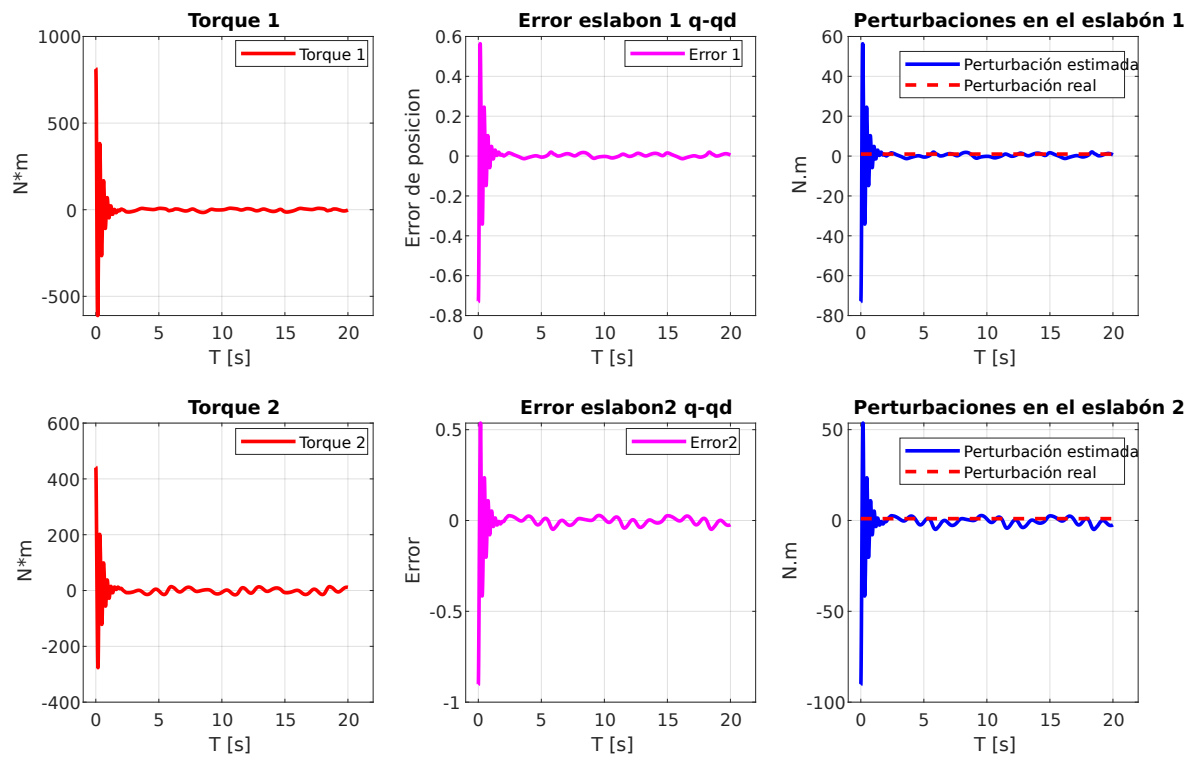


Figura 4.27: Trayectoria cartesiana: Hipocicloide, caso local, torques, errores y perturbaciones.

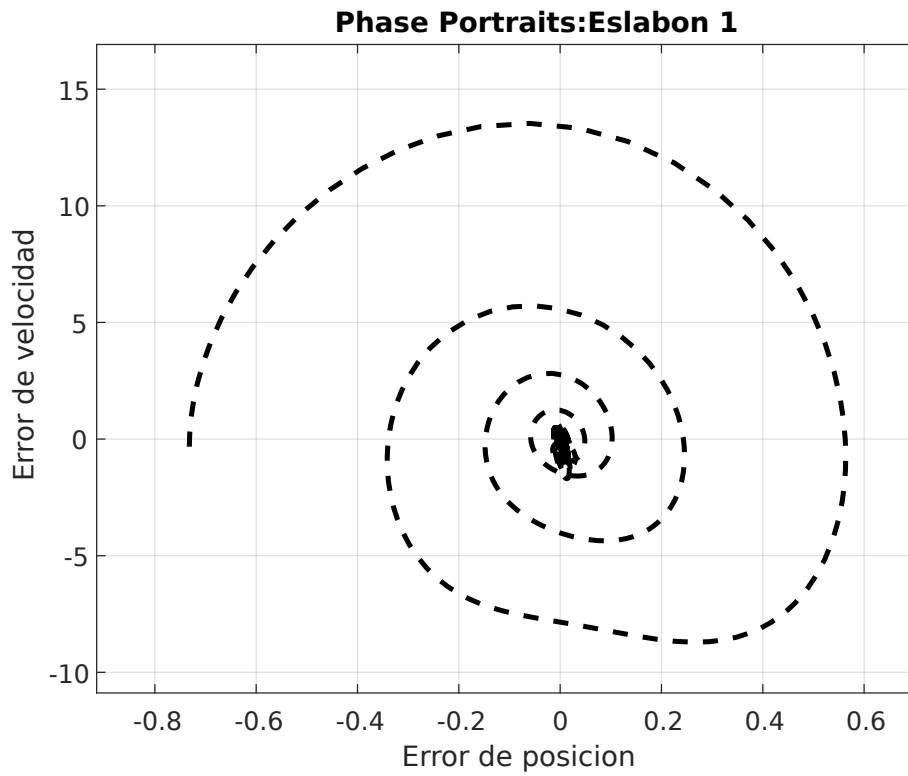


Figura 4.28: Trayectoria cartesiana: Hipocicloide, caso local, retrato de fase articulación 1.

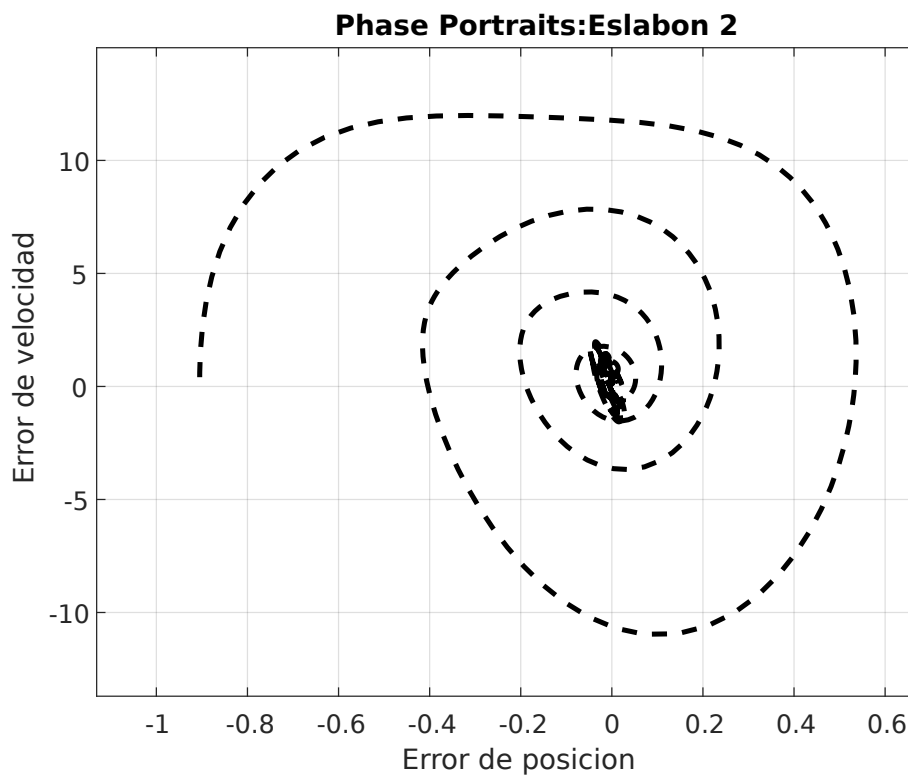
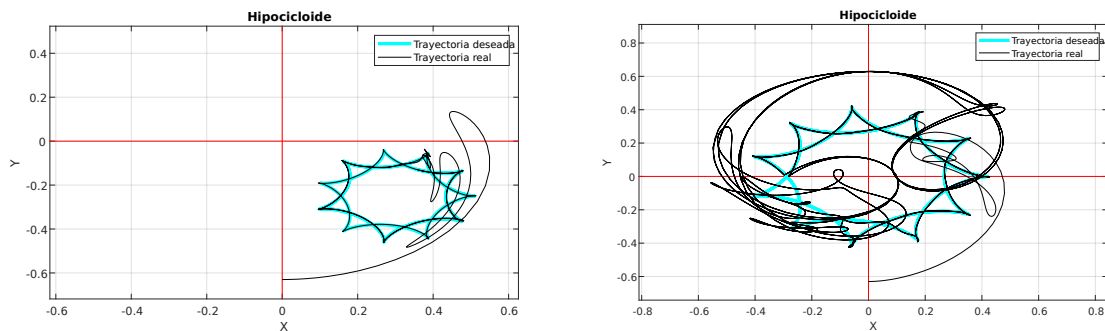


Figura 4.29: Trayectoria cartesiana: Hipocicloide, caso local, retrato de fase articulación 2.

4.1.6. Análisis trayectorias cartesianas: Hipocicloides



(a) Trayectoria cartesiana: Hipocicloide caso local.

(b) Trayectoria cartesiana: Hipocicloide caso global.

Figura 4.30: Comparación de trayectorias cartesianas: Hipocicloides

Como se puede visualizar en la figura (4.30b), cuando la trayectoria pasa del cuadrante II al cuadrante III, el robot se descontrola, esto se debe a la cinemática inversa, hay una singularidad. Para la zona en donde sí alcanza a sintonizar (figura 4.21), se observa que hay un seguimiento de trayectoria.

Al igual que con las trayectorias cartesianas con forma de corazón, se puede observar, en la tabla 4.8 y la tabla 4.7, la ecuación paramétrica que dicta la forma de la figura es la misma. Para el caso local, se cambió el centro de la ecuación al cuarto cuadrante y se escaló.

4.1.7. Trayectoria cartesiana: Figuras de Lissajous

Ecuación paramétrica	
Centro de la ecuación	$x_c = -0.05 ; y_c = -0.3;$
Escala	$s=0.5;$
x	$x = s(0.4\sin(1T)) + x_c;$
y	$y = s(0.3\sin(4T)) + y_c;$

Tabla 4.9: Parámetros de trayectoria: Figura de Lissajous.

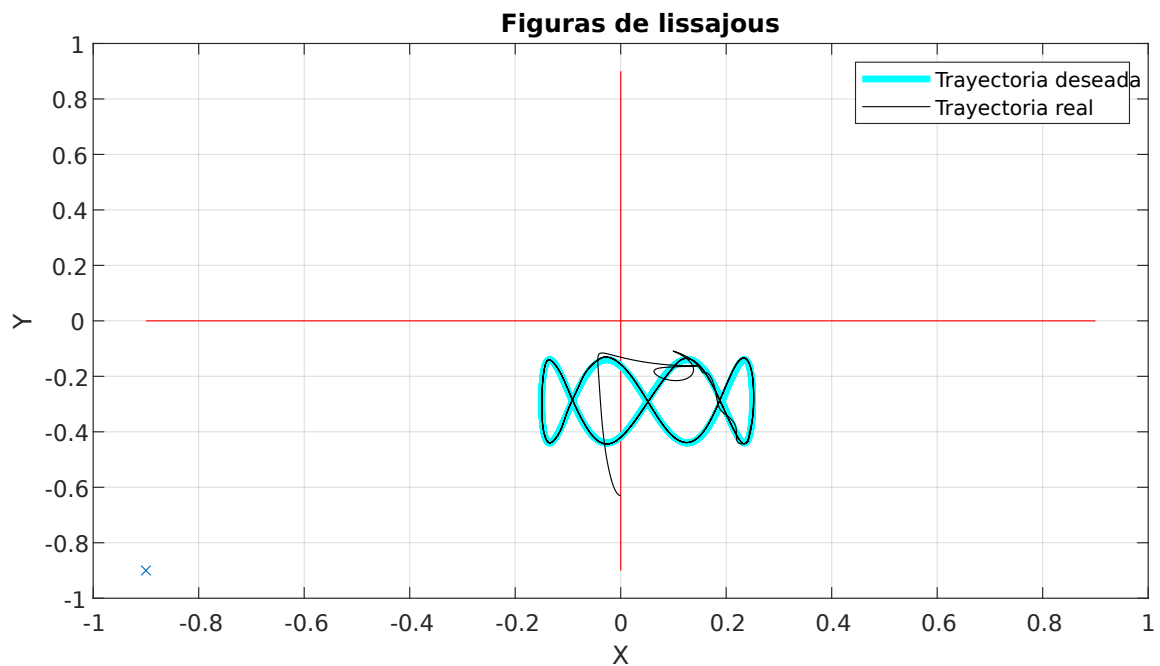


Figura 4.31: Trayectoria cartesiana: Figura de Lissajous

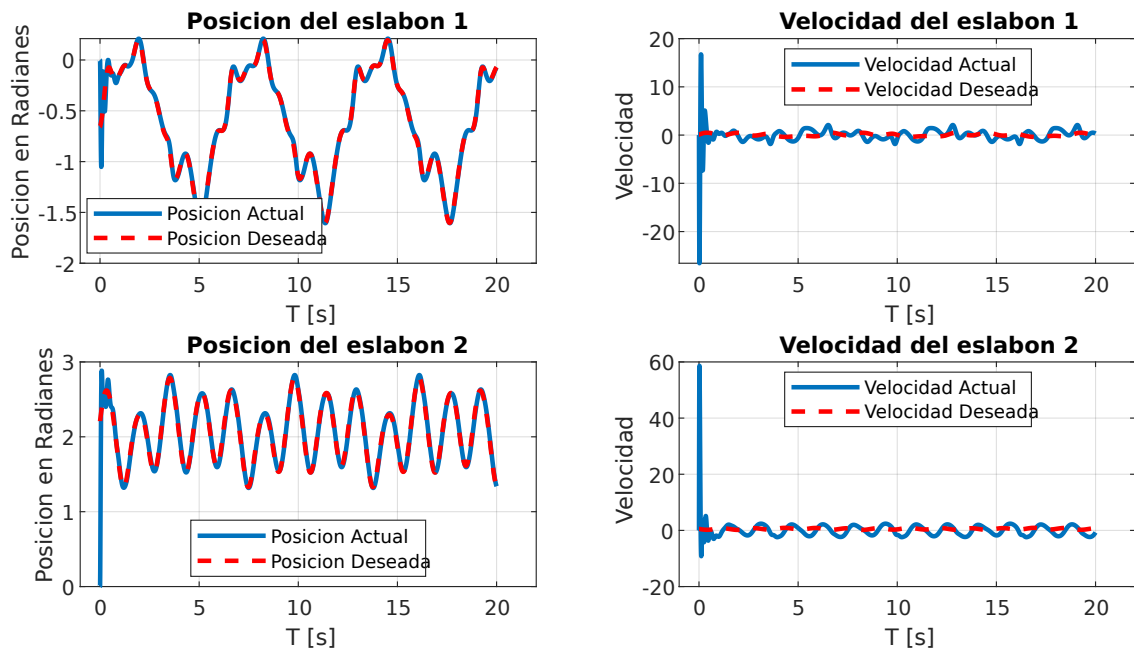


Figura 4.32: Trayectoria cartesiana: Figura de Lissajous, posición y velocidad

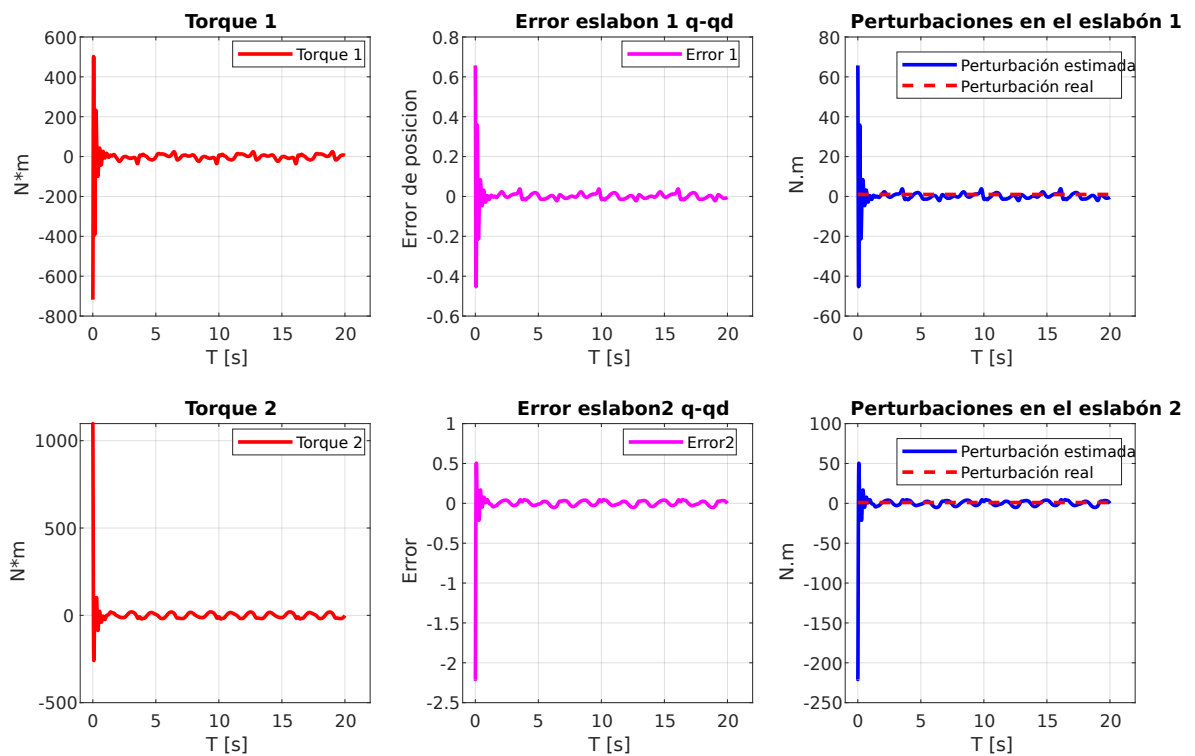


Figura 4.33: Trayectoria cartesiana: Figura de Lissajous, torque, errores y perturbaciones

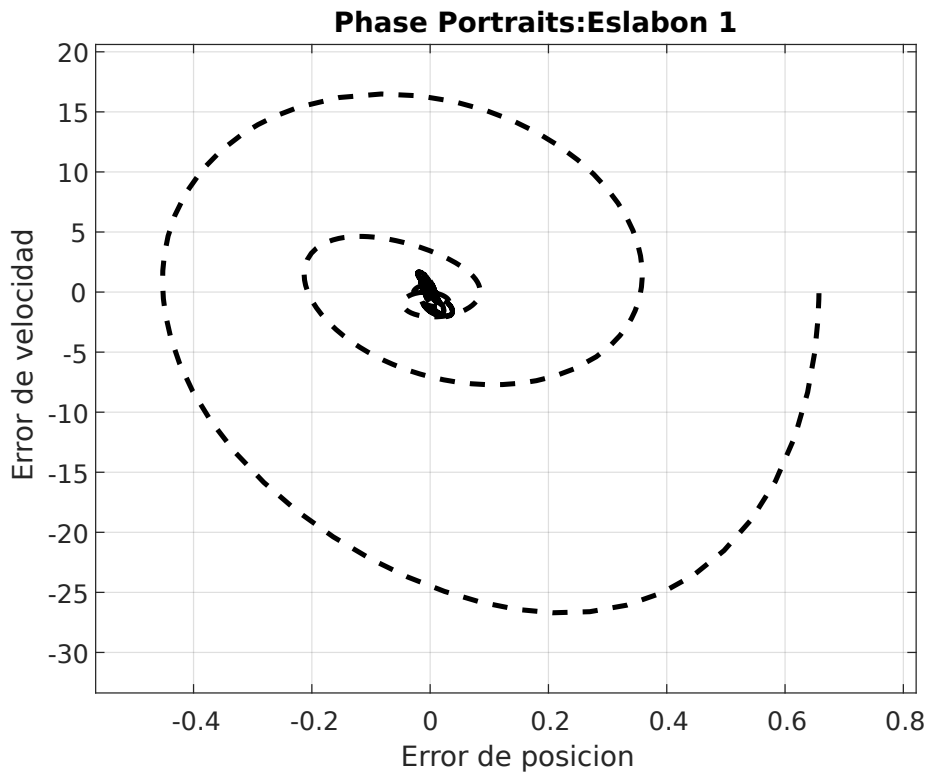


Figura 4.34: Trayectoria cartesiana: Figura de Lissajous, plano fase articulación 2

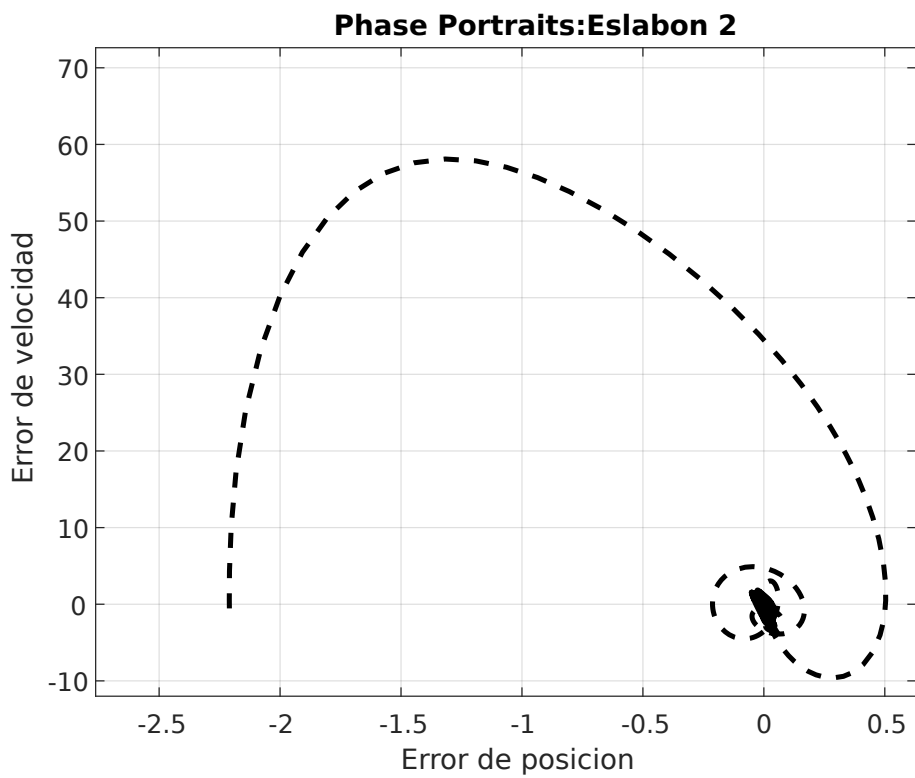


Figura 4.35: Trayectoria cartesiana: Figura de Lissajous, plano fase articulación 2

4.1.8. Experimento físico

Se realizó una etapa de experimentación en laboratorio de la Universidad Autónoma de Baja California, en un robot tipo SCARA. Por motivos de seguridad interna, las ganancias k_p fueron sintonizadas de manera gradual cada 6 segundos con los siguientes valores: [20, 100, 500]. Se eligió la trayectoria cartesiana en forma de corazón, tal como se puede visualizar en la figura (4.36), cabe recordar que dicha figura es como se visualiza en espacio cartesiano, mientras que la figura (4.37) corresponde a la posición angular de las dos articulaciones del robot SCARA, trayectoria deseada y trayectoria medida, y es como se visualiza en espacio articular.

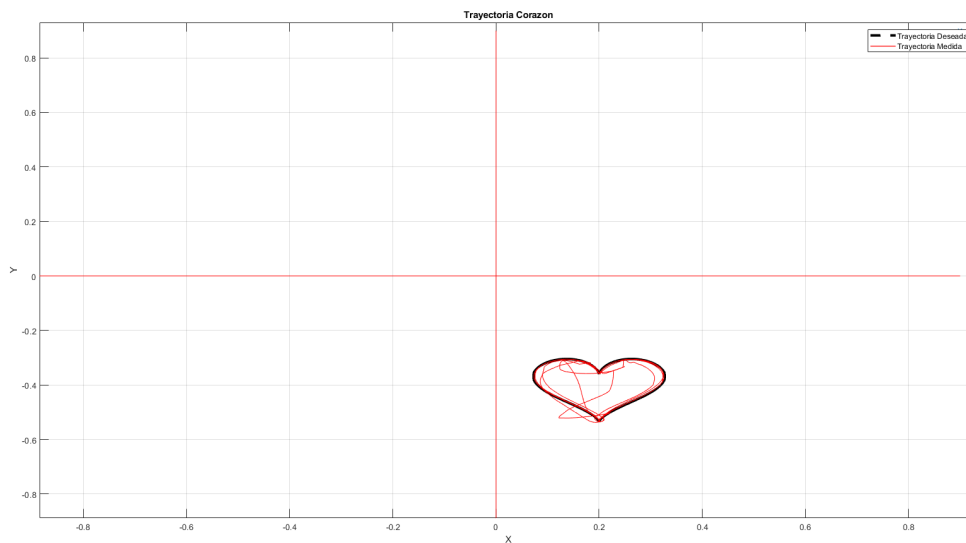


Figura 4.36: Experimento en robot SCARA

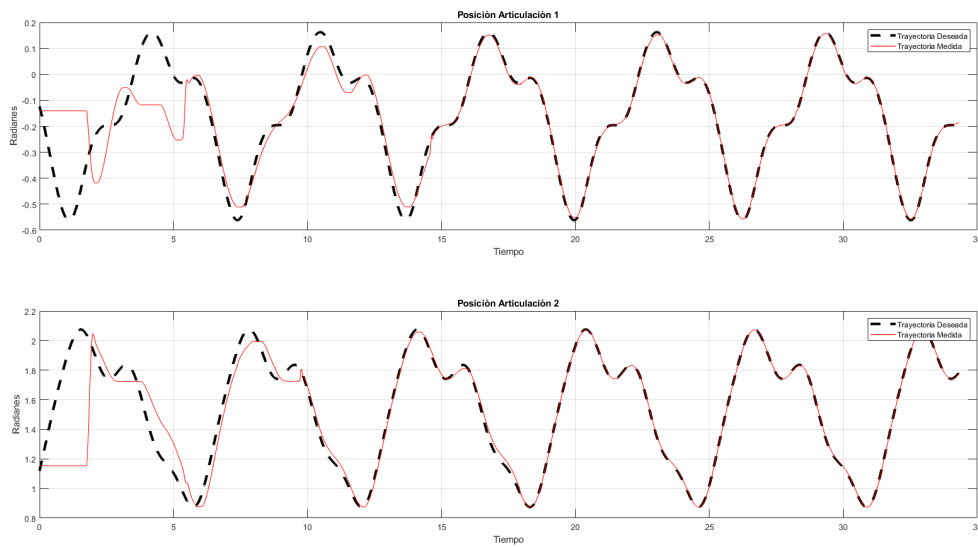


Figura 4.37: Gráfica de posición

Como se puede apreciar en ambas figuras (fig. 4.36 y fig. 4.37) hay un movimiento abrupto al inicio, esto se debe a lo que se explicó anteriormente, la sintonización del controlador no se

hace de manera directa por motivos de seguridad al usuario. Para esto la posición inicial de la segunda articulación este lo más cerca posible de la trayectoria deseada.

Se puede visualizar que las trayectorias convergen asintóticamente, cabe resaltar que el controlador funciona con altas ganancias.

Capítulo 5 | Conclusiones

En esta tesis se abordó el problema de control de seguimiento de trayectoria para un robot con dos ejes rotacionales, donde, las trayectorias son en espacio cartesiano y espacio articular.

El control propuesto “P+” está basado en una ganancia proporcional más una pre-compensación que servirá como un algoritmo estimador de perturbaciones. La principal ventaja de este control de seguimiento de trayectoria es que solo se necesitan mediciones de posición para que $\tilde{q} = q - q_d$ sea cero y $\dot{\tilde{q}} = \dot{q} - \dot{q}_d$ llega a cero de manera asintótica cuando el tiempo tiende a infinito, en el sistema de lazo cerrado; la implementación de observadores, filtros o diferenciadores es innecesario. Esta ventaja se logra usando un control estático “P” simple, por lo tanto, la señal de control es continua, los parámetros de la planta deben ser conocidos.

Teóricamente, la señal de control converge a la señal de referencia de forma asintótica. Se logró estabilidad asintótica global para el caso de seguimiento de trayectoria en espacio articular. Hay estabilidad asintótica local para cuando la trayectoria se formula en espacio cartesiano.

El desarrollo de la metodología de simulación y animación permite al usuario una mejor visualización de la implementación de los controladores.

Bibliografía

- [1] A. Andreev y O. Peregudova, «On Global Trajectory Tracking Control of Robot Manipulators in Cylindrical Phase Space,» *International Journal of Control*, vol. 93, págs. 1-27, 2019. DOI: 10.1080/00207179.2019.1575526.
- [2] —, «Trajectory tracking control for robot manipulators using only position measurements,» *International Journal of Control*, vol. 92, n.º 7, págs. 1490-1496, 2019, ISSN: 13665820. DOI: 10.1080/00207179.2017.1397755. dirección: <https://doi.org/10.1080/00207179.2017.1397755>.
- [3] Can Huang, Fangxing Li y Zhiqiang Jin, «Maximum power point tracking strategy for large-scale wind generation systems considering wind turbine dynamics,» en *2015 IEEE Power Energy Society General Meeting*, 2015, pág. 1. DOI: 10.1109/PESGM.2015.7286039.
- [4] G. C. Lazaroiu, M. Longo, M. Roscia y M. Pagano, «Comparative analysis of fixed and sun tracking low power PV systems considering energy consumption,» *Energy Conversion and Management*, vol. 92, págs. 143-148, 2015, ISSN: 01968904. DOI: 10.1016/j.enconman.2014.12.046. dirección: <http://dx.doi.org/10.1016/j.enconman.2014.12.046>.
- [5] Q. Zhou, H. Li y P. Shi, «Decentralized adaptive fuzzy tracking control for robot finger dynamics,» *IEEE Transactions on Fuzzy Systems*, vol. 23, n.º 3, págs. 501-510, 2015, ISSN: 10636706. DOI: 10.1109/TFUZZ.2014.2315661.
- [6] Z. Tian, Q.-C. Zhong, B. Ren y J. Yuan, «UDE-based robust control for systems with mismatched uncertainties via feedback compensation,» *International Journal of Control*, vol. 0, n.º 0, págs. 1-11, 2019. DOI: 10.1080/00207179.2019.1669826. dirección: <https://doi.org/10.1080/00207179.2019.1669826>.
- [7] Y. Yan, J. Yang, Z. Sun, S. Li y H. Yu, «Non-linear-disturbance-observer-enhanced MPC for motion control systems with multiple disturbances,» *IET Control Theory and Applications*, vol. 14, n.º 1, págs. 63-72, 2020, ISSN: 17518652. DOI: 10.1049/iet-cta.2018.5821.
- [8] V. I. Zubov, A. M. Lyapunov y U. S. A. E. Commission, *Methods of A. M. Lyapunov and Their Application*. P. Noordhoff, 1964, pág. 263. DOI: <https://doi.org/10.1137/1007127>. dirección: <https://books.google.com.mx/books?id=QBwIAQAAIAAJ>.
- [9] D. Shevitz y B. Paden, «Lyapunov stability theory of nonsmooth systems,» en *Proceedings of 32nd IEEE Conference on Decision and Control*, 1993, 416-421 vol.1. DOI: 10.1109/CDC.1993.325114.
- [10] M. S. Branicky, «Multiple Lyapunov functions and other analysis tools for switched and hybrid systems,» *IEEE Transactions on Automatic Control*, vol. 43, n.º 4, págs. 475-482, 1998, ISSN: 00189286. DOI: 10.1109/9.664150.
- [11] *Cleaning Robot Market by Product , Application | COVID-19 Impact Analysis | MarketsandMarkets™*.

- [12] A. Cuendis-Velázquez, M. A. López y López, O. Bada-Yllan, J. M. Zozaya-García, J. Melchor-Ruan, I. Vela-Sarmiento y M. Moreno-Portillo, «Pancreatoduodenectomía totalmente robótica. Primer procedimiento en México,» *Revista de Gastroenterología de México*, vol. 85, n.º 1, págs. 100-102, 2020, ISSN: 0375-0906. DOI: <https://doi.org/10.1016/j.rgmx.2019.03.009>.
- [13] J. F. Avila-Tomás, M. A. Mayer-Pujadas y V. J. Quesada-Varela, «La inteligencia artificial y sus aplicaciones en medicina I: introducción antecedentes a la IA y robótica,» *Atención Primaria*, 2020, ISSN: 0212-6567. DOI: <https://doi.org/10.1016/j.aprim.2020.04.013>.
- [14] G. Odekerken-Schröder, C. Mele, T. Russo-Spena, D. Mahr y A. Ruggiero, «Mitigating loneliness with companion robots in the COVID-19 pandemic and beyond: an integrative framework and research agenda,» *Journal of Service Management*, 2020, ISSN: 17575818. DOI: 10.1108/JOSM-05-2020-0148.
- [15] N. G. Hockstein, C. G. Gourin, R. A. Faust y D. J. Terris, «A history of robots: From science fiction to surgical robotics,» *Journal of Robotic Surgery*, vol. 1, n.º 2, págs. 113-118, 2007, ISSN: 18632483. DOI: 10.1007/s11701-007-0021-2.
- [16] W. Jochen, P. P. G., K. W. H., G. Thorsten, L. V. Nhat, P. Stefanie y M. Antje, «Brave new world: service robots in the frontline,» *Journal of Service Management*, vol. 29, n.º 5, págs. 907-931, ene. de 2018, ISSN: 1757-5818. DOI: 10.1108/JOSM-04-2018-0119.
- [17] M. M. van Pinxteren, R. W. Wetzels, J. Rüger, M. Pluymaekers y M. Wetzels, «Trust in humanoid robots: implications for services marketing,» *Journal of Services Marketing*, vol. 33, n.º 4, págs. 507-518, 2019, ISSN: 08876045. DOI: 10.1108/JSM-01-2018-0045.
- [18] A. Barrientos, A. B. Cruz, L. F. Peñiín y C. Balaguer, *Fundamentos de robótica*. McGraw-Hill, 2007, ISBN: 9788448156367. dirección: <https://books.google.com.mx/books?id=ArEMPAAACAAJ>.
- [19] O. K. Bruno Siciliano, *Springer Handbook of Robotics*. Springer, 2016.
- [20] *Control y Robótica*. dirección: http://www.portaleso.com/web_robot_3/robot_indice.html (visitado 14-05-2021).
- [21] L. Sciavicco y B. Siciliano, «Modelling and Control of Robot Manipulators,» *Measurement Science and Technology*, vol. 11, n.º 12, págs. 1828-1829, nov. de 2000. DOI: 10.1088/0957-0233/11/12/709. dirección: <https://doi.org/10.1088/0957-0233/11/12/709>.
- [22] S. K. Saha, *Introducción a la robótica*. McGraw Hill, 2010. dirección: <https://books.google.com.mx/books?id=UAWYAQAACAAJ>.
- [23] K. Ogata, *Ingeniería de Control Moderna*. 2013, 987:897-912, ISBN: 9788578110796. arXiv: arXiv:1011.1669v3.
- [24] R. Miranda Colorado, *Cinemática y Dinámica de Robots Manipuladores*. Alfaomega Grupo Editor, 2016, pág. 363, ISBN: 978-607-622-048-1.
- [25] F. REYES, *Robótica - control de robots manipuladores*. Alfaomega Grupo Editor, 2011, pág. 592, ISBN: 9786077073079. dirección: <https://books.google.com.mx/books?id=cULVDQAAQBAJ>.
- [26] A. Loria, «Global Tracking Control of One Degree of Freedom Euler-Lagrange Systems without Velocity Measurements,» *IFAC Proceedings Volumes*, vol. 29, n.º 1, págs. 2295-2300, 1996, ISSN: 14746670. DOI: 10.1016/S1474-6670(17)58015-5. dirección: [http://dx.doi.org/10.1016/S1474-6670\(17\)58015-5](http://dx.doi.org/10.1016/S1474-6670(17)58015-5).

- [27] S. Islam y P. X. Liu, «Robust control for robot manipulators by using only joint position measurements,» *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, n.º October, págs. 4013-4018, 2009, ISSN: 1062922X. DOI: 10.1109/ICSMC.2009.5346662.
- [28] A. Loria, «Uniform global position feedback tracking control of mechanical systems without friction,» *Proceedings of the American Control Conference*, n.º 1, págs. 5722-5727, 2013, ISSN: 07431619. DOI: 10.1109/acc.2013.6580734.
- [29] —, «Observers are Unnecessary for Output-Feedback Control of Lagrangian Systems,» *IEEE Transactions on Automatic Control*, vol. 61, n.º 4, págs. 905-920, 2016, ISSN: 00189286. DOI: 10.1109/TAC.2015.2446831. arXiv: arXiv:1307.4659.
- [30] A. Gutiérrez-Giles y M. A. Arteaga-Pérez, «GPI based velocity/force observer design for robot manipulators,» *ISA Transactions*, vol. 53, n.º 4, págs. 929-938, 2014, ISSN: 00190578. DOI: 10.1016/j.isatra.2014.03.002. dirección: <http://dx.doi.org/10.1016/j.isatra.2014.03.002>.
- [31] A. Gutiérrez-Giles, M. A. Arteaga-Pérez y H. Sira-Ramírez, «Control de Fuerza de Robots Manipuladores Basado en Observadores Proporcionales Integrales Generalizados,» *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 13, n.º 2, págs. 238-246, 2016, ISSN: 16977920. DOI: 10.1016/j.riai.2016.01.004.
- [32] H. Sira-Ramirez, E. W. Zurita-Bustamante y C. Huang, «Equivalence among flat filters, dirty derivative-based PID controllers, ADRC, and integral reconstructor-based sliding mode control,» *IEEE Transactions on Control Systems Technology*, vol. 28, n.º 5, págs. 1696-1710, 2020, ISSN: 15580865. DOI: 10.1109/TCST.2019.2919822.
- [33] M. Lu, L. Liu y G. Feng, «Adaptive tracking control of uncertain Euler-Lagrange systems subject to external disturbances,» *Automatica*, vol. 104, págs. 207-219, 2019, ISSN: 00051098. DOI: 10.1016/j.automatica.2019.02.048. dirección: <https://doi.org/10.1016/j.automatica.2019.02.048>.
- [34] M. A. Arteaga-Perez, J. Pliego-Jimenez y J. G. Romero, «Experimental Results on the Robust and Adaptive Control of Robot Manipulators without Velocity Measurements,» *IEEE Transactions on Control Systems Technology*, vol. 28, n.º 6, págs. 2770-2773, 2020, ISSN: 15580865. DOI: 10.1109/TCST.2019.2945915.
- [35] R. Rascón y J. Moreno-Valenzuela, «Output feedback controller for trajectory tracking of robot manipulators without velocity measurements nor observers,» *IET Control Theory and Applications*, vol. 14, n.º 14, págs. 1819-1827, 2020, ISSN: 17518652. DOI: 10.1049/iet-cta.2020.0037.
- [36] R. Kelly y V. Santibáñez, *Control de movimiento de robots manipuladores*, ép. Automática robótica. Pearson Educación, 2003, ISBN: 9788420538310. dirección: <https://books.google.com.mx/books?id=LPwMAAAACAAJ>.
- [37] M. D. Weir, J. Hass, F. R. Giordano y R. L. Finney, *Thomas' calculus*. Pearson Addison Wesley, 2005, ISBN: 0321185587.