

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE CIENCIAS DE LA INGENIERÍA, ADMINISTRATIVAS
Y SOCIALES



Diseño y simulación de un controlador no lineal, MEMS y discreto aplicado al robot Niryo One

TESIS

que presenta para obtener el grado de
MAESTRÍA EN INGENIERÍA

EDGAR RUBÉN REYES SÁNCHEZ

DIRECTOR DE TESIS:
Carlos Alberto Chávez Guzmán

CODIRECTOR DE TESIS:
Rigoberto Martínez Clark

TECATE, B.C.

Septiembre del 2022.



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE CIENCIAS DE LA INGENIERÍA, ADMINISTRATIVAS Y SOCIALES

VOTOS APROBATORIOS DE TESIS DE MAESTRÍA

Fecha: 4 de noviembre de 2022

Nombre del autor: Edgar Rubén Reyes Sánchez

Título de la tesis: Diseño y simulación de un controlador no lineal, MEMS y discreto aplicado al robot Niryo One

CERTIFICAMOS QUE ESTA TESIS HA SIDO APROBADA POR LOS MIEMBROS DEL COMITÉ DE TESIS PARA SU DEFENSA EN EXAMEN ORAL.

Favor de regresar esta forma debidamente firmada y acompañada de la tesis del estudiante, para su revisión a la Coordinación del programa y Coordinación de Investigación y Posgrado de la Facultad de Ingeniería.

Grado Académico	Nombre	Firma	Función
Doctor	Carlos Alberto Chávez Guzmán		Presidente
Doctor	Arturo Sinue Ontiveros Zepeda		Secretario
Doctor	Jesús David Avilés Velázquez		Sinodal
Doctor	Rigoberto Martinez Clark		Sinodal
Doctor	Jován Oseas Mérida Rubio		Sinodal
Doctor	Juan Francisco Flores Reséndiz		Sinodal

Este formato deberá entregarse a la coordinación del programa por lo menos dos semanas antes de la fecha de examen elegida, con las firmas correspondientes y acompañado de un ejemplar de la tesis.

Dr. Carlos Alberto Chávez Guzmán
Coordinación de Investigación y Posgrado



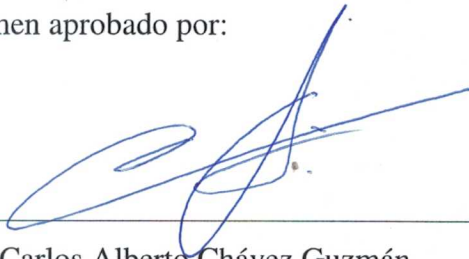
FACULTAD DE CIENCIAS DE LA INGENIERIA ADMINISTRATIVAS Y SOCIALES TECATE, B.C.

Dr. Jesús David Avilés Velázquez
Nombre y firma del Presidente del SACC (programa).

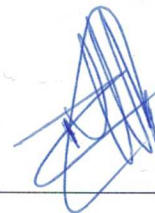
Resumen de la tesis de **Edgar Rubén Reyes Sánchez**, presentada como requisito parcial para la obtención del grado de MAESTRO EN INGENIERÍA del programa de Maestría y Doctorado en Ciencias e Ingeniería (MYDCI) de la UABC. Tecate Baja California, México, agosto del 2022.

Diseño y simulación de un controlador no lineal, MEMS y discreto aplicado al robot Niryo One

Resumen aprobado por:



Dr. Carlos Alberto Chávez Guzmán.
- Director de Tesis.



Dr. Rigoberto Martínez Clark
Codirector de Tesis.

Resumen

En el presente trabajo de tesis se aborda el problema de seguimiento de trayectorias mediante un controlador PD discreto no lineal con compensación de la dinámica del robot sobre la trayectoria deseada, aplicado a un robot manipulador de tres grados de libertad denominado Niryo One, el cual es completamente actuado con articulaciones rígidas y del tipo rotacional, que opera en presencia de perturbaciones acopladas, por ejemplo incertidumbre del modelo derivado de los efectos de los mecanismos de acoplamiento. El análisis y síntesis del controlador se realiza en el dominio del tiempo continuo. Se inicia con el cálculo de la cota mínima que garantiza condiciones suficientes para la existencia y unicidad del punto de equilibrio, posteriormente se calculan las cotas mínimas que garantizan condiciones suficientes para alcanzar estabilidad asintótica en forma global del sistema en lazo cerrado libre de perturbaciones, mediante estabilidad de Lyapunov, derivado de estas condiciones de suficiencia se procede al diseño del controlador. Con apoyo de simulación en MatLab/Simulink se comprueba la estabilidad asintótica y el objetivo de control. La segunda etapa de este trabajo de tesis consiste en discretizar el sistema en lazo cerrado mediante el algoritmo numérico de Euler y verificar su desempeño mediante simulación en MatLab/Simulink y CoppeliaSim.

Palabras Clave: *Control PD discreto no lineal, Robot manipulador, Simulación en CoppeliaSim*

Abstract

In the present thesis, the trajectory tracking problem is addressed by means of a nonlinear discrete PD controller with compensation of the robot dynamics on the desired trajectory, applied to a manipulator robot with three degrees of freedom called Niryo One, which is fully actuated with rigid and rotational type joints, operating in the presence of coupled disturbances, e.g. model uncertainty arising from the effects of coupling mechanisms. The analysis and synthesis of the controller was carried out in the domain of continuous time. It begins with the calculation of the minimum bound that guarantees sufficient conditions for existence and uniqueness of the equilibrium point, then the minimum bound that guarantee sufficient conditions to achieve global asymptotic stability of the disturbance-free closed loop system, through Lyapunov stability; derived from these sufficiency conditions, we proceed to the design of the controller. With simulation support in MatLab/Simulink, the asymptotic stability and the control objective are validated. The second stage of this thesis consists of discretizing the closed-loop system using the Euler numerical method and verifying its performance through simulation in MatLab/Simulink and CoppeliaSim.

Keywords: *Nonlinear Discrete PD Control, Robot Manipulator, Simulation in Coppelia-Sim*

Dedicatoria

El producto de este trabajo está dedicado a mis padres, por haberme inculcado buenos valores y permitirme formarme académicamente.

A mis familiares, amigos y compañeros homólogos, que de alguna manera me apoyaron y motivaron al ingresar a los estudios de posgrado.

A mis hijos por ser el principal motor de mi vida, ya que ustedes fueron una de las razones por las que decidí iniciar esto. Lamento mucho los ratos que no pude estar por dedicarme a terminar alguna tarea.

Y por último pero no menos importante, a mi esposa, por apoyarme de inicio a fin, otorgándome la oportunidad de poder cumplir con las actividades y pendientes, por ser un ejemplo de superación académica, por sus palabras cuando estaba a punto de rendirme y por ser siempre ese complemento importante de mi vida que me mantiene de vuelta en el camino.

Agradecimientos

A Dios por brindarme las oportunidades y los medios para realizar los estudios de posgrado.

A mi familia por ser el motivo de mi superación, por su apoyo y comprensión en momentos de flaqueza.

A la Universidad Autónoma de Baja California, por permitirme el espacio y la posibilidad de ingresar al programa de Maestría.

A mis maestros por su enseñanza, tiempo y dedicación. Su profesionalismo y experiencia han sido fundamentales para la realización de este trabajo.

A mi codirector de tesis, el Dr. Rigoberto Martínez Clark por su enseñanza, guía y contribución con conocimiento en software.

A mi director de tesis, el Dr. Carlos Alberto Chávez Guzmán por su orientación desde mi ingreso hasta el último momento. Por su paciencia y asistencia oportuna en las observaciones de la tesis.

Índice general

Índice de figuras	XV
Índice de tablas	XVII
Lista de símbolos	XIX
1. Planteamiento del problema	1
1.1. Introducción	1
1.2. Estructura de la tesis	3
1.3. Objetivos	4
1.3.1. Objetivo general	4
1.3.2. Objetivos específicos	4
1.4. Hipótesis	4
1.5. Limitaciones	5
2. Fundamentos de la robótica	7
2.1. Historia de la robótica	7
2.2. Fundamentos de robots manipuladores	9
2.2.1. Tipos de articulaciones	11
2.2.2. Grados de libertad	12
3. Modelado dinámico del robot manipulador Niryo One	17
3.1. Modelado dinámico	17
3.1.1. Representación en su forma compacta	23
3.1.2. Cálculo de parámetros	25
3.1.3. Representación en espacios de estados	27
3.1.4. Simulación y resultados del modelo continuo en MatLab/Simulink del robot Niryo One	29

3.2. Discretización del modelo dinámico del robot Niryo One.	30
3.2.1. Discretización del modelo continuo del robot Niryo One	31
3.2.2. Simulación y resultados del modelo discreto en MatLab/Simulink del robot Niryo One	32
4. Análisis y diseño en tiempo continuo del controlador PD no lineal con compen- sación de trayectoria deseada aplicado al robot Niryo One	35
4.1. Objetivo de control	37
4.2. Análisis de existencia y unicidad del punto de equilibrio	37
4.3. Análisis de estabilidad asintótica en forma global	39
4.4. Simulación y resultados del sistema continuo en lazo cerrado	45
5. Discretización del controlador PD no lineal con compensación de trayectoria deseada.	51
5.1. Discretización del sistema en lazo cerrado	51
5.2. Simulación y resultados del sistema en lazo cerrado mediante MatLab/Simulink	52
5.3. Simulación y resultados del sistema en lazo cerrado mediante CoppeliaSim	57
5.4. Resultados de la Tesis	59
6. Conclusiones y trabajo futuro	61
6.1. Conclusiones generales	61
6.2. Trabajo a futuro	62
7. Anexos	63
7.1. Código de Matlab en tiempo continuo	63
7.1.1. Controlador	63
7.1.2. Planta	65
7.1.3. Señal de referencia	67
7.2. Código de Matlab en tiempo discreto	68
7.2.1. Planta	68
7.2.2. Controlador	70
7.2.3. Señal de referencia	71
7.3. Código de Coppelia	72
7.4. Cálculo de parámetros	83
7.4.1. Cálculo de $\lambda_{Max}\{M\}$	83
7.4.2. Cálculo de k_M	83

7.4.3.	Cálculo de k_{C_1}	83
7.4.4.	Cálculo de K_{C_2}	85
7.4.5.	Cálculo de k_g	86
7.4.6.	Cálculo de k_{h_1}	87
7.4.7.	Cálculo de k_{h_2}	87

Bibliografía	89
---------------------	-----------

Índice de figuras

2.1. Gallo del reloj de Estrasburgo	8
2.2. Fuente en forma de pavo real	8
2.3. Máquina de aspecto humano	9
2.4. León mecánico	9
2.5. Configuraciones de robots manipuladores	10
2.6. Robot manipulador del fabricante ABB	11
2.7. Articulaciones y enlaces de un robot manipulador	12
2.8. Articulaciones mecánicas de robots	12
2.9. Brazo robótico de 3 GDL	13
2.10. Brazo robótico de 6 GDL	14
2.11. Brazo robótico de 11 GDL	14
2.12. Articulaciones rotacionales y traslacionales	15
3.1. Robot manipulador Niryo One	18
3.2. Diagrama de cuerpo libre del robot Niryo One con 3 GDL	18
3.3. Articulaciones del robot Niryo One	28
3.4. Diagrama de bloques del modelado del robot	29
3.5. Desplazamiento angular de las articulaciones	30
3.6. Diagrama de bloques de la discretización del modelo dinámico del robot	32
3.7. Respuesta de las articulaciones en modo discreto	33
4.1. Diagrama del controlador PD continuo, aplicado al robot	45
4.2. Diagrama de bloques de funciones en Simulink	47
4.3. Respuesta en lazo cerrado del robot manipulador	47
4.4. Vista ampliada de la respuesta en lazo cerrado del robot manipulador	48
4.5. Plano de fase de la articulación $q_1(t)$	48
4.6. Plano de fase de la articulación $q_2(t)$	49

4.7. Plano de fase de la articulación $q_3(t)$	49
5.1. Representación del control discreto aplicado al robot Niryo One	52
5.2. Diagrama de bloques en modo discreto del robot de 3 GDL	54
5.3. Respuesta de las articulaciones en modo discreto	54
5.4. Respuesta de las articulaciones en modo discreto	55
5.5. Plano de fase de la articulación $q_1(k)$	55
5.6. Plano de fase de la articulación $q_2(k)$	56
5.7. Plano de fase de la articulación $q_3(k)$	56
5.8. Robot usado para simulación en CoppeliaSim	57
5.9. Robot usado para simulación en CoppeliaSim	58
5.10. Respuesta en el tiempo discreto de las articulaciones $q_1(k)$, $q_2(k)$ y $q_3(k)$ en Coppelia	59

Índice de tablas

3.1. Parámetros de robot Niryo One	27
3.2. Magnitudes físicas del robot manipulador Niryo One	28
3.3. Límites mecánicos de las articulaciones	28
5.1. Resultados de la simulación del desempeño del sistema en lazo cerrado . .	60

Lista de símbolos

La siguiente lista muestra los símbolos y acrónimos que serán usados en este documento

$>$	Mayor que
α	Coeficiente de fricción viscosa de las articulaciones
\approx	Aproximado a
\cong	Congruente con
\ddot{x}	Segunda derivada con respecto al tiempo
\dot{x}	Primera derivada con respecto al tiempo
$diag\{\}$	Matriz diagonal
\forall	Para todo valor
\geq	Mayor o igual que
\in	Pertenece a
$\lambda_{max}\{A\}$	Valor propio máximo de la matriz A
$\lambda_{min}\{A\}$	Valor propio mínimo de la matriz A
$ x $	Valor absoluto de x
$\ A\ $	Norma Euclidiana de la matriz A
\mathbb{R}^3	Vector de 3 elementos reales
$\mathbb{R}^{3 \times 3}$	Matriz de 3x3, de números reales

\neq	Diferente de
\bar{q}	Variables de estado
∂	Derivada parcial
τ	Par aplicado a las articulaciones
A^T	Transpuesta de la matriz A
$sech$	Secante hiperbólica
$tanh$	Tangente hiperbólica
GDL	Grados de libertad

Capítulo 1

Planteamiento del problema

1.1. Introducción

La robótica tiene profundas raíces en la historia humana, el origen de la palabra robot data de 1920. A lo largo de los años, ingenieros y científicos han tratado de crear sustitutos capaces de suplantar nuestro comportamiento en muchos campos (artesanía, artes, manufactura, medicina, etcétera), tratando de imitar nuestra interacción con el medio ambiente y con otros agentes y personas [31]. El avance en la robótica ha sido gigantesco, y en la actualidad, existen distintos tipos de robots, como industriales o de servicios, así como definiciones para ellos que son aceptadas por distintos organismos o instituciones, como la Organización Internacional de Estandarización (ISO), que define a los robots industriales como “un manipulador automáticamente controlado, programable y multiusos, programable en tres o más ejes, que pueden ser usados para distintos procesos industriales, ya sean móviles o fijos”. La estructura mecánica de los robots manipuladores consiste en una secuencia de cuerpos rígidos (eslabones) interconectados mediante articulaciones rígidas en una cadena cinemática en serie o abierta cuya movilidad está asegurada gracias a sus articulaciones [14]. El movimiento entre dos eslabones consecutivos se realiza por medio de una articulación prismática o rotacional, por lo que cada articulación proporciona un solo grado de libertad (GDL), las articulaciones empleadas son activas, lo que significa que están equipadas con actuadores.

Respecto al modelado del comportamiento dinámico de los robots manipuladores se han propuesto varios enfoques, como el Lagrangiano [32], el Lagrangiano Recursivo [11] y el Newton-Euler Recursivo [21]. De la estructura de un robot manipulador, son sistemas altamente no lineales y tienen un comportamiento complicado que incluye interacciones entre múltiples articulaciones, efectos no lineales como fuerzas centrífugas y de Coriolis, inercia variable según la configuración del brazo, y a medida que se persigue un mayor rendimiento en términos de velocidad y precisión, estas dinámicas se vuelven más importantes [26], esto llevó a la construcción de robot manipuladores con articulaciones rígidas, para ser operados con altas velocidades y precisión [31], sin embargo los robots manipuladores con articulaciones rígidas al igual que todos los sistemas mecánicos están sujetos

a deformación bajo carga y alta velocidad [35]. Otros factores que afectan el rendimiento de los robots manipuladores son los efectos inherentes a los sistemas mecánicos como el efecto de elasticidad en el acoplamiento mecánico entre el actuador y la articulación, la fricción, el juego angular entre los engranes, zona muerta provocado por el uso de engranajes, reductores cyclo, reductores armónicos, bandas dentadas, cadenas, cables, tornillos sin fin, además del ruido en las mediciones, perturbaciones exógenas como pares o fuerzas debido a un posible contacto físico del cuerpo del robot o interacción con objetos externos y la incertidumbre en el modelo. Existen excelentes trabajos de investigación donde se abordan los problemas relacionados con el desempeño de los robots manipuladores, por ejemplo en [2], [12], [36] y [38] se reportaron los diseños de sistemas de control que aseguran estabilidad asintótica en forma global para robots manipuladores libres de perturbaciones. Respecto a los controladores robustos aplicados a los robot que operan bajo condiciones inciertas y reportan las condiciones de estabilidad asintótica en forma global y local se encuentra lo reportado por [15], [29], [40] y [33]. En los últimos años, se han reportado numerosos trabajos donde abordan de manera exitosa los problemas ya mencionados que afectan a los robots manipuladores, cada una de estas metodologías atacan las no linealidades de manera distinta o independiente, por ejemplo, en [6] proponen un control adaptativo para resolver el problema de regulación de posición en un robot manipulador con incertidumbre en el modelo y demostrar estabilidad asintótica en forma global. Otra técnica de control muy empleada para resolver el problema de incertidumbre en el modelo y para el rechazo de perturbaciones externas del tipo acopladas es el control por modos deslizantes formulada por [41], donde logra estabilidad en tiempo finito, pero uno de sus inconvenientes es el efecto de *chattering* en la señal de control (oscilaciones de alta frecuencia y baja amplitud). En [40] resuelven el inconveniente de alcance de la superficie deslizante y se reduce el efecto del *chattering* con la propuesta del controlador por modos deslizantes integral; en [3] proponen un control de estructura variable compuesto por el control por modos deslizantes integral y un controlador lineal óptimo cuadrático para sistemas no lineales variantes en el tiempo en presencia de perturbaciones no acopladas. Por su parte, en [9] se propone el control *super-twisting* aplicado a un robot manipulador que opera bajo incertidumbre de Lipschitz, los autores emplean la derivada de la función de Lyapunov para un modelo nominal del robot que se usa para el diseño de la superficie deslizante, proporcionando una convergencia teóricamente exacta de los estados del sistema incierto al origen mediante una señal de control continua.

Una vez determinada la ecuación de movimiento del robot manipulador, se determina la ley de control que cumpla con el objetivo de control, la implementación del sistema en lazo cerrado en el dominio discreto puede considerarse una buena aproximación si las tasas de muestreo son lo suficientemente altas en comparación con la dinámica del sistema [18], sin embargo, el aumento del tiempo de muestreo, conduce a la degradación e inestabilidad del rendimiento [5]. Normalmente las leyes de control se derivan con la hipótesis implícita de discretizar los controladores continuos, y derivado de la complejidad de los algoritmos de control es indispensable el uso de un sistema digital, usualmente, las leyes de control se discretizan con una aproximación aceptable a la integración continua [26]. Este método es comúnmente utilizado, sin embargo, según [25] no es completamente correcto, incluso si

tenemos resultados que muestran la efectividad práctica del mismo. La metodología correcta según [30] debería ser, obtener primero el modelo discreto del sistema y luego diseñar una ley de control discreta, sin embargo solo se demostró para sistemas lineales invariantes en el tiempo. A la par que [20] propone que los modelos discretos de los sistemas no lineales sean compatibles con las capacidades de los sistemas digitales y preserven las características físicas fundamentales (conservación de la energía y el momento) de los sistemas continuos, satisfaciendo los principios fundamentales de la mecánica clásica, bajo este contexto propuso un modelo para un robot manipulador discreto para aplicaciones de control y simulación. En la literatura existen una gran variedad de metodologías que apoyan en el proceso de discretización de los sistemas lineales y no lineales, por ejemplo: en [39] discretizan directamente la ecuación de movimiento despreciando los términos no lineales, en [26] usaron una discretización directa de Euler de la ecuación de movimiento completa y obtuvieron resultados adaptativos discretos alentadores, en [23], conservan las integrales de movimiento y se aproxima a la dinámica con la regla trapezoidal, en [26] discretizan la integral de movimiento utilizando aproximaciones de Euler, en [10] emplean la regla trapezoidal conjugada con la regla del punto medio implícito para la implementación discreta de alta precisión de controladores de robots manipuladores, en [1] se propone un control basado en energía en tiempo discreto con el método IDA-PBC para sistemas mecánicos lineales que utilizan el gradiente discreto de punto medio que conserva la energía, en [10] emplean un esquema de integración con el objetivo de preservar el espacio de fase del flujo de un sistema hamiltoniano en tiempo discreto. Recientemente, en [17] propusieron la asignación de valores propios en tiempo discreto basada en la definición de un sistema objetivo, utilizando la regla del punto medio. Las ventajas en comparación con la implementación del controlador en tiempo continuo incluyen la estabilidad incondicional y establecer el periodo de muestreo dentro de los límites del teorema de muestreo. En [18] se presenta la formulación de una técnica de control basada en energía en tiempo discreto para sistemas mecánicos, en particular robots manipuladores, basados en integración numérica simpléctica. La conformación de la energía es el argumento central de los controles basados en la pasividad, y los esquemas de integración numérica simpléctica conservan un hamiltoniano modificado. Por lo tanto, según Kotyczka y Toma este método es apropiado para traducir el paso de formación de energía a tiempo discreto.

La aportación de este trabajo de tesis consiste en la discretización del controlador PD no lineal con compensación de la dinámica del robot sobre la trayectoria deseada mediante la técnica numérica de Euler, aplicado a un robot manipulador de tres grados de libertad denominado Niryo One. El desempeño del sistema de control en lazo cerrado se verifica empleando MatLab/Simulink y CoppeliaSim.

1.2. Estructura de la tesis

Este trabajo de tesis está estructurado de la siguiente manera:

El Capítulo 2 se desarrolla con la historia de los robots de manera general, hasta llegar a los fundamentos de los robots manipuladores.

En el Capítulo 3 se detalla el modelado dinámico y parametrización del robot manipulador Niryo One en tiempo continuo.

El Capítulo 4 muestra el diseño del control no lineal PD aplicado al robot manipulador Niryo One en tiempo continuo, el análisis de unicidad del punto de equilibrio, el análisis de estabilidad asintótica en forma global y la simulación del caso numérico.

En el Capítulo 5 se muestra la discretización del sistema de control en lazo cerrado y la simulación tanto en Matlab/Simulink como en CoppeliaSim.

Para finalizar, en el Capítulo 6 se presentan los resultados obtenidos de las simulaciones y conclusiones, la comparación de parámetros entre ellas, así como sugerencias para trabajos futuros.

1.3. Objetivos

1.3.1. Objetivo general

Diseñar un controlador PD discreto no lineal de múltiples entradas y múltiples salidas (MEMS) con compensación de la trayectoria deseada, aplicado al robot manipulador Niryo One, para resolver el problema de seguimiento de trayectorias en presencia de perturbaciones acopladas.

1.3.2. Objetivos específicos

1. Obtener el modelo dinámico del robot manipulador Niryo One.
2. Diseñar el controlador en tiempo continuo.
3. Discretizar el controlador.
4. Validar numéricamente el sistema de control.
5. Analizar el desempeño del sistema perturbado en lazo cerrado.

1.4. Hipótesis

El controlador PD no lineal discreto con MEMS con compensación de dinámica residual aplicado al robot Niryo One en la plataforma de CoppeliaSim cumple con su objetivo de control.

1.5. Limitaciones

El controlador PD discreto no lineal, con compensación de la trayectoria deseada aplicado al robot Niryo One, para resolver el problema de seguimiento de trayectorias, se limita a robots manipuladores de l GDL con articulaciones rígidas, sujetas a perturbaciones acopladas.

Capítulo 2

Fundamentos de la robótica

2.1. Historia de la robótica

El origen de la palabra *robot* data del año 1920, cuando apareció por primera vez en R.U.R. (Rossum's Universal Robots), obra literaria del dramaturgo Checoslovaco, Karel Čapek. *Robota* es una palabra Checoslovaca que significa trabajador o siervo[42].

En 1942, Isaac Asimov, escritor de novelas de ciencia ficción, creó la palabra *Robótica* para describir el campo de estudio de los robots. Fué él también quien inventó las tres leyes de la robótica [43].

A pesar de que el término *robot* es reciente, se tienen registros que siglos atrás aparecieron máquinas autómatas para realizar tareas humanas, por ejemplo, entre 1352 y 1354 se creó un reloj astronómico en la Catedral de Notre-Dame en Estrasburgo. Tenía un gallo mecánico hecho de madera, que movía sus alas y pico y cantaba por medio de fuelles. Mostrado en la imagen 2.1, este reloj es conocido también como el reloj de "los Tres Reyes".

Al-Jazari, conocido por algunos como el padre de la robótica, sirvió como inventor en la provincia de Diyarbakır. En 1206, el año de su muerte, se publicó su trabajo *El libro del conocimiento de los ingeniosos dispositivos mecánicos*, donde muestra un catálogo de sus máquinas con diagramas detallados e ilustraciones coloridas como la que se aprecia en la Figura 2.2, una fuente con forma de pavo real, que se usaba para elevar agua contenida en un tanque mediante la activación mecánica de flotadores [8].

Siglos más tarde, Leonardo da Vinci menciona en sus trabajos a Al-Jazari, pues estaba fascinado con sus autómatas. Leonardo crea en 1495 el caballero mecánico, visto en la Figura 2.3. Esta era una máquina que imitaba los movimientos humanos por medio de un sistema de palancas y cables que controlaban las extremidades superiores e inferiores por separado.

La historia cuenta que Leonardo da Vinci creó un león mecánico, el cual caminaba unos cuantos pasos por sí solo y al final, de su pecho, presentaba unas lilis al Rey de Francia,



Figura 2.1 Rossi, et al., 2009. Apariencia externa del gallo del reloj de Estrasburgo

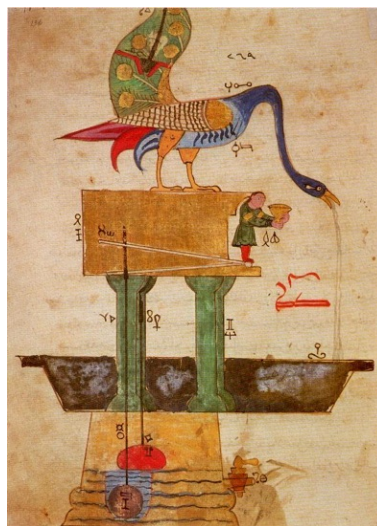


Figura 2.2 Fuente en forma de pavo real. Recuperado de <https://muslimheritage.com/al-jazari-the-mechanical-genius/>

Francis I, como un gesto diplomático (el león es el símbolo de Florencia y las lilis la flor de lis de Francia). En la Figura 2.4 se muestra una recreación computarizada de lo que pudo ser el león mecánico.

El avance en la robótica ha sido gigantesco, y en la actualidad, existen distintos tipos de robots, como industriales o de servicios, así como definiciones para ellos que son aceptadas por distintos organismos o instituciones, como la Organización Internacional de Estandarización, que define a los robots industriales como un *manipulador automáticamente controlado, programable y multiusos, programable en tres o más ejes*, que pueden ser usados para distintos procesos industriales, ya sean móviles o fijos.

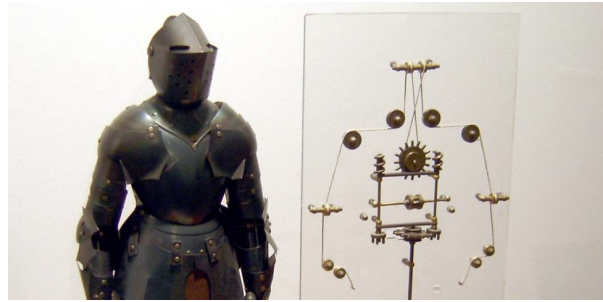


Figura 2.3 Máquina con aspecto humano. Recuperado de https://elpais.com/tecnologia/2019/04/25/actualidad/1556188737_786641.html



Figura 2.4 León mecánico. Recuperado de <https://cacm.acm.org/blogs/blog-cacm/240615-leonardo-da-vincis-robot-lion/fulltext>

2.2. Fundamentos de robots manipuladores

Los robots manipuladores son un tipo de robots usados ampliamente en la industria para la manipulación de objetos que para un humano podría ser pesado, tedioso o peligroso. Existen diferentes configuraciones que atienden a los movimientos necesarios para realizar sus tareas. Los robots manipuladores pueden ser clasificados de la siguiente manera (ver Figura 2.5):

- Cartesiano, que se desplaza linealmente sobre los ejes x , y , z .
- SCARA, con dos articulaciones rotacionales y una prismática.
- Articulado, tienen al menos 3 ejes articulados.
- Paralelo/Delta, un robot de uniones prismáticas o rotativas.
- Cilíndrico, cuyos ejes forman un sistema de coordenadas cilíndricas.

En la Figura 2.6 podemos ver un robot manipulador de 6 GDL de la empresa ABB, capaz de levantar cargas de hasta 800 Kg a un alcance de 4.2 m, magnitudes muy superiores

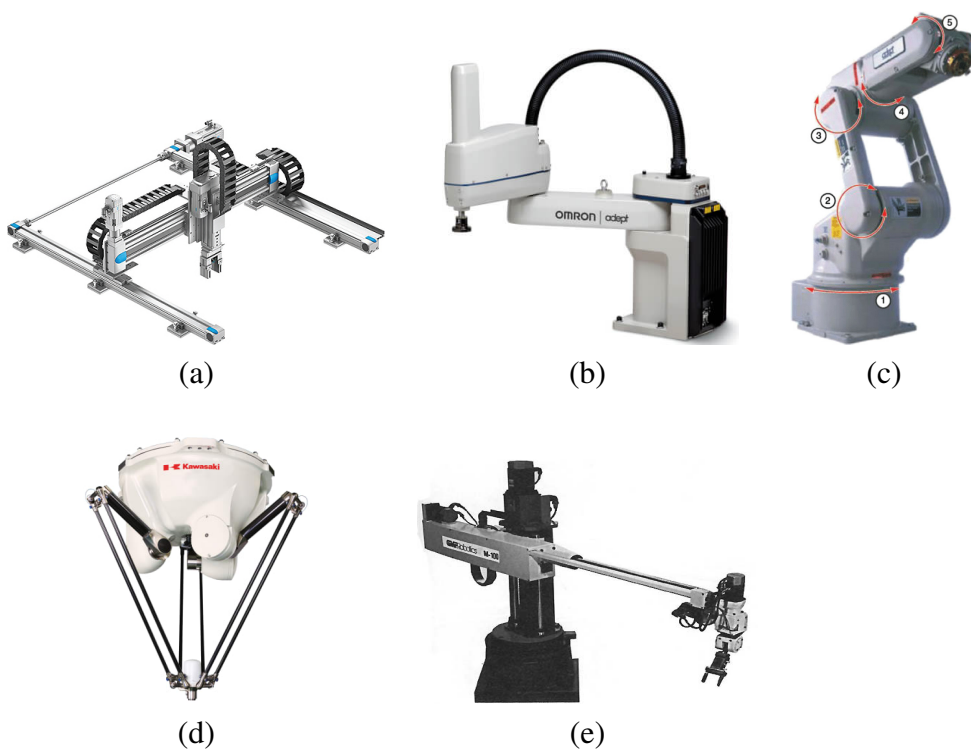


Figura 2.5 Algunas configuraciones de robots manipuladores: (a) Cartesiano, (b) SCARA, (c) Articulado, (d) Paralelo - Delta, (e) Cilíndrico.

a las logradas por un humano.



Figura 2.6 Robot manipulador del fabricante ABB. Recuperado de <https://new.abb.com/products/robotics/industrial-robots/irb-8700>

2.2.1. Tipos de articulaciones

Un robot manipulador industrial consiste en un conjunto de enlaces y articulaciones. Tal como sucede en el cuerpo humano, las articulaciones permiten el movimiento relativo entre dos enlaces del robot, que son la parte rígida. Los eslabones, también llamados ejes, unidos por cada articulación son llamados también como eslabón de entrada y de salida.

El enlace que comúnmente se encuentra fijo a la base sería el enlace de entrada de la articulación 1, el enlace 2 es el de salida de la articulación 1 y el de entrada de la articulación 2. En la Figura 2.7 se observa un ejemplo de los enlaces y articulaciones.

Las articulaciones se dividen en 5 tipos:

- Lineal (tipo L): movimiento relativo entre el enlace de entrada y de salida en un movimiento traslacional telescópico, siendo los 2 ejes paralelos.
- Ortogonal (tipo O): también es un movimiento traslacional, pero los enlaces son perpendiculares.
- Rotacional (tipo R): movimiento relativo rotacional, con el eje de rotación perpendicular a los ejes de entrada y salida.
- Torsión (tipo T): movimiento rotacional, pero el eje de rotación es paralelo a los 2 ejes.

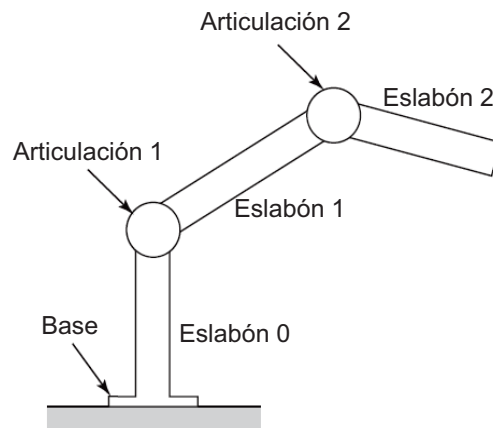


Figura 2.7 Articulaciones y enlaces de un robot manipulador.

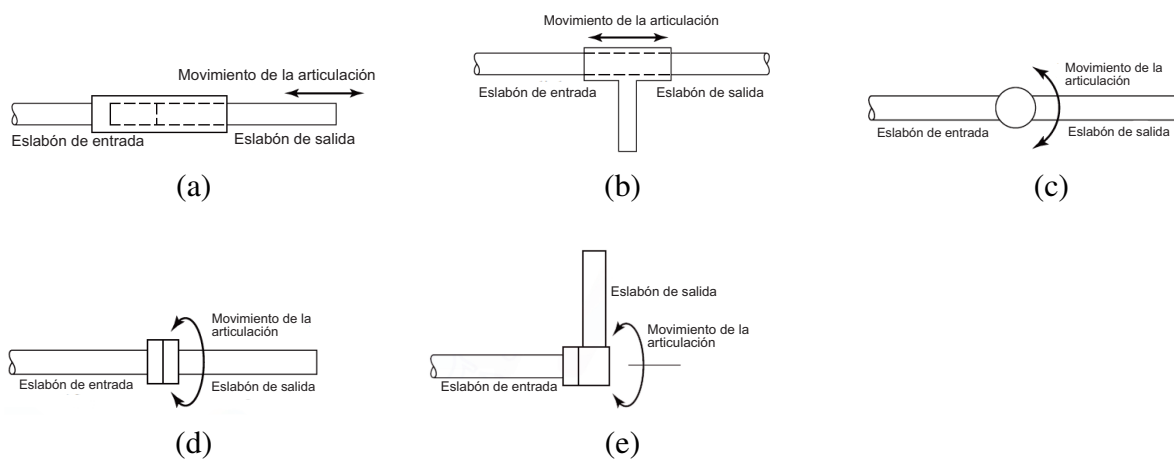


Figura 2.8 Ejemplo de los tipos de articulación mecánica a) Lineal (tipo L), b) Ortogonal (tipo O), c) Rotacional (tipo R), d) Torsión (tipo T), e) Revolución (tipo V)

- Revolución (tipo V): con el eje de entrada paralelo al eje de rotación y el eje de salida, perpendicular a éste último.

El tipo de articulaciones, determinará el alcance de movimiento que tendrá el robot. Las traslacionales comúnmente son de 1 metro, y las rotacionales pueden ser desde unos cuantos grados hasta varias revoluciones completas.

2.2.2. Grados de libertad

Al hablar GDL, comúnmente estamos refiriéndonos a las articulaciones móviles que tiene el robot, existiendo una relación directa entre estos elementos y los enlaces del mismo.

Los GDL de un robot manipulador, representan el hombro, codo y muñeca de un brazo humano, y se encargan de definir la ubicación de un objeto en el espacio de la siguiente manera:

- De 1 a 3, ubica el objeto en un espacio desde unidimensional hasta tridimensional (x , y , z). En la Figura 2.9 se muestra un brazo robótico de 3 GDL.
- De 4 a 6, ubica el objeto en un espacio tridimensional, así como también modifica su orientación (guiñada, cabeceo y balanceo o alabeo). En la Figura 2.10 se muestran las articulaciones de un brazo de 6 GDL.
- El robot de 7 o más GDL ubica en el espacio tridimensional y orientación, justo como uno de 6, pero puede agregarse un desplazamiento del robot por medio de rieles o movimientos redundantes en sus articulaciones, como podemos ver en la Figura 2.11.

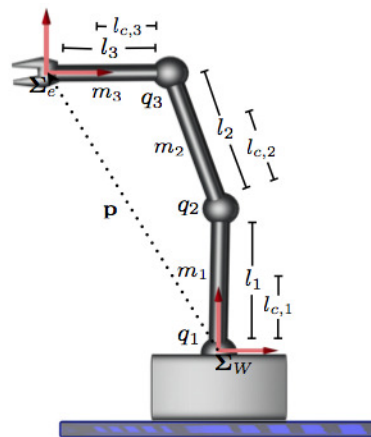


Figura 2.9 Brazo robótico de 3 GDL. Recuperado de <https://sir.upc.edu/wikis/roblab/index.php/Projects/3DOFRegressor>

El robot manipulador es conocido también como brazo robótico, los cuales son sistemas mecánicos acoplados con Múltiples Entradas Múltiples Salidas (MIMO, por sus siglas en inglés) [7], con una dinámica no lineal e incertidumbres, como fricción y perturbaciones [19].

Comúnmente el robot está construido además de sus brazos, de un efector final, como pinzas o dispositivos para tareas específicas. Las articulaciones del brazo, rotativas o traslacionales como se observa en la Figura 2.12, permiten su movilidad en el espacio, y mientras más articulaciones tenga, se facilita su acceso a zonas más difíciles; el inconveniente es, que aumenta la complejidad del controlador, ya que existen más errores que deben ser corregidos en los parámetros deseados.

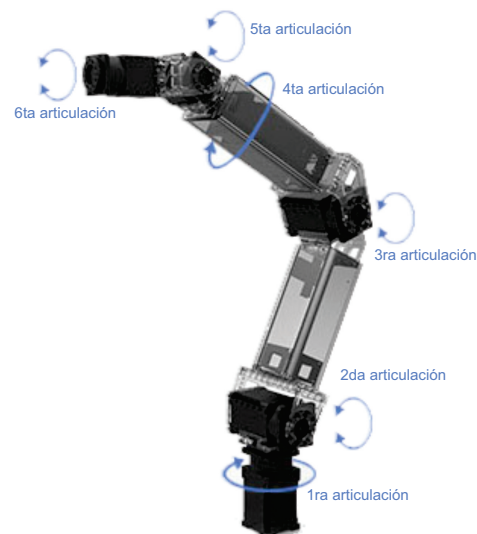


Figura 2.10 Brazo robótico de 6 GDL. Recuperado de <https://www.thingbits.in/products/robotis-manipulator-h-6dof-robotic-arm>

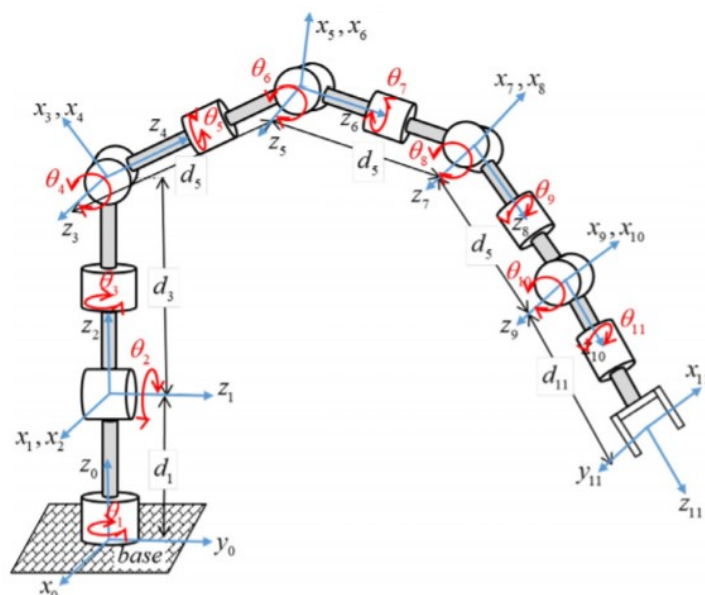


Figura 2.11 Kouabon, et al. 2020. Brazo robótico de 11 GDL

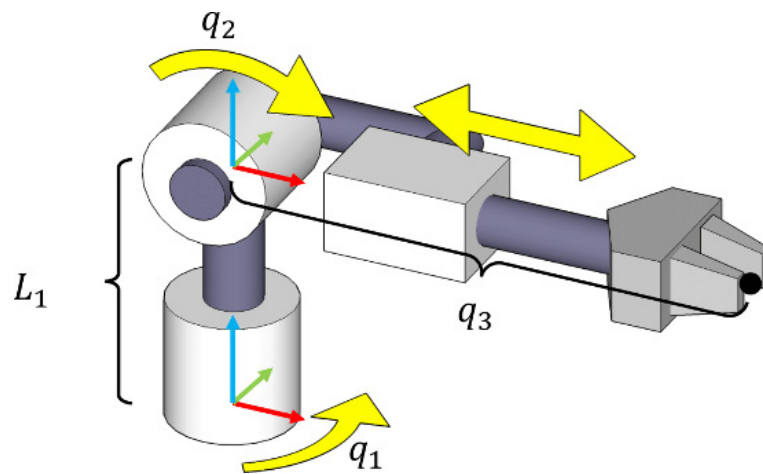


Figura 2.12 Articulaciones rotacionales y traslacionales. Recuperado de <http://motion.cs.illinois.edu/RoboticSystems/Kinematics.html>

Capítulo 3

Modelado dinámico del robot manipulador Niryo One

3.1. Modelado dinámico

El modelo dinámico del robot manipulador de tres GDL que se muestra en la Figura 3.1, donde se aprecian los parámetros del robot, se describe a continuación:

$q_i(t)$ es la posición angular de la i -ésima articulación rígida del robot, para valores de $i = 1, 2, 3$, y l_i son las longitudes en metros de los eslabones.

El análisis del modelado dinámico inicia con la ecuación de Euler-Lagrange

$$L(q, \dot{q}) = K(q, \dot{q}) - U(q), \quad (3.1)$$

donde $K(q, \dot{q})$ es un vector de \mathbb{R}^3 que representa la energía cinética de los eslabones del robot y $U(q)$ es un vector de \mathbb{R}^3 que representa la energía potencial.

Para calcular la energía cinética del diagrama de cuerpo libre del robot (ver Figura 3.2) se emplea la fórmula enunciada a continuación:

$$K_i(q, \dot{q}) = \frac{1}{2} m_i V_i^T(t) V_i(t), \quad (3.2)$$

donde i es el número de eslabones, m_i es la masa del i -ésimo eslabón en Kilogramos y $V_i(t)$ es el vector de velocidad ($\frac{m}{s}$) del eslabón respecto al plano en tres dimensiones.

Para proceder al cálculo de la energía cinética de cada eslabón, se necesita conocer el vector de velocidades de los eslabones, para esto, una parte fundamental es el conocimiento de las coordenadas de la posición del eslabón 1 de acuerdo a la Figura 3.2

$$\left. \begin{aligned} x_1(t) &= l_1 \cos(q_1) \cos(q_2) \\ y_1(t) &= l_1 \sin(q_1) \cos(q_2) \\ z_1(t) &= l_1 \sin(q_2) \end{aligned} \right\} \quad (3.3)$$

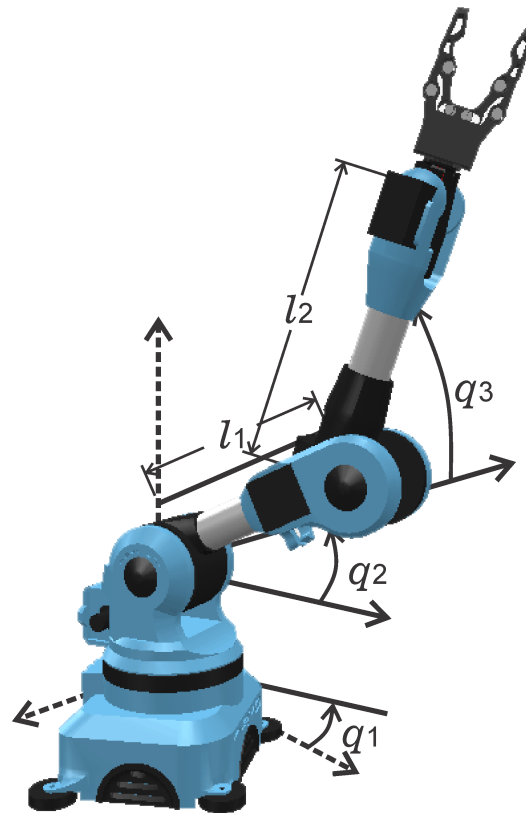


Figura 3.1 Robot manipulador Niryo One, donde $q_1(t)$, $q_2(t)$ y $q_3(t)$ representan las articulaciones que indicarán la posición del brazo. *Software Coppeliasim*

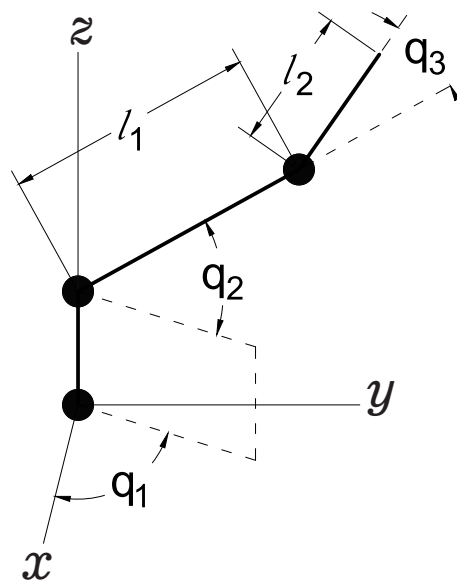


Figura 3.2 Diagrama de cuerpo libre del robot Niryo One con 3 GDL. Elaboración propia

y el vector de velocidades se determina por la derivada con respecto al tiempo de $x_1(t)$, $y_1(t)$ y $z_1(t)$. Haciendo uso de identidades trigonométricas pitagóricas y simplificando términos semejantes por álgebra, se llega a las siguientes ecuaciones:

$$\left. \begin{aligned} \dot{x}_1 &= -l_1 \sin(q_2) \cos(q_1) \dot{q}_2 - l_1 \sin(q_1) \cos(q_2) \dot{q}_1 \\ \dot{y}_1 &= -l_1 \sin(q_1) \sin(q_2) \dot{q}_2 + l_1 \cos(q_1) \cos(q_2) \dot{q}_1 \\ \dot{z}_1 &= l_1 \cos(q_2) \dot{q}_2 \end{aligned} \right\} \quad (3.4)$$

Una vez obtenidas las derivadas, se obtiene el producto de vectores de velocidad para cada eslabón mediante la siguiente expresión:

$$V_i^T(t) V_i(t) = [\dot{x}_i \quad \dot{y}_i \quad \dot{z}_i] \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \end{bmatrix} \quad (3.5)$$

y al multiplicar los vectores se obtiene un trinomio al cuadrado, el cual al ser desarrollado se obtiene la siguiente ecuación que indica el producto vectorial de la velocidad del eslabón 1

$$V_1^T(t) V_1(t) = l_1^2 \cos^2(q_2) \dot{q}_1^2 + l_1^2 \dot{q}_2^2 \quad (3.6)$$

Ya con el producto de vectores de velocidad, la ecuación 3.6 se sustituye en la fórmula 3.2 para obtener lo siguiente:

$$K_1(q, \dot{q}) = \frac{1}{2} m_1 l_1^2 \cos^2(q_2) \dot{q}_1^2 + \frac{1}{2} m_1 l_1^2 \dot{q}_2^2 + \frac{1}{2} I_1 \dot{q}_1^2 + \frac{1}{2} I_2 \dot{q}_2^2 \quad (3.7)$$

Continuando con la ecuación de Lagrange, se procede al cálculo de la energía potencial, a través de la siguiente ecuación:

$$U_i(q) = m_i g h_i(t), \quad (3.8)$$

donde $U_i(q)$ representa la energía potencial del eslabón, m la masa del eslabón en Kilogramos, g la constante gravitacional en metros por segundo cuadrado y $h_i(t)$ la altura o posición del eslabón en metros, en referencia a la base.

Tratándose del cálculo de la energía potencial del eslabón 1, la altura de éste se encuentra sobre el eje z , por lo tanto, al sustituir los datos de ese eslabón obtenemos:

$$U_1(q) = m_1 g l_1 \sin(q_2) \quad (3.9)$$

Usando la ecuación de Lagrange descrita en la ecuación (3.1) y las ecuaciones (3.2) y (3.9) de las energías cinética y potencial, tenemos como resultado lo siguiente:

$$L_1(q, \dot{q}) = \frac{1}{2} m_1 l_1^2 \cos^2(q_2) \dot{q}_1^2 + \frac{1}{2} m_1 l_1^2 \dot{q}_2^2 + \frac{1}{2} I_1 \dot{q}_1^2 + \frac{1}{2} I_2 \dot{q}_2^2 - m_1 g l_1 \sin(q_2) \quad (3.10)$$

Para obtener la ecuación de Lagrange del eslabón 2, se repiten los pasos anteriores con las coordenadas x_2 , y_2 y z_2

$$\left. \begin{aligned} x_2(t) &= [l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)] \cos(q_1) \\ y_2(t) &= [l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)] \sin(q_1) \\ z_2(t) &= l_1 \sin(q_2) + l_2 \sin(q_2 + q_3) \end{aligned} \right\} \quad (3.11)$$

Derivando (3.11) y después de aplicar identidades trigonométricas y álgebra, se obtienen las siguientes expresiones:

$$\left. \begin{aligned} \dot{x}_2 &= -l_1 \sin(q_1) \cos(q_2) \dot{q}_1 - l_2 \sin(q_1) \cos(q_2 + q_3) \dot{q}_1 - l_1 \sin(q_2) \cos(q_1) \dot{q}_2 \\ &\quad - l_2 \sin(q_2 + q_3) \cos(q_1) (\dot{q}_2 + \dot{q}_3) \\ \dot{y}_2 &= l_1 \cos(q_1) \cos(q_2) \dot{q}_1 + l_2 \cos(q_1) \cos(q_2 + q_3) \dot{q}_1 - l_1 \sin(q_1) \sin(q_2) \dot{q}_2 \\ &\quad - l_2 \sin(q_1) \sin(q_2 + q_3) (\dot{q}_2 + \dot{q}_3) \\ \dot{z}_2 &= l_1 \cos(q_2) \dot{q}_2 + l_2 \cos(q_2 + q_3) (\dot{q}_2 + \dot{q}_3) \end{aligned} \right\} \quad (3.12)$$

Se sustituye (3.12) en (3.5) para el eslabón 2, y después de aplicar propiedades trigonométricas y álgebra, se obtiene el cuadrado del vector velocidad 2:

$$V_2^T(t) V_2(t) = l_1^2 \dot{q}_2^2 + l_2^2 (\dot{q}_2 + \dot{q}_3)^2 + 2l_1 l_2 \cos(q_3) \dot{q}_2 (\dot{q}_2 + \dot{q}_3) + \dot{q}_1^2 [l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)]^2 \quad (3.13)$$

Al tener el cuadrado de la velocidad, se sustituye en (3.2), teniendo como resultado la energía cinética para el eslabón 2:

$$\begin{aligned} K_2(q, \dot{q}) &= \frac{1}{2} m_2 l_1^2 \dot{q}_2^2 + \frac{1}{2} m_2 l_2^2 (\dot{q}_2 + \dot{q}_3)^2 + m_2 l_1 l_2 \cos(q_3) \dot{q}_2 (\dot{q}_2 + \dot{q}_3) \\ &\quad + \frac{1}{2} m_2 \dot{q}_1^2 [l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)]^2 + \frac{1}{2} I_3 (\dot{q}_2 + \dot{q}_3)^2 \end{aligned} \quad (3.14)$$

La segunda parte de la ecuación de Lagrange es la energía potencial (3.8), y para este caso la altura a considerar es desde el origen, por lo tanto se toma en cuenta $l_1 + l_2$:

$$U_2(q) = m_2 g l_1 \sin(q_2) + m_2 g l_2 \sin(q_2 + q_3) \quad (3.15)$$

Al tener las ecuaciones de las energías del eslabón 2, se sustituyen en la ecuación de Lagrange (3.1) y se consigue llegar a la expresión siguiente:

$$\begin{aligned}
L_2(q, \dot{q}) = & \frac{1}{2}m_2l_1^2\dot{q}_2^2 + \frac{1}{2}m_2l_2^2(\dot{q}_2 + \dot{q}_3)^2 + m_2l_1l_2\cos(q_3)\dot{q}_2(\dot{q}_2 + \dot{q}_3) \\
& + \frac{1}{2}m_2\dot{q}_1^2[l_1\cos(q_2) + l_2\cos(q_2 + q_3)]^2 + \frac{1}{2}I_3(\dot{q}_2 + \dot{q}_3)^2 - m_2gl_1\sin(q_2) - m_2gl_2\sin(q_2 + q_3)
\end{aligned} \quad (3.16)$$

La ecuación de Lagrange total de los 2 eslabones se obtendrá con la suma de las ecuaciones (3.10) y (3.16):

$$\begin{aligned}
L(q, \dot{q}) = & \frac{1}{2}m_1l_1^2\cos^2(q_2)\dot{q}_1^2 + \frac{1}{2}m_1l_1^2\dot{q}_2^2 + \frac{1}{2}I_1\dot{q}_1^2 + \frac{1}{2}I_2\dot{q}_2^2 + \frac{1}{2}m_2l_1^2\dot{q}_2^2 + \frac{1}{2}m_2l_2^2(\dot{q}_2 + \dot{q}_3)^2 \\
& + m_2l_1l_2\cos(q_3)\dot{q}_2(\dot{q}_2 + \dot{q}_3) + \frac{1}{2}m_2\dot{q}_1^2[l_1\cos(q_2) + l_2\cos(q_2 + q_3)]^2 + \frac{1}{2}I_3(\dot{q}_2 + \dot{q}_3)^2 \\
& - m_1gl_1\sin(q_2) - m_2gl_1\sin(q_2) - m_2gl_2\sin(q_2 + q_3)
\end{aligned} \quad (3.17)$$

Con las ecuaciones de Lagrange, lo siguiente es obtener la ecuación de movimiento de Lagrange (3.18), y de esta forma tener la ecuación del par aplicado a cada articulación

$$\frac{d}{dt} \left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}_i} \right] - \left[\frac{\partial L(q, \dot{q})}{\partial q_i} \right] = \tau_i \quad (3.18)$$

Desarrollar (3.18) para cada articulación del robot, iniciando con $q_1(t)$

$$\frac{\partial L(q, \dot{q})}{\partial \dot{q}_1} = m_1l_1^2\cos^2(q_2)\dot{q}_1 + I_1\dot{q}_1 + m_2[l_1\cos(q_2) + l_2\cos(q_2 + q_3)]^2\dot{q}_1 \quad (3.19)$$

$$\begin{aligned}
\frac{d}{dt} \left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}_1} \right] = & m_1l_1^2\cos^2(q_2)\ddot{q}_1 - 2m_1l_1^2\sin(q_2)\cos(q_2)\dot{q}_1\dot{q}_2 + I_1\ddot{q}_1 + m_2l_1^2\cos^2(q_2)\ddot{q}_1 \\
& + 2m_2l_1l_2\cos(q_2)\cos(q_2 + q_3)\ddot{q}_1 + m_2l_2^2\cos^2(q_2 + q_3)\ddot{q}_1 - 2m_2l_1^2\sin(q_2)\cos(q_2)\dot{q}_1\dot{q}_2 \\
& - 2m_2l_1l_2\cos(q_2)\sin(q_2 + q_3)(\dot{q}_2 + \dot{q}_3)\dot{q}_1 - 2m_2l_1l_2\sin(q_2)\cos(q_2 + q_3)\dot{q}_1\dot{q}_2 \\
& - 2m_2l_2^2\sin(q_2 + q_3)\cos(q_2 + q_3)(\dot{q}_2 + \dot{q}_3)\dot{q}_1
\end{aligned} \quad (3.20)$$

Al desarrollar la derivada parcial del Lagrangiano con respecto a q_1 el resultado es cero

$$\frac{\partial L(q, \dot{q})}{\partial q_1} = 0 \quad (3.21)$$

Desarrollar (3.18), ahora para la articulación $q_2(t)$

$$\begin{aligned} \frac{\partial L(q, \dot{q})}{\partial \dot{q}_2} = & m_1 l_1^2 \dot{q}_2 + m_2 l_1^2 \dot{q}_2 + m_2 l_2^2 (\dot{q}_2 + \dot{q}_3) + 2m_2 l_1 l_2 \cos(q_3) \dot{q}_2 \\ & + (I_2 + I_3) \dot{q}_2 + m_2 l_1 l_2 \cos(q_3) \dot{q}_3 + I_3 \dot{q}_3 \end{aligned} \quad (3.22)$$

$$\begin{aligned} \frac{d}{dt} \left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}_2} \right] = & m_1 l_1^2 \ddot{q}_2 + m_2 l_1^2 \ddot{q}_2 + m_2 l_2^2 (\ddot{q}_2 + \ddot{q}_3) + 2m_2 l_1 l_2 \cos(q_3) \ddot{q}_2 - 2m_2 l_1 l_2 \sin(q_3) \dot{q}_2 \dot{q}_3 \\ & + (I_2 + I_3) \ddot{q}_2 + m_2 l_1 l_2 \cos(q_3) \ddot{q}_3 - m_2 l_1 l_2 \sin(q_3) \dot{q}_3^2 + I_3 \ddot{q}_3 \end{aligned} \quad (3.23)$$

$$\begin{aligned} \frac{\partial L(q, \dot{q})}{\partial q_2} = & -m_1 l_1^2 \sin(q_2) \cos(q_2) \dot{q}_1^2 \\ & + m_2 [l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)] [-l_1 \sin(q_2) - l_2 \sin(q_2 + q_3)] \dot{q}_1^2 - m_1 g l_1 \cos(q_2) \\ & - m_2 g l_1 \cos(q_2) - m_2 g l_2 \cos(q_2 + q_3) \end{aligned} \quad (3.24)$$

Desarrollar (3.18), ahora para la articulación $q_3(t)$

$$\frac{\partial L(q, \dot{q})}{\partial \dot{q}_3} = m_2 l_2^2 (\dot{q}_2 + \dot{q}_3) + m_2 l_1 l_2 \cos(q_3) \dot{q}_2 + I_3 \dot{q}_2 + I_3 \dot{q}_3 \quad (3.25)$$

$$\frac{d}{dt} \left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}_3} \right] = m_2 l_2^2 (\ddot{q}_2 + \ddot{q}_3) + m_2 l_1 l_2 \cos(q_3) \ddot{q}_2 - m_2 l_1 l_2 \sin(q_3) \dot{q}_2 \dot{q}_3 + I_3 \ddot{q}_2 + I_3 \ddot{q}_3 \quad (3.26)$$

$$\begin{aligned} \frac{\partial L(q, \dot{q})}{\partial q_3} = & -m_2 l_1 l_2 \sin(q_3) \dot{q}_2 (\dot{q}_2 + \dot{q}_3) - m_2 g l_2 \cos(q_2 + q_3) \\ & - m_2 [l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)] [l_2 \sin(q_2 + q_3)] \dot{q}_1^2 \end{aligned} \quad (3.27)$$

Los pares aplicados a cada articulación del robot manipulador están determinados por las siguientes expresiones:

$$\begin{aligned}\tau_1(t) = & (m_1 l_1^2 \cos^2(q_2) + I_1) \ddot{q}_1 + m_2 [l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)] \ddot{q}_1 \\ & - 2m_2 [l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)] [l_1 \sin(q_2) \dot{q}_2 + l_2 \sin(q_2 + q_3) (\dot{q}_2 + \dot{q}_3)] \dot{q}_1 \\ & - 2m_1 l_1^2 \sin(q_2) \cos(q_2) \dot{q}_1 \dot{q}_2 \quad (3.28)\end{aligned}$$

$$\begin{aligned}\tau_2(t) = & [m_1 l_1^2 + m_2 l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(q_3) + I_2 + I_3] \ddot{q}_2 + [m_2 l_2^2 + m_2 l_1 l_2 \cos(q_3) + I_3] \ddot{q}_3 \\ & - 2m_2 l_1 l_2 \sin(q_3) \dot{q}_2 \dot{q}_3 - m_2 l_1 l_2 \sin(q_3) \dot{q}_3^2 + m_1 l_1^2 \sin(q_2) \cos(q_2) \dot{q}_1^2 \\ & + m_1 g l_1 \cos(q_2) + m_2 g l_1 \cos(q_2) + m_2 g l_2 \cos(q_2 + q_3) \\ & - m_2 [l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)] [-l_1 \sin(q_2) - l_2 \sin(q_2 + q_3)] \dot{q}_1^2 \quad (3.29)\end{aligned}$$

$$\begin{aligned}\tau_3(t) = & [m_2 l_2^2 + m_2 l_1 l_2 \cos(q_3) + I_3] \ddot{q}_2 + [m_2 l_2^2 + I_3] \ddot{q}_3 + m_2 l_1 l_2 \sin(q_3) \dot{q}_2 (\dot{q}_2 + \dot{q}_3) \\ & + m_2 g l_2 \cos(q_2 + q_3) + m_2 [l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)] [l_2 \sin(q_2 + q_3)] \dot{q}_1^2 \quad (3.30)\end{aligned}$$

3.1.1. Representación en su forma compacta

La representación del modelo dinámico de un robot manipulador de l grados de libertad propuesta por Spong [37], está dada por la siguiente ecuación.

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) + f(\dot{q}) \dot{q} = \tau(t) + w(t) \quad (3.31)$$

donde $q, \dot{q}, \ddot{q} \in \mathbb{R}^3$ son los vectores de posición, velocidad y aceleración de las articulaciones, respectivamente, los cuales están formados por $q(t) = [q_1(t) \ q_2(t) \ q_3(t)]^T$, $\dot{q}(t) = [\dot{q}_1(t) \ \dot{q}_2(t) \ \dot{q}_3(t)]^T$, $M(q) \in \mathbb{R}^{3 \times 3}$ es la matriz de inercia, la cual es simétrica y definida positiva, $C(q, \dot{q}) \in \mathbb{R}^{3 \times 3}$ es la matriz de fuerzas centrífugas y de Coriolis, $g(q) \in \mathbb{R}^3$ es el vector de torque gravitacional, $f(\dot{q}) \in \mathbb{R}^3$ donde f es el vector de fricción, la cual es definida positiva y en la diagonal principal se localizan las constantes de fricción viscosa asociadas a cada articulación, $\tau(t) \in \mathbb{R}^3$ es el vector de fuerzas y pares aplicados a las articulaciones y $w(t) \in \mathbb{R}^3$ es el vector de perturbación formado por $w(t) = [0.38 \sin(2t) \ 0.38 \sin(4t) \ 0.38 \sin(8t)]^T$

Una vez obtenidas las ecuaciones de los pares aplicados, éstas se acomodan en forma matricial para generar la ecuación dinámica compacta (3.32) de la manera que se aprecia a

continuación:

$$\left\{ \begin{aligned} & \begin{bmatrix} m_{11}(q) & 0 & 0 \\ 0 & m_{22}(q) & m_{23}(q) \\ 0 & m_{32}(q) & m_{33}(q) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} c_{11}(q, \dot{q}) & c_{12}(q, \dot{q}) & 0 \\ c_{21}(q, \dot{q}) & c_{22}(q, \dot{q}) & c_{23}(q, \dot{q}) \\ c_{31}(q, \dot{q}) & c_{32}(q, \dot{q}) & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \\ & + \begin{bmatrix} 0 \\ g_2(q) \\ g_3(q) \end{bmatrix} + \begin{bmatrix} \alpha_{11}\dot{q}_1 \\ \alpha_{22}\dot{q}_2 \\ \alpha_{33}\dot{q}_3 \end{bmatrix} = \begin{bmatrix} \tau_1(t) \\ \tau_2(t) \\ \tau_3(t) \end{bmatrix} + \begin{bmatrix} w_1(t) \\ w_2(t) \\ w_3(t) \end{bmatrix} \end{aligned} \right\} \quad (3.32)$$

Los elementos de la matriz de inercia se muestran a continuación:

$$m_{11}(q) = m_1 l_1^2 \cos^2(q_2) + I_1 + m_2 [l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)]$$

$$m_{22}(q) = m_1 l_1^2 + m_2 l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(q_3) + I_2 + I_3$$

$$m_{23}(q) = m_2 l_2^2 + m_2 l_1 l_2 \cos(q_3) + I_3$$

$$m_{32}(q) = m_2 l_2^2 + m_2 l_1 l_2 \cos(q_3) + I_3$$

$$m_{33}(q) = m_2 l_2^2 + I_3$$

La matriz de fuerzas centrífugas y de Coriolis, tiene como elementos las siguientes funciones:

$$c_{11}(q, \dot{q}) = -2m_2 [l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)] [l_1 \sin(q_2) \dot{q}_2 + l_2 \sin(q_2 + q_3) (\dot{q}_2 + \dot{q}_3)]$$

$$c_{12}(q, \dot{q}) = -2m_1 l_1^2 \sin(q_2) \cos(q_2) \dot{q}_1$$

$$c_{21}(q, \dot{q}) = m_1 l_1^2 \sin(q_2) \cos(q_2) \dot{q}_1 - m_2 [l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)] [-l_1 \sin(q_2) - l_2 \sin(q_2 + q_3)] \dot{q}_1$$

$$c_{22}(q, \dot{q}) = -2m_2 l_1 l_2 \sin(q_3) \dot{q}_3$$

$$c_{23}(q, \dot{q}) = -m_2 l_1 l_2 \sin(q_3) \dot{q}_3$$

$$c_{31}(q, \dot{q}) = m_2 [l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)] [l_2 \sin(q_2 + q_3)] \dot{q}_1$$

$$c_{32}(q, \dot{q}) = m_2 l_1 l_2 \sin(q_3) (\dot{q}_2 + \dot{q}_3)$$

El vector de fuerzas gravitacionales se conforma por las funciones:

$$g_2(q) = [m_1 l_1 \cos(q_2) + m_2 l_1 \cos(q_2) + m_2 l_2 \cos(q_2 + q_3)] g$$

$$g_3(q) = [m_2 l_2 \cos(q_2 + q_3)] g$$

Elementos que componen la matriz de fricción, donde α_{11} , α_{22} , y α_{33} son las constantes de fricción viscosa asociadas a cada articulación:

$$\alpha_{11} = 0.01$$

$$\alpha_{22} = 0.02$$

$$\alpha_{33} = 0.015$$

3.1.2. Cálculo de parámetros

Se calculan valores límites de las matrices de inercia, fuerza centrífuga y de Coriolis y el vector gravitacional [13], para el robot manipulador Niryo One, mismos que servirán para el cálculo de las ganancias al sintonizar el control. El desarrollo completo de estos cálculos se encuentra en la sección de Anexos.

Cálculo de $\lambda_{Max}\{M\}$

$$\max_{i,j,\mathbf{q}} |M_{i,j}(\mathbf{q})| = m_1 l_1^2 \cos^2(q_2) + I_1 + m_2 [l_1 \cos(q_2) + l_2 \cos(q_2 + q_3)]$$

La constante β deberá ser un valor mayor o igual a n veces la expresión anterior,

$$\beta \geq \left(\max_{i,j,\mathbf{q}} |M_{i,j}(\mathbf{q})| \right)$$

y definiendo $\lambda_{Max}\{M\} = \beta$, se obtiene que

$$\lambda_{Max}\{M\} = 1.5532 \text{ kg} \cdot \text{m}^2 \quad (3.33)$$

Cálculo de k_M

Considerando la matriz de inercia $M(q)$ y sus elementos, se calcula $\frac{\partial M_{i,j,\mathbf{q}}}{\partial q_k}$ de cada elemento (ver anexos), y para obtener la constante k_M se tiene que:

$$k_M \geq n^2 \left(\max_{i,j,k,\mathbf{q}} \left| \frac{\partial M_{i,j,\mathbf{q}}}{\partial q_k} \right| \right)$$

por lo tanto, la constante k_M estará dada por:

$$k_M \geq n^2 [-m_2 l_2 \sin(q_2 + q_3)]$$

es así como la constante k_M resulta como:

$$k_M \geq 1.5523 \text{ kg} \cdot \text{m}^2 \quad (3.34)$$

Cálculo de k_{C_1}

Considerando el vector de fuerzas centrífugas y de Coriolis de la siguiente forma:

$$C(q, \dot{q})\dot{q} = \begin{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}^T \\ \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}^T \\ \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}^T \end{bmatrix} \begin{bmatrix} C_1(q) \\ C_2(q) \\ C_3(q) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

y que la constante k_{C_1} se determina de la siguiente manera:

$$k_{C_1} \geq n^2 \left(\max_{i,j,k,\mathbf{q}} |C_{k_{ij}}(q)| \right)$$

la constante k_{C_1} resultará de calcular lo siguiente:

$$k_{C_1} \geq n^2 |-2m_2l_1l_2\cos(q_2)\sin(q_2+q_3) - 2m_2l_2^2\cos(q_2+q_3)\sin(q_2+q_3)|$$

por lo tanto, la constante k_{C_1} está determinada como:

$$k_{C_1} \geq 0.7751 \text{ kg} \cdot \text{m}^2 \quad (3.35)$$

Cálculo de k_{C_2}

Tomando de nuevo en cuenta el vector de fuerzas centrífugas y de Coriolis, se calcula la derivada parcial de cada elemento con respecto a las derivada parcial de los elementos del vector de articulaciones q .

Una vez calculado esto, se utiliza la siguiente fórmula para encontrar la constante k_{C_2}

$$k_{C_2} \geq n^3 \left(\max_{i,j,k,l,\mathbf{q}} \left| \frac{\partial C_{k_{ij}}(q)}{\partial q_l} \right| \right),$$

por lo tanto, la constante k_{C_2} se obtendrá con:

$$k_{C_2} \geq n^3 |-2m_2l_1l_2\cos q_2\cos(q_2+q_3) - 2m_2l_2^2\cos^2(q_2+q_3) + 2m_2l_2^2\sin^2(q_2+q_3)|,$$

resultando de esta manera:

$$k_{C_2} \geq 3.5457 \text{ kg} \cdot \text{m}^2 \quad (3.36)$$

Cálculo de k_g

Para el cálculo de esta constante, se toma en cuenta el vector de fuerzas gravitacionales y se obtiene la derivada parcial con respecto al vector de articulaciones $q(t)$, y utilizando la fórmula que se ve en seguida:

$$k_g \geq n \left(\max_{i,j,q} \left| \frac{\partial g_i(q)}{\partial q_j} \right| \right),$$

entonces la constante k_g se obtendrá de la siguiente manera:

$$k_g \geq n [-m_1l_1\sin(q_2) - m_2l_1\sin(q_2) - m_2l_2\sin(q_2+q_3)]g,$$

resultando la constante con la siguiente desigualdad:

$$k_g \geq 15.9025 \frac{\text{kg} \cdot \text{m}^2}{\text{s}^2} \quad (3.37)$$

Parámetro	Valor	Unidad
$\lambda_{\max}\{M\}$	1.5532	$kg \cdot m^2$
k_M	1.5523	$kg \cdot m^2$
k_{C_1}	0.7751	$kg \cdot m^2$
k_{C_2}	3.5457	$kg \cdot m^2$
k_g	15.9025	$\frac{kg \cdot m^2}{s^2}$
k_{h_1}	3.1004	$\frac{kg \cdot m^2}{s}$
k_{h_2}	106.1685	Nm

Tabla 3.1 Parámetros de robot Niryo One

3.1.3. Representación en espacios de estados

Un sistema puede ser representado también con variables de estado, las cuales son variables internas del sistema y la cantidad de éstas es igual al orden del sistema y su comportamiento es descrito para un determinado tiempo (t), además, permiten determinar cómo se comportará para otros valores de t [28].

Primero se definen las variables de estado, que serán la posición (rad) y velocidad angular ($\frac{rad}{s}$) de las articulaciones:

$$q(t), \dot{q}(t) \in \mathbb{R}^3$$

donde los vectores $q(t)$ y $\dot{q}(t)$ se forman por $q(t) = [q_1 \ q_2 \ q_3]^T$ y $\dot{q}(t) = [\dot{q}_1 \ \dot{q}_2 \ \dot{q}_3]^T$

Posteriormente, se derivan estas ecuaciones con respecto al tiempo, para obtener las ecuaciones de estado del sistema.

$$\left. \begin{aligned} \frac{d}{dt} q(t) &= \dot{q}(t) \\ \frac{d}{dt} \dot{q}(t) &= \ddot{q}(t) \end{aligned} \right\} \quad (3.38)$$

donde $\dot{q}_1(t)$ y $\ddot{q}(t)$ se despejan de (3.31), y las ecuaciones se definen en función de las variables de estado, obteniendo así las ecuaciones de estado del robot manipulador

$$\frac{d}{dt} \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} \dot{q}(t) \\ M^{-1}(q)[-C(q, \dot{q})\dot{q} - g(q) - f(\dot{q} + \tau(t) + w(t))] \end{bmatrix} \quad (3.39)$$

En la Tabla 3.2 se pueden apreciar las magnitudes físicas del robot Niryo One, los cuales serán aplicados para la simulación del modelado y control del mismo.

Descripción	Notación	Valor	Unidad
Longitud del enlace 1	l_1	0.16048	m
Longitud del enlace 2	l_2	0.297	m
Masa del enlace 1	m_1	1	kg
Masa del enlace 2	m_2	1.631	kg
Inercia 1	I_1	4.676×10^{-3}	kgm^2
Inercia 2	I_2	1.468×10^{-2}	kgm^2
Inercia 3	I_3	7.468×10^{-3}	kgm^2
Coefficiente de fricción 1	α_{11}	0.01	Nm
Coefficiente de fricción 2	α_{22}	0.02	Nm
Coefficiente de fricción 3	α_{33}	0.015	Nm
Aceleración gravitacional	g	9.8	m/s^2

Tabla 3.2 Magnitudes físicas del robot manipulador Niryo One

Es importante tomar en cuenta que los robots manipuladores cuentan con un rango de movilidad según su estructura, y en este caso, el robot Niryo One presenta las siguientes limitantes mecánicas en sus articulaciones usadas para este control, vistas en la Tabla 3.3. En la Figura 3.3 se observa el robot y se indican sus articulaciones.

Articulación	Mínimo	Máximo
$q_1(t)$	-175°	175°
$q_2(t)$	-90°	36.7°
$q_3(t)$	-80°	90°

Tabla 3.3 Límites mecánicos de las articulaciones

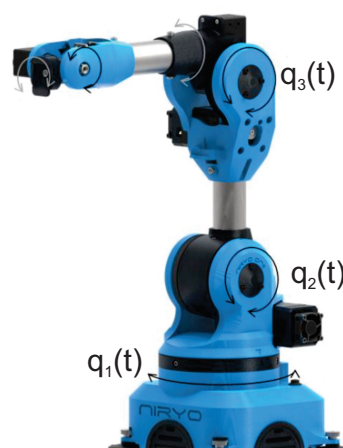


Figura 3.3 Articulaciones del robot Niryo One. Recuperado del manual de especificaciones mecánicas del fabricante.

3.1.4. Simulación y resultados del modelo continuo en MatLab/Simulink del robot Niryo One

Al tener los valores de los parámetros del robot, se realiza la simulación del modelo dinámico del robot en Simulink (ver Figura 3.4), asignando estos valores en código de Matlab. En esta simulación se observa el comportamiento dinámico de las articulaciones al tener un par cero, que se interpreta como una ausencia de par aplicado en las articulaciones, dependiendo solamente de las condiciones iniciales.

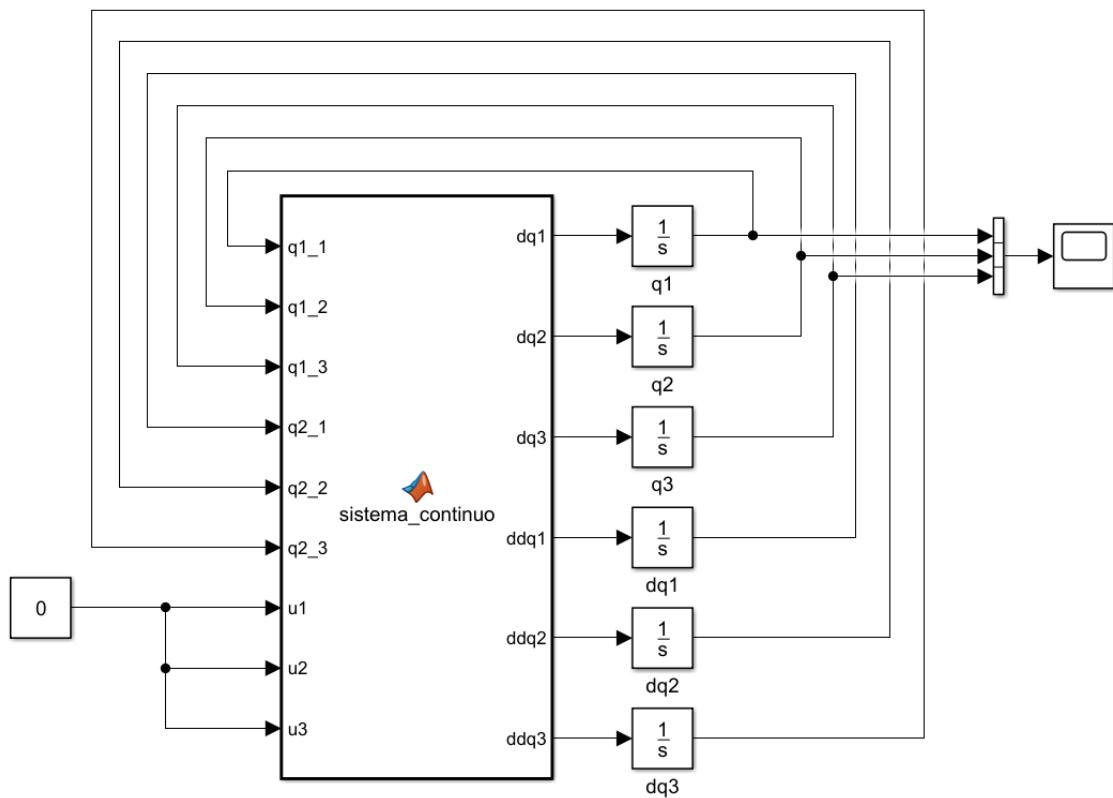


Figura 3.4 Diagrama de bloques en Simulink del modelado del robot Niryo One

La simulación de este modelado proporciona la gráfica vista en la Figura 3.5, donde se observan los desplazamientos angulares de las articulaciones en el transcurso del tiempo.

La articulación $q_1(t)$ no depende de la gravedad, ya que ésta se encuentra sobre la base del robot manipulador y carece del término altura (h) en la ecuación de energía cinética (3.8), es por esto que el desplazamiento angular al transcurrir el tiempo se mantiene en su condición inicial (0 rad), pues su movimiento es solamente horizontal.

El eslabón 1 de la articulación $q_2(t)$ tiene un movimiento sobre el plano xz y ésta sí depende de la gravedad, por esto es que la gráfica muestra una respuesta oscilante partiendo de la condición inicial de 0 rad , hasta llegar y mantenerse en $-\frac{\pi}{2} \text{ rad}$ (su punto de equili-

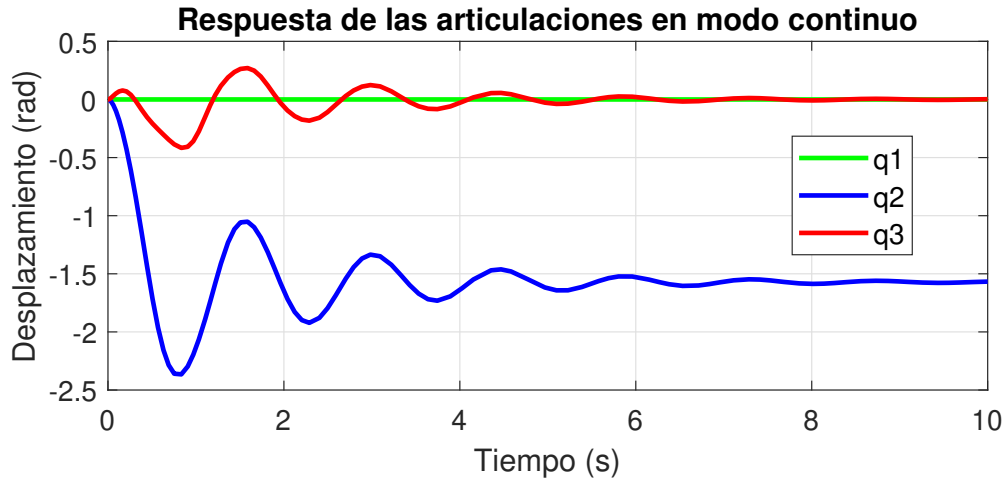


Figura 3.5 Desplazamiento angular de las articulaciones respecto al tiempo, del robot en modo continuo y lazo abierto

brio estable) por efecto de la constante de aceleración gravitacional y la fricción viscosa $f(\dot{q})$.

La articulación $q_3(t)$ tiene como eje de referencia al eslabón l_1 , entonces al partir $q_2(t)$ desde 0 rad (condición inicial), $q_3(t)$ lo hace desde 0 rad (condición inicial en referencia a q_2) y muestra también una respuesta oscilante hasta llegar a su punto de equilibrio estable en 0 rad (alineado con el eje de q_2) por efecto de la fricción viscosa y la constante de aceleración gravitacional aplicada a las articulaciones $q_2(t)$ y $q_3(t)$.

3.2. Discretización del modelo dinámico del robot Niryo One.

En la actualidad existen controles con un procesamiento complejo, lo cual lleva al uso de computadoras y controles digitales más potentes para procesar los algoritmos de control. Por esta razón, los controles son discretizados con una aproximación aceptable al control continuo.

Existen registros de algunos modelos discretos de robots, como Tomizuka, Horowitz y Landau en [39] discretizaron la ecuación de movimiento rechazando términos no lineales. Nicosia y Tomei en [24] usaron la discretización hacia adelante de Euler de la ecuación de movimiento completa. Monaco y Normand-Cyrot en [22] se basaron en la expansión de la solución de ecuaciones diferenciales con la serie de Volterra.

3.2.1. Discretización del modelo continuo del robot Niryo One

Para la discretización del modelo dinámico del robot manipulador, se realiza un cambio de variable de la siguiente forma como se encuentra en [27]:

$$t \rightarrow (kT - 1)$$

donde $T = 0.001s$ es el tiempo de muestreo y k es la k -ésima muestra, al redefinir los argumentos de la posición y velocidad angular, se tiene lo siguiente

$$\left. \begin{aligned} q(t) &\approx q(k-1) \\ \dot{q}(t) &\approx \frac{q(k) - q(k-1)}{T} \end{aligned} \right\} \quad (3.40)$$

Posteriormente, se definen los estados discretos $x_1(k-1)$ y $x_2(k-1)$ en función de la posición y velocidad articular, de la forma:

$$x_1(k-1) = q(k-1) \quad (3.41)$$

Para encontrar la ecuación del estado x_2 en modo discreto, se desarrolla con la aproximación numérica de Euler hacia atrás sobre el estado x_1

$$x_2(k-1) = \frac{x_1(k) - x_1(k-1)}{T} \quad (3.42)$$

Se procede a despejar el estado x_1 en la muestra k de (3.42), de la forma:

$$x_1(k) = x_2(k-1)T + x_1(k-1) \quad (3.43)$$

La derivada del estado discreto x_2 es la aceleración angular de la articulación, por lo cual se utiliza la aproximación numérica de Euler a dicho estado, quedando de la forma:

$$\ddot{q}(t) \approx \frac{x_2(k) - x_2(k-1)}{T} \quad (3.44)$$

La derivada numérica de x_2 es igual a la discretización de (3.39) obteniendo:

$$\begin{aligned} \frac{x_2(k) - x_2(k-1)}{T} &= M^{-1}(x_1(k-1))[\tau(k-1) + w(k-1) \\ &\quad - C(x_1(k), x_2(k-1))x_2(k-1) - g(x_1(k-1)) - f(x_2(k-1))] \end{aligned} \quad (3.45)$$

Para obtener el estado x_2 en la muestra k (presente), se despeja de (3.45) teniendo la forma

$$\begin{aligned} x_2(k) &= TM^{-1}(x_1(k-1))[\tau(k-1) + w(k-1) - C(x_1(k-1), x_2(k-1))x_2(k-1) \\ &\quad - g(x_1(k-1)) - f(x_2(k-1))] + x_2(k-1) \end{aligned} \quad (3.46)$$

Partiendo de la definición de los estados (3.43) y (3.46) y aplicando el método de Euler, se realiza un corrimiento de una muestra hacia adelante de las ecuaciones mencionadas, de la

forma:

$$\left. \begin{aligned} x_1(k+1) &= Tx_2(k) + x_1(k) \\ x_2(k+1) &= TM^{-1}(x_1(k))[\tau(k) + w(k) - C(x_1(k), x_2(k))x_2(k) \\ &\quad - g(x_1(k)) - f(x_2(k))] + x_2(k) \end{aligned} \right\} \quad (3.47)$$

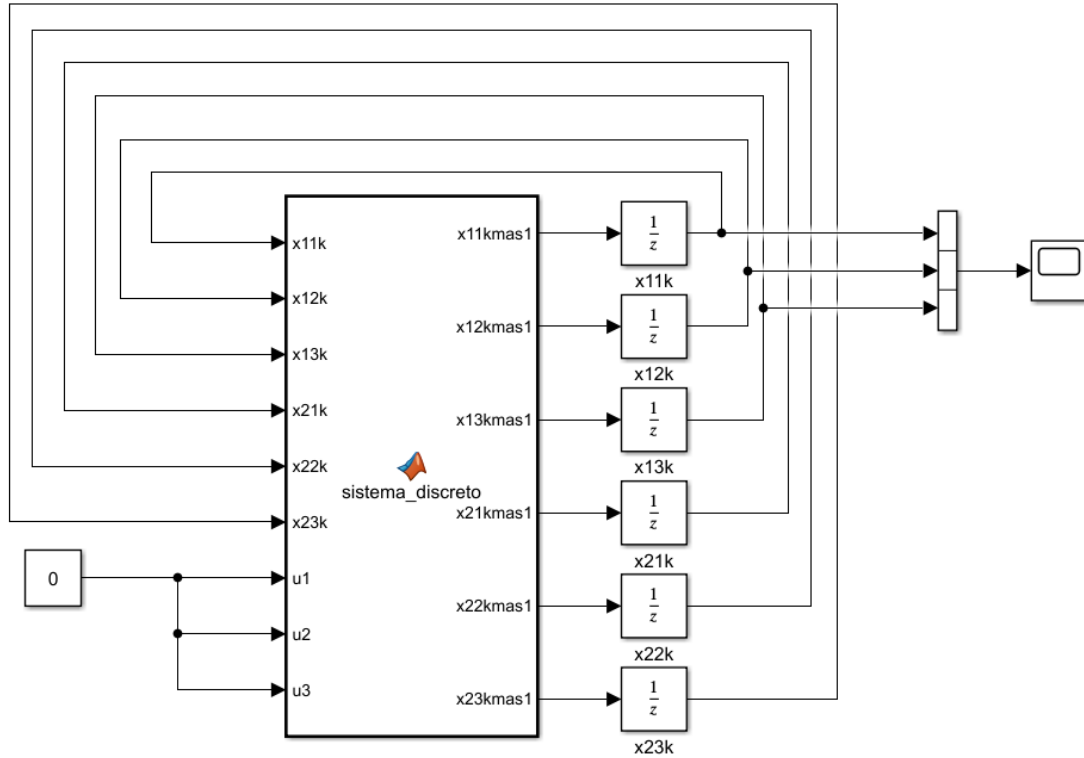


Figura 3.6 Diagrama de bloques en Simulink de la discretización del modelo dinámico del robot manipulador

3.2.2. Simulación y resultados del modelo discreto en MatLab/Simulink del robot Niryo One

La simulación de la representación en espacios de estados discretos del robot en mención (Figura 3.6), por medio de Matlab/Simulink, produce la gráfica vista en la Figura 3.7, donde se observan los desplazamientos angulares de las articulaciones en el transcurso del tiempo discreto cada k muestras.

Es así como el comportamiento dinámico del robot en modelo discreto queda validado, ya que las gráficas arrojadas por la simulación en modelo continuo de la Figura 3.5 y discreto de la Figura 3.7 presentan comportamientos similares.

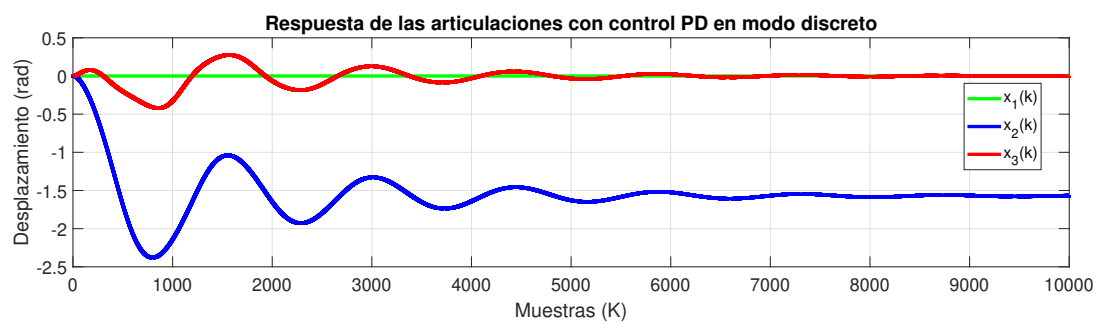


Figura 3.7 Desplazamiento angular de las articulaciones respecto al sistema en lazo abierto del modo discreto del robot de 3 GDL

Capítulo 4

Análisis y diseño en tiempo continuo del controlador PD no lineal con compensación de trayectoria deseada aplicado al robot Niryo One

En la literatura existen diferentes estrategias de control para llevar la posición del efector final siguiendo una trayectoria deseada, como el PID [4], superficies deslizantes [34] o redes neuronales [16], entre otros. A continuación se plantea el esquema del control PD no lineal.

Partiendo del modelo dinámico del robot manipulador de tres GDL descrito en (3.31) se procede a proponer una ley de control por retroalimentación de estados no lineales con la finalidad de resolver el problema de seguimiento de trayectorias.

$$\tau(t) = M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + g(q_d) + f(\dot{q}_d) - K_p\tilde{q} - K_v\dot{\tilde{q}} \quad (4.1)$$

donde K_p y $K_v \in \mathbb{R}^{3 \times 3}$, son matrices diagonales y definidas positivas que forman parte del control PD y el resto del controlador es la compensación de la dinámica del robot en la trayectoria deseada, donde $q_d(t) \in \mathbb{R}^3$ es la posición angular deseada definido por $q_d(t) = [q_{d1} \ q_{d2} \ q_{d3}]^T$, $\dot{q}_d \in \mathbb{R}^3$ es la velocidad angular de referencia y el error de posición y velocidad se definen en seguida:

$$\left. \begin{aligned} \tilde{q}(t) &= q(t) - q_d(t) \\ \dot{\tilde{q}}(t) &= \dot{q}(t) - \dot{q}_d(t) \end{aligned} \right\} \quad (4.2)$$

La representación en espacios de estados en lazo cerrado del robot manipulador (3.31) y el controlador (4.1) se definen en función de los nuevos estados del sistema (4.2), quedando de la forma:

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{q}} \\ \ddot{\tilde{q}} - \ddot{q}_d \end{bmatrix} \quad (4.3)$$

Despejando \ddot{q} de la ecuación (3.31), y sustituyendo el controlador (4.1), se tiene lo siguiente:

$$\ddot{q} = M^{-1}(q)[-C(q, \dot{q})\dot{q} - g(q) - f(\dot{q}) + w(t) + M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + g(q_d) + f(\dot{q}_d) - K_p\tilde{q} - K_v\dot{\tilde{q}}] \quad (4.4)$$

La ecuación anterior es sustituida en el estado $\frac{d}{dt}\dot{\tilde{q}}$, obteniendo así

$$\frac{d}{dt}\dot{\tilde{q}} = -\ddot{q}_d + M^{-1}(q)[-C(q, \dot{q})\dot{q} - g(q) - f(\dot{q}) + w(t) + M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + g(q_d) + f(\dot{q}_d) - K_p\tilde{q} - K_v\dot{\tilde{q}}] \quad (4.5)$$

Al sustituir los nuevos estados (4.2) en (4.5), se obtiene la ecuación de estados en lazo cerrado, representado por el robot manipulador y el controlador propuesto:

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{q}} \\ M^{-1}(\tilde{q} + q_d)[-M(\tilde{q} + q_d) - C(\tilde{q} + q_d, \dot{\tilde{q}} + \dot{q}_d)(\dot{\tilde{q}} + \dot{q}_d) - g(\tilde{q} + q_d) - f(\dot{\tilde{q}} + \dot{q}_d) + M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + g(q_d) + f(\dot{q}_d) + w(t) - K_p\tilde{q} - K_v\dot{\tilde{q}}] \end{bmatrix} \quad (4.6)$$

La dinámica residual se define como la diferencia entre la dinámica real del robot y la dinámica deseada, la cual se obtiene de (4.6) y queda de la forma:

$$h(t, \tilde{q}, \dot{\tilde{q}}) = [M(q_d) - M(\tilde{q} + q_d)]\ddot{q}_d + [C(q_d, \dot{q}_d) - C(\tilde{q} + q_d, \dot{\tilde{q}} + \dot{q}_d)]\dot{q}_d + g(\dot{q}_d) - g(\tilde{q} + q_d) + f(\dot{q}_d) - f(\dot{\tilde{q}} + \dot{q}_d) \quad (4.7)$$

De esta manera, las ecuaciones de estados en lazo cerrado entre el robot manipulador y el controlador propuesto, quedan expresadas a continuación:

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{q}} \\ M^{-1}(\tilde{q} + q_d)[-K_p\tilde{q} - K_v\dot{\tilde{q}} - C(\tilde{q} + q_d, \dot{\tilde{q}} + \dot{q}_d)\dot{\tilde{q}} + h(t, \tilde{q}, \dot{\tilde{q}}) + w(t)] \end{bmatrix} \quad (4.8)$$

4.1. Objetivo de control

El problema de seguimiento de trayectorias del robot manipulador (4.8) se establece de la siguiente manera: dado un conjunto acotado de funciones $q_d(t), \dot{q}_d(t), \ddot{q}_d(t) \in \mathbb{R}^3$ referidos como posiciones, velocidades y aceleraciones articulares de referencia, respectivamente; el objetivo de control consiste en asegurar la estabilidad asintótica en forma global del punto de equilibrio del sistema en lazo cerrado, es decir

$$\lim_{t \rightarrow \infty} \|\tilde{q}(t)\| = 0 \quad (4.9)$$

La ecuación (4.9) se satisface para cualquier condición inicial arbitraria $q(0)$ en un sistema libre de perturbaciones.

4.2. Análisis de existencia y unicidad del punto de equilibrio

Para poder determinar los puntos de equilibrio, las ecuaciones de estado en lazo cerrado (4.8) y el vector de perturbación acoplado, se igualan a cero

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = 0 \quad (4.10)$$

$$\left. \begin{aligned} \dot{\tilde{q}} &= 0 \\ M^{-1}(\tilde{q} + q_d)[-K_p \tilde{q} - K_v \dot{\tilde{q}} + h(t, \tilde{q}, \dot{\tilde{q}})] &= 0 \end{aligned} \right\} \quad (4.11)$$

De manera directa se conoce el estado $\dot{\tilde{q}} = 0$, y al ser sustituido en la ecuación de $\frac{d}{dt} \tilde{q}$ se obtiene $[-K_p \tilde{q} + h(t, \tilde{q}, \dot{\tilde{q}})] = 0$. Para obtener el segundo estado que conforma el punto de equilibrio $(\tilde{q}, \dot{\tilde{q}})$, se despeja \tilde{q} , obteniendo la siguiente ecuación:

$$\varphi(\tilde{q}^*) = K_p^{-1} h(t, \tilde{q}^*, 0) \quad (4.12)$$

Se continúa con el análisis de unicidad del punto de equilibrio por medio de la desigualdad de Lipschitz

$$\|\varphi(x) - \varphi(y)\| \leq L\|x - y\| \quad (4.13)$$

Sustituyendo (4.12) en (4.13) se obtiene que

$$\|\varphi(x) - \varphi(y)\| = \|K_p^{-1}[h(t, x, 0) - h(t, y, 0)]\| \leq \|K_p^{-1}\| \|h(t, x, 0) - h(t, y, 0)\| \quad (4.14)$$

Continuando con este análisis, se procede a evaluar las normas de cada término que compone a (4.14), iniciando con la dinámica residual.

$$\begin{aligned} \|h(t, x, 0) - h(t, y, 0)\| &= \| [M(q_d - y)\ddot{q}_d - M(q_d - x)\ddot{q}_d] \\ &+ \| [C(q_d - y, \dot{q}_d) - C(q_d, \dot{q}_d)]\dot{q}_d \| + \| [g(q_d - y) - g(q_d - x)] \| + \| f(\dot{q}_d) - f(\dot{\tilde{q}} + \dot{q}_d) \| \end{aligned} \quad (4.15)$$

Propiedades de matrices y vectores

Para el análisis de estabilidad y unicidad en el punto de equilibrio, se usa la desigualdad de Lipchitz $\|\varphi(x) - \varphi(y)\| \leq L\|x - y\|$, donde es necesario primero establecer las siguientes normas propuestas en [13], que son propiedades que se utilizan en el estudio de robots manipuladores para determinar las cotas máximas. Dichas propiedades son aplicadas a las matrices y vectores que son los parámetros del robot manipulador.

$$\|M(x)z - M(y)z\| \leq k_M\|x - y\|\|z\| \quad (4.16)$$

$$\|C(x, z)w - C(y, v)w\| \leq k_{C_1}\|z - v\|\|w\| + k_{C_2}\|x - y\|\|w\|\|z\| \quad (4.17)$$

$$\|g(x) - g(y)\| \leq k_g\|x - y\| \quad (4.18)$$

$$\left\|\frac{1}{K_p}\right\| \leq \lambda_{\min}\{K_p\} \quad (4.19)$$

Primero se evalúa la matriz de inercia usando la propiedad (4.16)

$$[M(q_d) - M(x + q_d)]\ddot{q}_d - [M(q_d) - M(y + q_d)]\ddot{q}_d \quad (4.20)$$

y al eliminar términos semejantes, se llega a la inecuación

$$\|M(x + q_d)\ddot{q}_d - M(y + q_d)\ddot{q}_d\| \leq k_M\|x - y\|\|\ddot{q}_d\| \quad (4.21)$$

Se realiza lo mismo con la matriz de fuerzas centrífugas y de Coriolis usando la propiedad (4.17)

$$[C(q_d, \dot{q}_d) - C(x + q_d, 0 + \dot{q}_d)]\dot{q}_d - [C(q_d, \dot{q}_d) - C(y + q_d, \dot{q}_d)]\dot{q}_d \quad (4.22)$$

y al eliminar términos semejantes, se llega a la inecuación

$$\|C(x + q_d, \dot{q}_d)\dot{q}_d - C(y + q_d, \dot{q}_d)\dot{q}_d\| \leq k_{C_2}\|\dot{q}_d\|^2\|x - y\| \quad (4.23)$$

Posteriormente con la matriz de fuerzas gravitacionales usando la propiedad (4.18)

$$\|g(q_d) - g(x + q_d) - g(q_d) + g(y + q_d)\| \quad (4.24)$$

y al eliminar términos semejantes, se llega a la inecuación

$$\|g(x + q_d) - g(y + q_d)\| \leq k_g\|x - y\| \quad (4.25)$$

Por último, se evalúa la matriz de fricción

$$\|f(0 + \dot{q}_d) + f(\dot{q}_d) - f(0 + \dot{q}_d) - f(\dot{q}_d)\| = 0 \quad (4.26)$$

en la ecuación anterior se logran eliminar todos los términos.

Usando las desigualdades de (4.21), (4.23), (4.25) y (4.26) en la desigualdad de Lipschitz(4.13) como se muestra enseguida, se llega a la inecuación

$$\|\varphi(x) - \varphi(y)\| \leq \frac{1}{\lambda_{\min}\{K_p\}} [k_M \|\ddot{q}_d\| + k_{C_2} \|\dot{q}_d\|^2 + k_g] \|x - y\| \quad (4.27)$$

al igualar la (4.27) con la desigualdad de Lipschitz, tenemos que su constante $L = \frac{1}{\lambda_{\min}\{K_p\}} [k_M \|\ddot{q}_d\| + k_{C_2} \|\dot{q}_d\|^2 + k_g]$. Si $L < 1$, es una condición suficiente para la existencia y unicidad del punto de equilibrio. Analizando los parámetros de la constante de Lipschitz, se observa que el único parámetro independiente de la manufactura del robot y de la señal de referencia es el control proporcional, por lo tanto se despeja de la desigualdad, quedando que $\frac{1}{\lambda_{\min}\{K_p\}} [k_M \|\ddot{q}_d\| + k_{C_2} \|\dot{q}_d\|^2 + k_g] < 1$ y para que esta condición pueda ser cumplida, $\lambda_{\min}\{K_p\}$ queda de la siguiente manera:

$$\lambda_{\min}\{K_p\} > k_M \|\ddot{q}_d\| + k_{C_2} \|\dot{q}_d\|^2 + k_g \quad (4.28)$$

por lo tanto, si se cumple (4.28), es una condición suficiente para demostrar existencia y unicidad del punto de equilibrio del robot Niryo One.

4.3. Análisis de estabilidad asintótica en forma global

Una vez demostrada la condición de existencia y unicidad del punto de equilibrio, se continúa con el análisis de estabilidad en el sentido de Lyapunov para establecer las condiciones suficientes para alcanzar la estabilidad asintótica en forma global del sistema en lazo cerrado. Para esto, se utiliza la función candidata de Lyapunov propuesta por Kelly [13]

$$V(t, \tilde{q}, \dot{\tilde{q}}) = \frac{1}{2} \dot{\tilde{q}}^T M(\tilde{q} + q_d) \dot{\tilde{q}} + \frac{1}{2} \tilde{q}^T K_p \tilde{q} + \gamma \tanh(\tilde{q})^T M(\tilde{q} + q_d) \dot{\tilde{q}} \quad (4.29)$$

Las condiciones que debe cumplir la función (4.29) para que sea considerada una función candidata, son las siguientes:

$V(0) = 0$, condición que se cumple,

$V(t, \tilde{q}, \dot{\tilde{q}}) > 0 \quad \forall t, \tilde{q}, \dot{\tilde{q}} \neq 0$. Esta condición se logra cumplir solo con el primer y segundo término de la función (4.29).

Para determinar si la función de Lyapunov (4.29) es definida positiva para cualquier valor de $t, \tilde{q}, \dot{\tilde{q}}$ diferente de cero, es necesario realizar un análisis para determinar la cota mínima de cada término que la compone, para lo cual se propone seccionar la función de Lyapunov.

El término $\frac{1}{2} \|\dot{\tilde{q}}^T M(\tilde{q} + q_d) \dot{\tilde{q}}\|$ no presenta problemas, pues al tener un producto del mismo vector y la matriz M ser definida positiva, éste término resultará positivo.

El término $\frac{1}{2} \|\tilde{q}^T K_p \tilde{q}\|$ tampoco presenta problemas, pues se tiene de nuevo un producto del vector del error por su transpuesta y ser K_p definida positiva, teniendo así valores positivos.

El problema existe con el término $\gamma \tanh(\tilde{q})^T M(\tilde{q} + q_d) \dot{\tilde{q}}$, pues al tener producto cruzado de vectores de estados, no se puede garantizar que la función sea definida positiva. Para poder hacer el análisis, se hace uso de las propiedades de tangente hiperbólica siguientes:

Para cualquier $x, \dot{x} \in \mathbb{R}^n$

- $\|\tanh(x)\| \leq \alpha_1 \|x\|$
- $\|\tanh(x)\| \leq \alpha_2$
- $\|\tanh(x)\|^2 \leq \alpha_3 \tanh(x)^T x$
- $\|\operatorname{sech}^2(x) \dot{x}\| \leq \alpha_4 \|\dot{x}\|$

donde $\alpha_1, \dots, \alpha_4 > 0$. Con $\tanh(x)$ de la forma vectorial que se ve en la siguiente ecuación, y las constantes $\alpha_1 = 1$, $\alpha_2 = \sqrt{n}$, $\alpha_3 = 1$, $\alpha_4 = 1$

$$\tanh(x) := \begin{bmatrix} \tanh(x_1) \\ \vdots \\ \tanh(x_n) \end{bmatrix}.$$

Para el análisis de la cota mínima, se aplican normas a la ecuación (4.29), obteniendo así la expresión

$$V(t, \tilde{q}, \dot{\tilde{q}}) \geq \frac{1}{2} \|\dot{\tilde{q}}^T M(\tilde{q} + q_d) \dot{\tilde{q}}\| + \frac{1}{2} \|\tilde{q}^T K_p \tilde{q}\| + \gamma \|\tanh(\tilde{q})^T M(\tilde{q} + q_d) \dot{\tilde{q}}\| \quad (4.30)$$

Iniciando con la norma de la matriz de inercia:

$$\begin{aligned} \frac{1}{2} \|\dot{\tilde{q}}^T M(\tilde{q} + q_d) \dot{\tilde{q}}\| &\leq \frac{1}{2} \|M(\tilde{q} + q_d)\| \|\dot{\tilde{q}}\|^2 \\ \frac{1}{2} \|\dot{\tilde{q}}^T M(\tilde{q} + q_d) \dot{\tilde{q}}\| &\leq \frac{1}{2} \lambda_{\max}\{M\} \|\dot{\tilde{q}}\|^2, \end{aligned} \quad (4.31)$$

posteriormente, la matriz de ganancia proporcional

$$\begin{aligned} \frac{1}{2} \|\tilde{q}^T K_p \tilde{q}\| &\leq \frac{1}{2} \|K_p\| \|\tilde{q}\|^2 \\ \frac{1}{2} \|\tilde{q}^T K_p \tilde{q}\| &\leq \frac{1}{2} \lambda_{\min}\{K_p\} \|\tilde{q}\|^2, \end{aligned} \quad (4.32)$$

y por último, el tercer término de la función de Lyapunov

$$\begin{aligned} \gamma \tanh(\tilde{q})^T M(\tilde{q} + q_d) \dot{\tilde{q}} &\leq \gamma \|\tanh(\tilde{q})\| \|M(\tilde{q} + q_d) \dot{\tilde{q}}\| \\ \gamma \tanh(\tilde{q})^T M(\tilde{q} + q_d) \dot{\tilde{q}} &\leq \gamma \alpha_1 \lambda_{\max}\{M\} \|\tilde{q}\| \|\dot{\tilde{q}}\| \\ -\gamma \tanh(\tilde{q})^T M(\tilde{q} + q_d) \dot{\tilde{q}} &\geq -\gamma \alpha_1 \lambda_{\max}\{M\} \|\tilde{q}\| \|\dot{\tilde{q}}\| \end{aligned} \quad (4.33)$$

Uniando las desigualdades de las ecuaciones (4.31, 4.32 y 4.33), se llega a la siguiente desigualdad:

$$V(t, \tilde{q}, \dot{\tilde{q}}) \geq \frac{1}{2} \lambda_{\max}\{M\} \|\dot{\tilde{q}}\|^2 + \frac{1}{2} \lambda_{\min}\{K_p\} \|\tilde{q}\|^2 - \gamma \alpha_1 \lambda_{\max}\{M\} \|\tilde{q}\| \|\dot{\tilde{q}}_d\| \quad (4.34)$$

Se procede a escribir la expresión anterior en su forma matricial

$$V(t, \tilde{q}, \dot{\tilde{q}}) \geq \frac{1}{2} \begin{bmatrix} \|\tilde{q}\| \\ \|\dot{\tilde{q}}\| \end{bmatrix}^T [Q] \begin{bmatrix} \|\tilde{q}\| \\ \|\dot{\tilde{q}}\| \end{bmatrix} \quad (4.35)$$

donde se define a la matriz Q como:

$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} = \begin{bmatrix} \lambda_{\min}\{K_p\} & -\gamma \alpha_1 \lambda_{\max}\{M\} \\ -\gamma \alpha_1 \lambda_{\max}\{M\} & \lambda_{\min}\{M\} \end{bmatrix}$$

Para determinar las condiciones bajo las cuales $Q > 0$, se emplea el teorema de Silvester [13] que establece que la matriz Q es definida positiva si y solo si:

$$\begin{aligned} q_{11} &> 0 \\ \det[Q] &> 0 \end{aligned}$$

cumpliendo de esta manera con $\lambda_{\min}\{K_p\} > 0$ y $\lambda_{\min}\{K_p\} \lambda_{\min}\{M\} - \gamma^2 \alpha_1^2 \lambda_{\max}^2\{M\} > 0$, y es a partir de esta desigualdad, que se despeja el valor propio mínimo de K_p

$$\lambda_{\min}\{K_p\} > \frac{\gamma^2 \alpha_1^2 \lambda_{\max}^2\{M\}}{\lambda_{\min}\{M\}} \quad (4.36)$$

Si $\lambda_{\min}\{K_p\} > 0$ y la desigualdad (4.36) se satisface entonces $Q > 0$ y por lo tanto $V(t, \tilde{q}, \dot{\tilde{q}}) > 0$

La tercera condición de Lyapunov es $\dot{V}(x) = \frac{\partial}{\partial x} V(t, \tilde{q}, \dot{\tilde{q}}) \dot{x} < 0$ (definida negativa) aplicada a la función candidata de Lyapunov (4.29) sobre las trayectorias del sistema en lazo cerrado (4.8), se debe considerar que el vector de perturbaciones es igual a cero

$$\begin{aligned} \dot{V} = \dot{\tilde{q}}^T M(\tilde{q} + q_d) \ddot{\tilde{q}} + \frac{1}{2} \dot{\tilde{q}}^T \dot{M}(\tilde{q} + q_d) \dot{\tilde{q}} + \dot{\tilde{q}}^T K_p \dot{\tilde{q}} + \gamma \operatorname{sech}^2(\tilde{q})^T \dot{\tilde{q}} M(\tilde{q} + q_d) \dot{\tilde{q}} \\ + \gamma \tanh(\tilde{q})^T \dot{M}(\tilde{q} + q_d) \dot{\tilde{q}} + \gamma \tanh(\tilde{q})^T M(\tilde{q} + q_d) \ddot{\tilde{q}} \end{aligned} \quad (4.37)$$

donde se sustituye la ecuación de estados de la dinámica del error (4.8)

$$\begin{aligned} \dot{V} = \dot{\tilde{q}}^T [-K_p \tilde{q} - K_v \dot{\tilde{q}} - C(\tilde{q} + q_d, \dot{\tilde{q}} + \dot{q}_d) \dot{\tilde{q}} + h(t, \tilde{q}, \dot{\tilde{q}})] \\ + \frac{1}{2} \dot{\tilde{q}}^T \dot{M}(\tilde{q} + q_d) \dot{\tilde{q}} + \dot{\tilde{q}}^T K_p \dot{\tilde{q}} + \gamma \operatorname{sech}^2(\tilde{q})^T \dot{\tilde{q}} M(\tilde{q} + q_d) \dot{\tilde{q}} + \gamma \tanh(\tilde{q})^T \dot{M}(\tilde{q} + q_d) \dot{\tilde{q}} \\ + \gamma \tanh(\tilde{q})^T [-K_p \tilde{q} - K_v \dot{\tilde{q}} - C(\tilde{q} + q_d, \dot{\tilde{q}} + \dot{q}_d) \dot{\tilde{q}} + h(t, \tilde{q}, \dot{\tilde{q}})] \end{aligned} \quad (4.38)$$

Después de obtener los productos algebraicos de los términos y eliminar los semejantes, se llega a la derivada de la función de Lyapunov

$$\begin{aligned} \dot{V} = & -\dot{\tilde{q}}^T K_v \dot{\tilde{q}} + \gamma \dot{\tilde{q}}^T \text{sech}^2(\tilde{q})^T M(\tilde{q} + q_d) \dot{\tilde{q}} - \gamma \tanh(\tilde{q})^T K_p \tilde{q} + \dot{\tilde{q}}^T h(t, \tilde{q}, \dot{\tilde{q}}) \\ & + \gamma \tanh(\tilde{q})^T \dot{M}(\tilde{q} + q_d) \dot{\tilde{q}} - \gamma \tanh(\tilde{q})^T K_v \dot{\tilde{q}} - \gamma \tanh(\tilde{q})^T C(\tilde{q} + q_d, \dot{\tilde{q}} + \dot{q}_d) \dot{\tilde{q}} \end{aligned} \quad (4.39)$$

De la expresión anterior se pueden simplificar algunos términos a partir de la propiedad siguiente:

$$\dot{M}(\tilde{q} + q_d) = C(\tilde{q} + q_d, \dot{\tilde{q}} + \dot{q}_d) + C^T(\tilde{q} + q_d, \dot{\tilde{q}} + \dot{q}_d)$$

Es así como al aplicar la propiedad y eliminar algunos términos, la expresión queda representada como:

$$\begin{aligned} \dot{V} = & -\dot{\tilde{q}}^T K_v \dot{\tilde{q}} + \gamma \dot{\tilde{q}}^T \text{sech}^2(\tilde{q})^T M(\tilde{q} + q_d) \dot{\tilde{q}} - \gamma \tanh(\tilde{q})^T K_p \tilde{q} + \dot{\tilde{q}}^T h(t, \tilde{q}, \dot{\tilde{q}}) \\ & + \gamma \tanh(\tilde{q})^T C(\tilde{q} + q_d, \dot{\tilde{q}} + \dot{q}_d) \dot{\tilde{q}} - \gamma \tanh(\tilde{q})^T K_v \dot{\tilde{q}} \end{aligned} \quad (4.40)$$

Para buscar que $\dot{V} \leq 0$, se aplican las normas a cada término de la ecuación anterior, convirtiendo a (4.40) en una desigualdad mostrada a continuación:

$$\begin{aligned} \dot{V} = & -\|\dot{\tilde{q}}^T K_v \dot{\tilde{q}}\| + \|\gamma \dot{\tilde{q}}^T \text{sech}^2(\tilde{q})^T M(\tilde{q} + q_d) \dot{\tilde{q}}\| - \|\gamma \tanh(\tilde{q})^T K_p \tilde{q}\| + \|\dot{\tilde{q}}^T h(t, \tilde{q}, \dot{\tilde{q}})\| \\ & + \|\gamma \tanh(\tilde{q})^T C(\tilde{q} + q_d, \dot{\tilde{q}} + \dot{q}_d) \dot{\tilde{q}}\| - \|\gamma \tanh(\tilde{q})^T K_v \dot{\tilde{q}}\| \end{aligned} \quad (4.41)$$

y se realiza un análisis por segmentos para encontrar las cotas mínimas.

$$-\|\dot{\tilde{q}}^T K_v \dot{\tilde{q}}\| \leq -\lambda_{\min}\{K_v\} \|\dot{\tilde{q}}\|^2 \quad (4.42)$$

$$\gamma \|\dot{\tilde{q}}^T \text{sech}^2(\tilde{q})^T M(\tilde{q} + q_d) \dot{\tilde{q}}\| \leq \gamma \alpha_4 \lambda_{\max}\{M\} \|\dot{\tilde{q}}\|^2 \quad (4.43)$$

$$-\gamma \|\tanh(\tilde{q})^T K_p \tilde{q}\| \leq -\frac{\gamma \lambda_{\min}\{K_p\}}{\alpha_3} \|\tanh(\tilde{q})\|^2 \quad (4.44)$$

Para el término de la dinámica residual, se retoma la propiedad vista en 4.15, por lo tanto se obtiene que

$$\begin{aligned} \|\dot{\tilde{q}}^T h(t, \tilde{q}, \dot{\tilde{q}})\| \leq & \|[M(q_d) - M(\tilde{q} + q_d)]\ddot{q}_d\| + \|[C(q_d, \dot{q}_d) - C(\tilde{q}_d + q_d, \dot{\tilde{q}} + \dot{q}_d)]\dot{q}_d\| \\ & + \|g(q_d) - g(\tilde{q} + q_d)\| + \|f(\dot{q}_d) - f(\dot{\tilde{q}} + \dot{q}_d)\|, \end{aligned}$$

y usando las propiedades 4.16, 4.17 y 4.18, se obtiene lo siguiente:

$$\|M(q_d)\ddot{q}_d - M(\tilde{q} + q_d)\ddot{q}_d\| \leq 2K_M \|\ddot{q}_d\| \quad (4.45)$$

$$\|C(q_d, \dot{q}_d)\dot{q}_d - C(\tilde{q} + q_d, \dot{\tilde{q}} + \dot{q}_d)\dot{q}_d\| \leq K_{C_1} \|\dot{q}_d\| \|\dot{\tilde{q}}\| + K_{C_2} \|\dot{q}_d\|^2 \|\tilde{q}\| \quad (4.46)$$

$$\|g(q_d) - g(\tilde{q} + q_d)\| \leq K_g \|\tilde{q}\| \quad (4.47)$$

Para el término $\|f(\dot{q}_d) - f(\dot{\tilde{q}} + \dot{q}_d)\|$ se define a la matriz $f(\dot{q})$ como $f(\dot{q}) = F_c \dot{q}$, $F_c = F_c^T > 0$, entonces:

$$\|F_c \dot{q}_d - F_c(\ddot{q} + \dot{q}_d)\| \leq \|F_c \dot{q}_d - F_c \ddot{q} - F_c \dot{q}_d\|$$

$$\|F_c \dot{q}_d - F_c(\ddot{q} + \dot{q}_d)\| \leq \lambda_{\max}\{F_c\} \|\ddot{q}\| \quad (4.48)$$

Uniendo todos los términos para formar la norma de la dinámica residual, es que se obtiene lo siguiente:

$$\|h(t, \tilde{q}, \dot{\tilde{q}})\| \leq K_M \|\ddot{q}_d\| \|\tilde{q}\| + K_{C_1} \|\dot{q}_d\| \|\dot{\tilde{q}}\| + K_{C_2} \|\dot{q}_d\|^2 \|\tilde{q}\| + K_g \|\tilde{q}\| + \lambda_{\max}\{F_c\} \|\ddot{q}\|$$

Factorizando los términos semejantes, la norma de la dinámica residual queda como:

$$\|h(t, \tilde{q}, \dot{\tilde{q}})\| \leq [K_{C_1} \|\dot{q}_d\| + \lambda_{\max}\{F_c\}] \|\dot{\tilde{q}}\| + [K_M \|\ddot{q}_d\| + K_{C_2} \|\dot{q}_d\|^2 + K_g] \|\tilde{q}\| \quad (4.49)$$

$$\begin{aligned} K_{h_1} &\geq K_{C_1} \|\dot{q}_d\| + \lambda_{\max}\{F_c\} \\ \lambda_{\max}\{F_c\} &= 0.2 \\ K_{C_1} \|\dot{q}_d\| &= 0.114 \end{aligned}$$

El siguiente término es $\gamma \tanh(\tilde{q})^T C(\tilde{q} + q_d, \ddot{q} + \dot{q}_d) \dot{\tilde{q}}$, por lo tanto, aplicando las normas se tiene que

$$\begin{aligned} \gamma \|\tanh(\tilde{q})^T C(\tilde{q} + q_d, \ddot{q} + \dot{q}_d) \dot{\tilde{q}}\| &\leq \gamma \|\dot{\tilde{q}}\| \|C(\tilde{q} + q_d, \ddot{q} + \dot{q}_d)\| \|\tanh(\tilde{q})\| \\ \gamma \|\tanh(\tilde{q})^T C(\tilde{q} + q_d, \ddot{q} + \dot{q}_d) \dot{\tilde{q}}\| &\leq \gamma K_{C_1} \|\dot{\tilde{q}}\|^2 \|\tanh(\tilde{q})\| + \gamma K_{C_1} \|\dot{q}_d\| \|\dot{\tilde{q}}\| \|\tanh(\tilde{q})\| \\ \gamma \|\tanh(\tilde{q})^T C(\tilde{q} + q_d, \ddot{q} + \dot{q}_d) \dot{\tilde{q}}\| &\leq \gamma K_{C_1} \alpha_2 \|\dot{\tilde{q}}\|^2 + \gamma K_{C_1} \|\dot{q}_d\| \|\dot{\tilde{q}}\| \|\tanh(\tilde{q})\| \end{aligned} \quad (4.50)$$

$$\begin{aligned} \gamma \|\tanh(\tilde{q})^T h(t, \tilde{q}, \dot{\tilde{q}})\| &\leq \gamma \|\tanh(\tilde{q})^T\| \|h(t, \tilde{q}, \dot{\tilde{q}})\| \\ \gamma \|\tanh(\tilde{q})^T h(t, \tilde{q}, \dot{\tilde{q}})\| &\leq \gamma K_{h_1} \|\dot{\tilde{q}}\| \|\tanh(\tilde{q})\| + \gamma K_{h_2} \|\tanh(\tilde{q})\|^2 \end{aligned} \quad (4.51)$$

Y el último término de la ecuación (4.40)

$$-\gamma \|\tanh(\tilde{q})^T K_v \dot{\tilde{q}}\| \leq \frac{\gamma}{\alpha_3} \lambda_{\min}\{K_v\} \|\tanh(\tilde{q})\|^2 \quad (4.52)$$

Uniendo las normas de las ecuaciones (4.42, 4.43, 4.44, 4.50, 4.51, 4.52), quedan de la siguiente manera:

$$\begin{aligned} \dot{V} &\leq -\lambda_{\min}\{K_v\} \|\dot{\tilde{q}}\|^2 + \gamma \alpha_4 \lambda_{\max}\{M\} \|\dot{\tilde{q}}\|^2 - \frac{\gamma \lambda_{\min}\{K_p\}}{\alpha_3} \|\tanh(\tilde{q})\|^2 + \gamma K_{C_1} \alpha_2 \|\dot{\tilde{q}}\|^2 + \\ &\gamma K_{C_1} \|\dot{q}_d\| \|\dot{\tilde{q}}\| \|\tanh(\tilde{q})\| + \gamma K_{h_1} \|\dot{\tilde{q}}\| \|\tanh(\tilde{q})\| + \gamma K_{h_2} \|\tanh(\tilde{q})\|^2 + \frac{\gamma}{\alpha_3} \lambda_{\min}\{K_v\} \|\tanh(\tilde{q})\|^2. \end{aligned} \quad (4.53)$$

La desigualdad (4.53) se organizan en forma matricial como se ve a continuación

$$\dot{V} \leq -\gamma \begin{bmatrix} \|\tanh(\tilde{q})\| \\ \|\dot{\tilde{q}}\| \end{bmatrix} \underbrace{\begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}}_Q \begin{bmatrix} \|\tanh(\tilde{q})\| \\ \|\dot{\tilde{q}}\| \end{bmatrix} \quad (4.54)$$

donde:

$$\begin{aligned} q_{11} &= \frac{1}{\alpha_3} \lambda_{\min}\{K_p\} - \frac{1}{\alpha_3} \lambda_{\min}\{K_v\} - K_{h_2} \\ q_{12} &= -\frac{1}{2} K_{h_2} - \frac{1}{2} K_{h_2} \|\dot{q}_d\| - \frac{1}{2} K_{h_1} \\ q_{21} &= -\frac{1}{2} K_{h_2} - \frac{1}{2} K_{h_2} \|\dot{q}_d\| - \frac{1}{2} K_{h_1} \\ q_{22} &= \frac{1}{\gamma} \lambda_{\min}\{K_v\} - \alpha_4 \lambda_{\max}\{M\} - \frac{K_{h_1}}{\gamma} - K_{C_1} \alpha_2 \end{aligned}$$

Para determinar las condiciones bajo las cuales $Q > 0$, se emplea el teorema de Silvester [13] que establece que la matriz Q es definida positiva si y solo si:

$$\begin{aligned} q_{11} &> 0 \\ \det[Q] &> 0 \end{aligned}$$

obteniendo así la primera condición del diseño del controlador

$$\frac{1}{\alpha_3} \lambda_{\min}\{K_p\} - \frac{1}{\alpha_3} \lambda_{\min}\{K_v\} - K_{h_2} > 0$$

y es a partir de esta desigualdad, que se despeja el valor propio mínimo de K_p

$$\lambda_{\min}\{K_p\} > \lambda_{\min}\{K_v\} + \alpha_3 K_{h_2}. \quad (4.55)$$

Ahora se busca cumplir con la segunda condición, que $\det[Q] > 0$. Por lo tanto, a partir de la determinante $[\frac{1}{\alpha_3} \lambda_{\min}\{K_p\} - \frac{1}{\alpha_3} \lambda_{\min}\{K_v\} - K_{h_2}][\frac{1}{\gamma} \lambda_{\min}\{K_v\} - \alpha_4 \lambda_{\max}\{M\} - \frac{K_{h_1}}{\gamma} - K_{C_1} \alpha_2] - (-\frac{1}{2} K_{h_2} - \frac{1}{2} K_{h_2} \|\dot{q}_d\| - \frac{1}{2} K_{h_1})^2 > 0$ se obtiene la segunda condición:

$$\lambda_{\min}\{K_p\} > \frac{\alpha_3 [(\frac{1}{2} K_{h_2} + \frac{1}{2} K_{h_2} \|\dot{q}_d\| + \frac{1}{2} K_{h_1})^2]}{\frac{1}{\gamma} \lambda_{\min}\{K_v\} - \alpha_4 \lambda_{\max}\{M\} - \frac{K_{h_1}}{\gamma} - K_{C_1} \alpha_2} + \lambda_{\min}\{K_v\} + \alpha_3 K_{h_2} \quad (4.56)$$

siendo necesario que para que se cumpla (4.56), la siguiente expresión sea verdadera:

$$\lambda_{\min}\{K_v\} > \gamma [\alpha_4 \lambda_{\max}\{M\} + \frac{K_{h_1}}{\gamma} + K_{C_1} \alpha_2] \quad (4.57)$$

Diseño del controlador K_p y K_v

El diseño del controlador PD debe cumplir con cada una de las siguientes desigualdades:

$$\lambda_{\min}\{K_p\} > \lambda_{\min}\{K_v\} + \alpha_3 K_{h_2}. \quad (4.58)$$

$$\lambda_{\min}\{K_p\} > \frac{\alpha_3[(\frac{1}{2}K_{h_2} + \frac{1}{2}K_{h_2}\|\dot{q}_d\| + \frac{1}{2}K_{h_1})^2]}{\frac{1}{\gamma}\lambda_{\min}\{K_v\} - \alpha_4\lambda_{\max}\{M\} - \frac{K_{h_1}}{\gamma} - K_{c_1}\alpha_2} + \lambda_{\min}\{K_v\} + \alpha_3K_{h_2} \quad (4.59)$$

$$\lambda_{\min}\{K_p\} > k_M\|\ddot{q}_d\| + k_{C_2}\|\dot{q}_d\|^2 + k_g \quad (4.60)$$

$$\lambda_{\min}\{K_p\} > \frac{\gamma^2\alpha_1^2\lambda_{\max}^2\{M\}}{\lambda_{\min}\{M\}} \quad (4.61)$$

$$\lambda_{\min}\{K_v\} > \gamma[\alpha_4\lambda_{\max}\{M\} + \frac{K_{h_1}}{\gamma} + K_{c_1}\alpha_2] \quad (4.62)$$

En la Figura 4.1 se observa un diagrama del controlador aplicado al robot, con los nuevos estados como entradas.

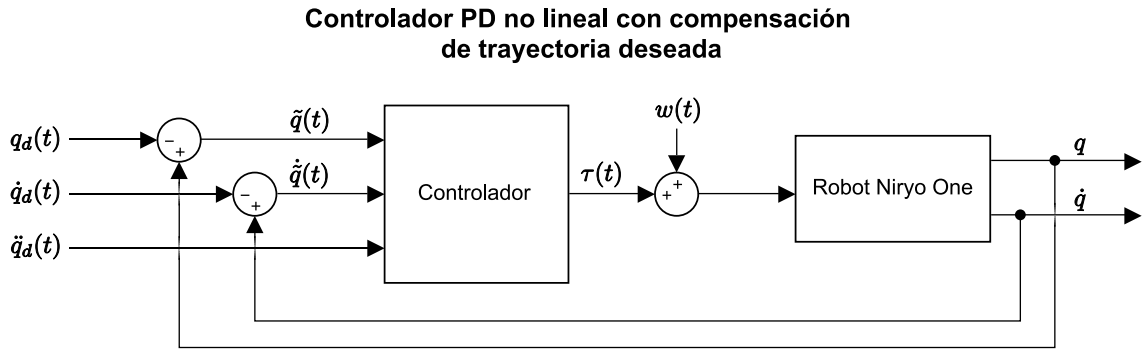


Figura 4.1 Diagrama del controlador PD continuo, aplicado al robot

4.4. Simulación y resultados del sistema continuo en lazo cerrado

La simulación se realizó en MatLab/Simulink ver. 2018, para evaluar el desempeño del sistema continuo en lazo cerrado representado por (4.8), partiendo de las siguientes posiciones iniciales: $q(0) = [\frac{\pi}{2}, -\frac{\pi}{2}, \frac{\pi}{2}]^T$, $\dot{q}(0) = [0 \ 0 \ 0]^T$ y respondiendo a los vectores de las señales de trayectorias deseadas, dado por:

$$q_d(t) = \begin{bmatrix} \frac{\pi}{4}(1 - e^{-2t^3}) + \frac{\pi}{4}(1 - e^{-2t^3})\text{sen}(0.2\pi t) \\ \frac{\pi}{12}(1 - e^{-2t^3}) + \frac{\pi}{12}(1 - e^{-2t^3})\text{sen}(0.4\pi t) \\ \frac{\pi}{20}(1 - e^{-2t^3}) + \frac{\pi}{20}(1 - e^{-2t^3})\text{sen}(0.6\pi t) \end{bmatrix} \quad (4.63)$$

$$\dot{q}_d(t) = \begin{bmatrix} \frac{6\pi}{4}t^2(e^{-2t^3}) + \frac{6\pi}{4}t^2(e^{-2t^3})\text{sen}(0.2\pi t) + \frac{\pi}{4}(1 - e^{-2t^3})\cos(0.2\pi t)0.2\pi \\ \frac{6\pi}{12}t^2(e^{-2t^3}) + \frac{6\pi}{12}t^2(e^{-2t^3})\text{sen}(0.4\pi t) + \frac{\pi}{4}(1 - e^{-2t^3})\cos(0.4\pi t)0.4\pi \\ \frac{6\pi}{20}t^2(e^{-2t^3}) + \frac{6\pi}{20}t^2(e^{-2t^3})\text{sen}(0.6\pi t) + \frac{\pi}{4}(1 - e^{-2t^3})\cos(0.6\pi t)0.6\pi \end{bmatrix} \quad (4.64)$$

$$\ddot{q}_d(t) = \begin{bmatrix} \frac{12\pi}{4}t(e^{-2t^3}) - \frac{36\pi}{4}t^4(e^{-2t^3}) + \frac{12\pi}{4}t(e^{-2t^3})\text{sen}(0.2\pi t) - \frac{36\pi}{4}t^4(e^{-2t^3})\text{sen}(0.2\pi t) \\ + \frac{12\pi}{4}t^2(e^{-2t^3})\cos(0.2\pi t)0.2\pi - \frac{\pi}{4}(1 - e^{-2t^3})\text{sen}(0.2\pi t)(0.2\pi)^2 \\ \frac{12\pi}{12}t(e^{-2t^3}) - \frac{36\pi}{12}t^4(e^{-2t^3}) + \frac{12\pi}{12}t(e^{-2t^3})\text{sen}(0.2\pi t) - \frac{36\pi}{12}t^4(e^{-2t^3})\text{sen}(0.2\pi t) \\ + \frac{12\pi}{12}t^2(e^{-2t^3})\cos(0.2\pi t)0.2\pi - \frac{\pi}{12}(1 - e^{-2t^3})\text{sen}(0.2\pi t)(0.2\pi)^2 \\ \frac{12\pi}{20}t(e^{-2t^3}) - \frac{36\pi}{20}t^4(e^{-2t^3}) + \frac{12\pi}{20}t(e^{-2t^3})\text{sen}(0.2\pi t) - \frac{36\pi}{20}t^4(e^{-2t^3})\text{sen}(0.2\pi t) \\ + \frac{12\pi}{20}t^2(e^{-2t^3})\cos(0.2\pi t)0.2\pi - \frac{\pi}{20}(1 - e^{-2t^3})\text{sen}(0.2\pi t)(0.2\pi)^2 \end{bmatrix} \quad (4.65)$$

El sistema es sometido a perturbaciones del tipo acopladas, definidas por el vector $w(t) = [0.38\text{sen}(2t) \ 0.38\text{sen}(4t) \ 0.38\text{sen}(8t)]^T$. Las ganancias del controlador PD son las siguientes: $K_p = \text{diag}\{632.8 \ 631.8 \ 630.8\}$ y $K_v = \text{diag}\{4.49 \ 4.39 \ 4.29\}$.

En la Figura 4.2 se tiene el diagrama con bloques de funciones en Simulink del robot manipulador, el controlador y la señal de referencia.

En las figuras 4.3 y 4.4 se aprecia el desempeño del sistema arriba mencionado con los tiempos de asentamiento de 500 ms, 100 ms y 60 ms para $q_1(t)$, $q_2(t)$ y $q_3(t)$ respectivamente, respecto al máximo sobrepaso se reporta 70 % para $q_1(t)$, 60 % para $q_2(t)$ y 20 % para $q_3(t)$ los cuales se pueden mejorar disminuyendo la ganancia del controlador PD. En lo que se refiere al error en estado estacionario el sistema reporta los siguientes valores: para la articulación $q_1(t)$ el error es de $0.3 \times 10^{-3} \text{ rad}$, para $q_2(t)$ el error es de $0.5 \times 10^{-3} \text{ rad}$ y para $q_3(t)$ el error es de $0.7 \times 10^{-3} \text{ rad}$. La magnitud del par máximo que se aplica a cada articulación varía dependiendo de lo alejado de las condiciones iniciales al origen, los cuales son: -993 Nm para el par de $q_1(t)$, 1001 Nm para el par de $q_2(t)$ y -986 Nm para el par de $q_3(t)$, los cuales decrecen asintóticamente conforme se alcanza el objetivo de control.

Respecto a las figuras 4.5, 4.6 y 4.7, éstas representan la estabilidad asintótica del punto de equilibrio del sistema en lazo cerrado (4.8), se observa que los estados $q_1(t)$, $q_2(t)$ y $q_3(t)$ parten de su condición inicial $\frac{\pi}{2}$, $-\frac{\pi}{2}$, $\frac{\pi}{2}$ respectivamente y convergen al origen, el cual es un punto de equilibrio estable.

Modelo continuo del Control por Retroalimentación de Estados PD aplicado al robot Niryo One.

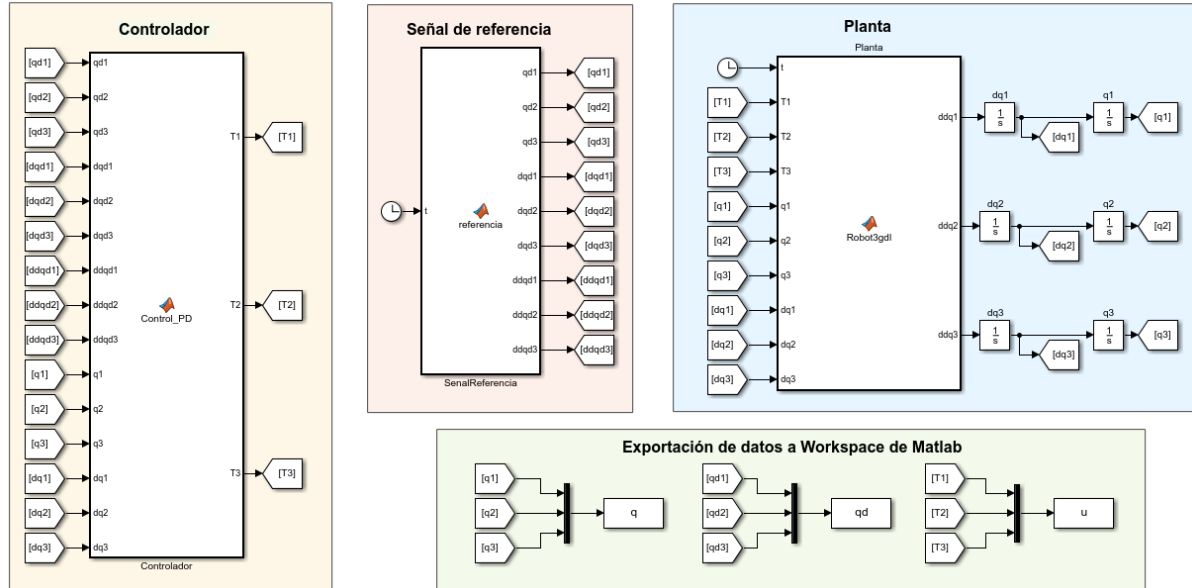


Figura 4.2 Diagrama de bloques de funciones en Simulink del sistema en lazo cerrado y tiempo continuo

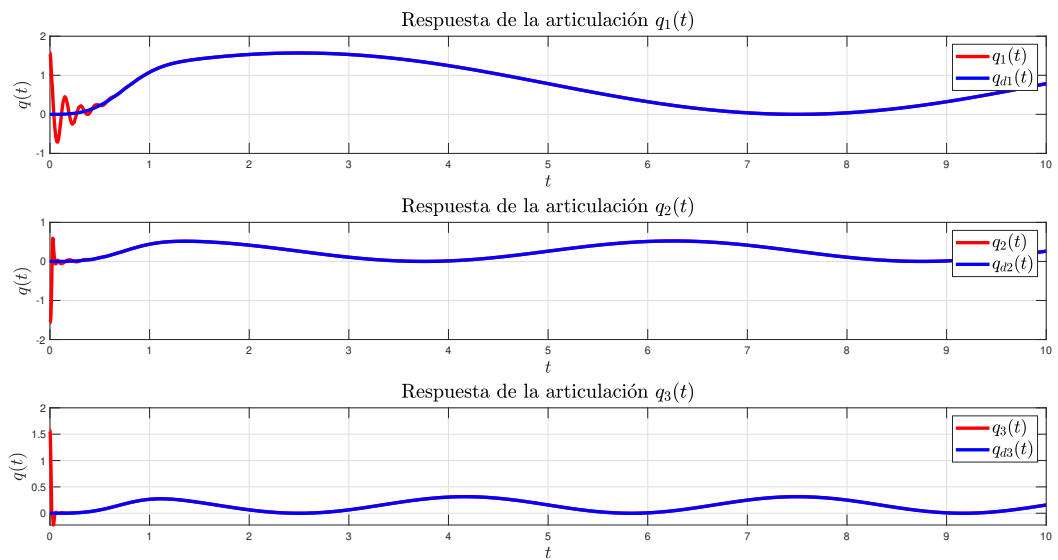


Figura 4.3 Respuesta en lazo cerrado del robot manipulador al problema de seguimiento de trayectorias, donde $q_1(t), q_2(t)$ y $q_3(t)$ son las articulaciones del robot y $q_{d1}(t), q_{d2}(t)$ y $q_{d3}(t)$ son las señales de referencia.

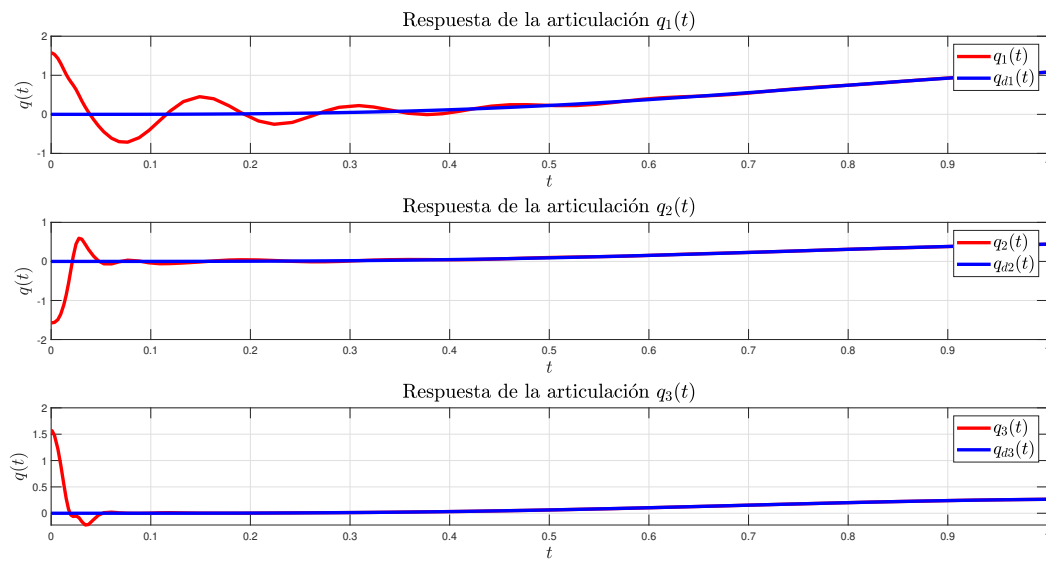


Figura 4.4 Vista ampliada de la Figura 4.3

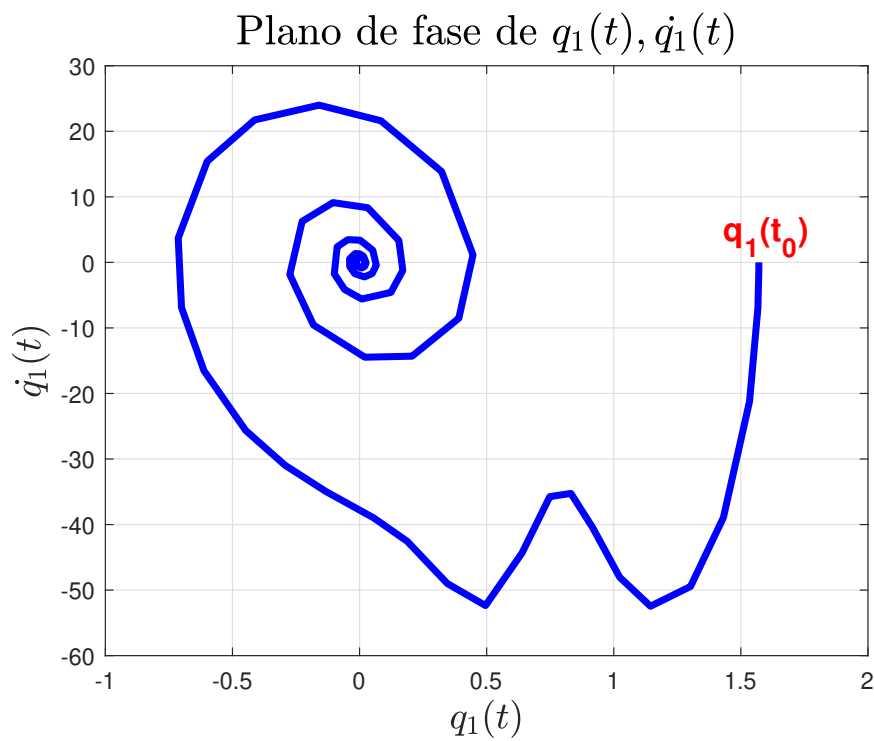
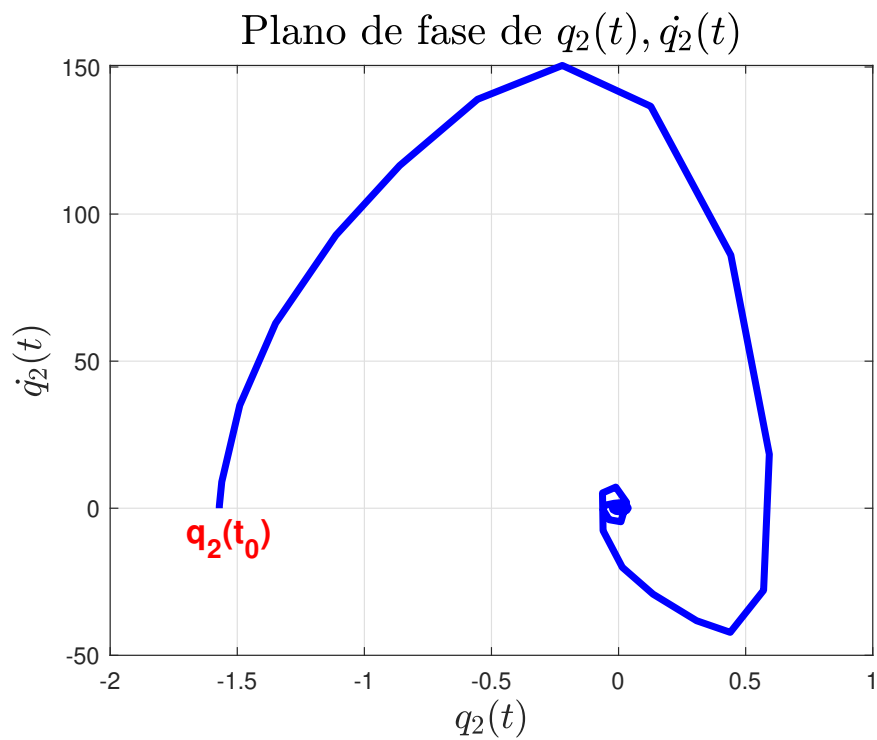
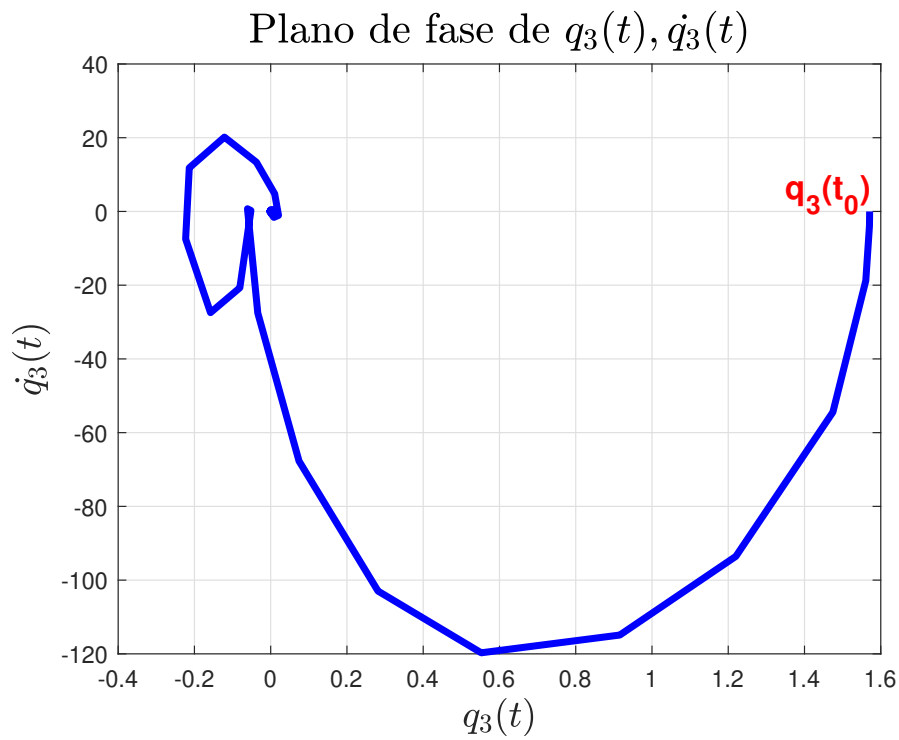


Figura 4.5 Plano de fase de la articulación $q_1(t)$

Figura 4.6 Plano de fase de la articulación $q_2(t)$ Figura 4.7 Plano de fase de la articulación $q_3(t)$

Capítulo 5

Discretización del controlador PD no lineal con compensación de trayectoria deseada.

5.1. Discretización del sistema en lazo cerrado

Al trabajar con sistemas computarizados, las señales análogas son convertidas a digitales, para esto la señal de entrada u se define de la siguiente forma

$$u(t) = u(kT) \text{ donde } kT \leq t < (k+1)T$$

Retomando las ecuaciones de estado del robot manipulador en lazo cerrado vistas en (4.8), se procede a la discretización del controlador.

$$\begin{aligned}\tilde{q}(t) &\cong x_1(k-1) \\ \dot{\tilde{q}}(t) &\cong \frac{x_1(k) - x_1(k-1)}{T} = x_2(k-1)\end{aligned}\tag{5.1}$$

Al despejar la ecuación (5.1) se obtiene el estado del error de trayectoria x_1 en modo discreto

$$x_1(k) = Tx_2(k-1) + x_1(k-1)$$

Se hace un corrimiento en las muestras, donde $k \rightarrow k+1$

$$x_1(k+1) = x_1(k) + Tx_2(k)$$

La segunda derivada de la posición en tiempo continuo \ddot{q} se compara con la de modo discreto de la siguiente manera

$$\ddot{q} \cong \frac{x_2(k) - x_2(k-1)}{T}$$

$$\frac{x_2(k) - x_2(k-1)}{T} = M^{-1}(q_d - x_1(k-1))[-C(q_d - x_1(k-1), \frac{q_d(k) - q_d(k-1)}{T} - x_2(k-1))x_2(k-1) + K_p x_1(k-1) + K_d x_2(k-1) + h(x_1(k-1), x_2(k-1), k-1) - w(k-1)]$$

$$x_2(k) = TM^{-1}(q_d - x_1(k-1))[-C(q_d - x_1(k-1), \frac{q_d(k) - q_d(k-1)}{T} - x_2(k-1))x_2(k-1) + K_p x_1(k-1) + K_d x_2(k-1) + h(x_1(k-1), x_2(k-1), k-1) - w(k-1)] + x_2(k-1)$$

Se hace un corrimiento en las muestras, donde $k \rightarrow k+1$. El estado $x_2(k+1)$ en modelo discreto resulta:

$$x_2(k+1) = x_2(k) + TM^{-1}(q_d - \tilde{q})[-C(q_d - \tilde{q}, \frac{q_d(k) - q_d(k-1)}{T} - \dot{\tilde{q}})\dot{\tilde{q}} + K_p \tilde{q} + K_d \dot{\tilde{q}} + h(\tilde{q}, \dot{\tilde{q}}, k-1) - w(k-1)] \quad (5.2)$$

$$\left. \begin{aligned} x_1(k+1) &= T\dot{\tilde{q}}(k) + x_1(k) \\ x_2(k+1) &= x_2(k) + TM^{-1}(\tilde{q} + q_d)[-C(\tilde{q} + q_d, \dot{\tilde{q}} + \frac{q_d(k) - q_d(k-1)}{T})\dot{\tilde{q}} + K_p \tilde{q} + K_d \dot{\tilde{q}} + h(\tilde{q}, \dot{\tilde{q}}, k) - w(k)] \end{aligned} \right\} \quad (5.3)$$

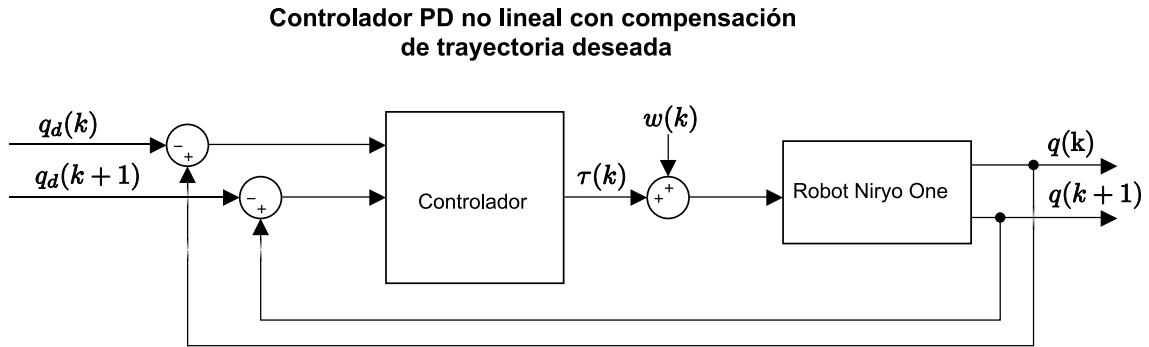


Figura 5.1 Representación del control discreto de la ecuación 5.3 aplicado al robot Niryo One

5.2. Simulación y resultados del sistema en lazo cerrado mediante MatLab/Simulink

La simulación se realizó en MatLab/Simulink ver 2018, para evaluar el desempeño que presenta el sistema discreto en lazo cerrado representado por (5.3), partiendo de las siguientes posiciones iniciales: $q(0) = [\frac{\pi}{2}, -\frac{\pi}{2}, \frac{\pi}{2}]^T$, $q(1) = [0 \ 0 \ 0]^T$ y respondiendo a los vectores de las señales de trayectorias deseadas, dado por:

$$\begin{aligned}
q_d(k) &= \left[\frac{\pi}{4}(1 - e^{-2(kT)^3}) + \frac{\pi}{4}(1 - e^{-2(kT)^3})\text{sen}(0.2\pi kT) \frac{\pi}{12}(1 - e^{-2(kT)^3}) \right. \\
&\quad \left. + \frac{\pi}{12}(1 - e^{-2(kT)^3})\text{sen}(0.4\pi kT) \frac{\pi}{20}(1 - e^{-2(kT)^3}) + \frac{\pi}{20}(1 - e^{-2(kT)^3})\text{sen}(0.6\pi kT) \right] \\
q_d(k+1) &= \left[\frac{6\pi}{4}(kT)^2 e^{-2(kT)^3} + \frac{6\pi}{4}(kT)^2 e^{-2(kT)^3} \text{sen}(0.2\pi kT) \right. \\
&\quad \left. + \frac{\pi}{4}(1 - e^{-2(kT)^3})\cos(0.2\pi kT) 0.2\pi \frac{6\pi}{12}(kT)^2 e^{-2(kT)^3} \right. \\
&\quad \left. + \frac{6\pi}{12}(kT)^2 e^{-2(kT)^3} \text{sen}(0.4\pi kT) + \frac{\pi}{12}(1 - e^{-2(kT)^3})\cos(0.4\pi kT) 0.4\pi \frac{6\pi}{20}(kT)^2 e^{-2(kT)^3} \right. \\
&\quad \left. + \frac{6\pi}{20}(kT)^2 e^{-2(kT)^3} \text{sen}(0.6\pi kT) + \frac{\pi}{20}(1 - e^{-2(kT)^3})\cos(0.6\pi kT) 0.6\pi \right] \\
q_d(k+2) &= \left[3\pi kT e^{-2(kT)^3} - 9\pi(kT)^4 e^{-2(kT)^3} + 3\pi(kT) e^{-2(kT)^3} \text{sen}(0.2\pi kT) \right. \\
&\quad \left. - 9\pi(kT)^4 e^{-2(kT)^3} \text{sen}(0.2\pi kT) + 3\pi(kT)^2 e^{-2(kT)^3} \cos(0.2\pi kT) 0.2\pi \right. \\
&\quad \left. - \frac{\pi}{4}(1 - e^{-2(kT)^3})\text{sen}(0.2\pi kT) 0.04\pi(kT) e^{-2(kT)^3} - 3\pi(kT)^4 e^{-2(kT)^3} \right. \\
&\quad \left. + \pi(kT) e^{-2(kT)^3} \text{sen}(0.4\pi kT) - 3\pi(kT)^4 e^{-2(kT)^3} \text{sen}(0.4\pi kT) \right. \\
&\quad \left. + \pi(kT)^2 e^{-2(kT)^3} \cos(0.4\pi kT) 0.4\pi - \frac{\pi}{12}(1 - e^{-2(kT)^3})\text{sen}(0.4\pi kT) 0.16\pi^2 0.6\pi kT e^{-2(kT)^3} \right. \\
&\quad \left. - 1.8\pi(kT)^4 e^{-2(kT)^3} + 0.6\pi(kT) e^{-2(kT)^3} \text{sen}(0.6\pi kT) - 1.8\pi(kT)^4 e^{-2(kT)^3} \text{sen}(0.6\pi kT) \right. \\
&\quad \left. + 0.6\pi(kT)^2 e^{-2(kT)^3} \cos(0.6\pi kT) 0.4\pi - \frac{\pi}{20}(1 - e^{-2(kT)^3})\text{sen}(0.6\pi kT) 0.16\pi^2 \right]
\end{aligned} \tag{5.4}$$

El sistema es sometido a perturbaciones del tipo acopladas, definidos por el vector: $w(kT) = [0.38\text{sen}(2kT) \ 0.38\text{sen}(4kT) \ 0.38\text{sen}(8kT)]^T$. Las ganancias del controlador PD son las mismas a las del sistema continuo (4.8). En las Figuras 5.3 y 5.4 se aprecia el desempeño del sistema arriba mencionado con los tiempos de asentamientos de 500 ms, 800 ms y 500 ms para $q_1(k)$, $q_2(k)$ y $q_3(k)$ respectivamente, el retardo en la convergencia de los estados a la trayectoria deseada se debe al algoritmo numérico para discretizar hacia adelante de Euler [23].

Respecto al máximo sobrepaso se reporta 90 % para $q_1(k)$, 50 % para $q_2(k)$ y 33 % para $q_3(k)$, los cuales son similares a los del sistema continuo. En lo que se refiere al error en estado estacionario el sistema reporta los siguientes valores: para la articulación $q_1(k)$ el error es de $0.5 \times 10^{-3} \text{ rad}$, para $q_2(k)$ el error es de $0.6 \times 10^{-3} \text{ rad}$ y para $q_3(k)$ el error es de $0.1 \times 10^{-3} \text{ rad}$, muy semejantes a los obtenidos en el sistema en lazo cerrado continuo.

La magnitud del par máximo que se aplica a cada articulación varía dependiendo de lo alejado de las condiciones iniciales al origen, los cuales son: -994 Nm para el par de $q_1(k)$, 1001 Nm para el par de $q_2(k)$ y -986 Nm para el par de $q_3(k)$, los cuales decrecen asintóticamente conforme se alcanza el objetivo de control.

Respecto a las Figuras 5.5, 5.6 y 5.7, representan la estabilidad asintótica del punto de equilibrio del sistema en lazo cerrado discreto (5.3), se observa que los estados $q_1(k)$, $q_2(k)$ y $q_3(k)$ parten de su condición inicial $[\frac{\pi}{2}, -\frac{\pi}{2}, \frac{\pi}{2}]$ respectivamente y convergen al origen, el

54 Discretización del controlador PD no lineal con compensación de trayectoria deseada.

cual es un punto de equilibrio estable. Las gráficas de los planos de fase para el sistema de lazo cerrado continuo y discreto son similares en su comportamiento.

Modelo discreto del Control por Retroalimentación de Estados PD aplicado al robot Niryo One

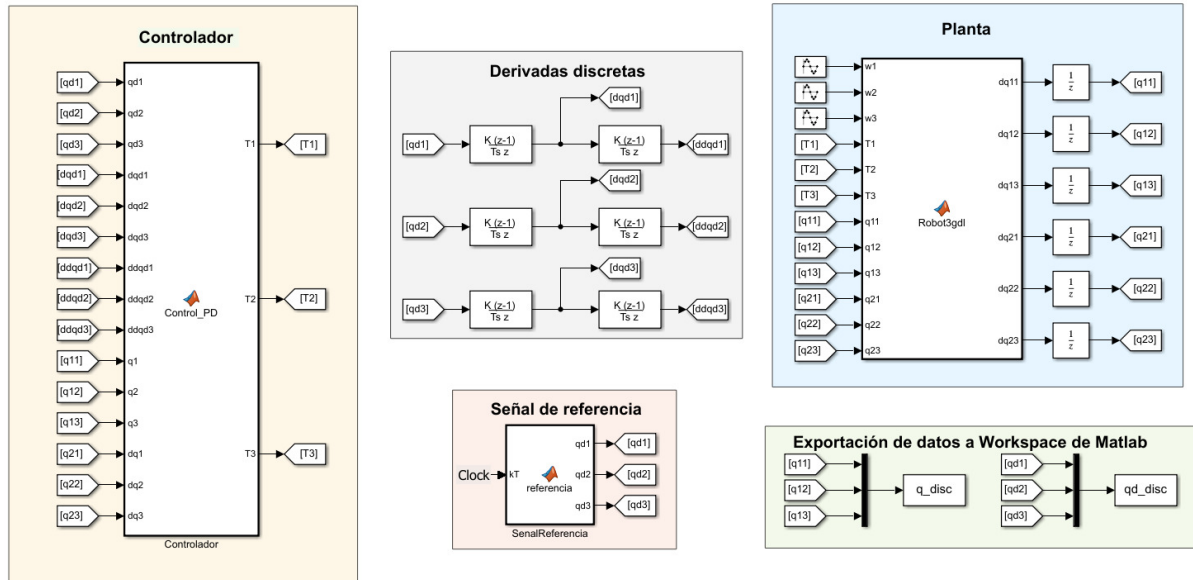


Figura 5.2 Diagrama de bloques en Simulink de la discretización del sistema a controlar

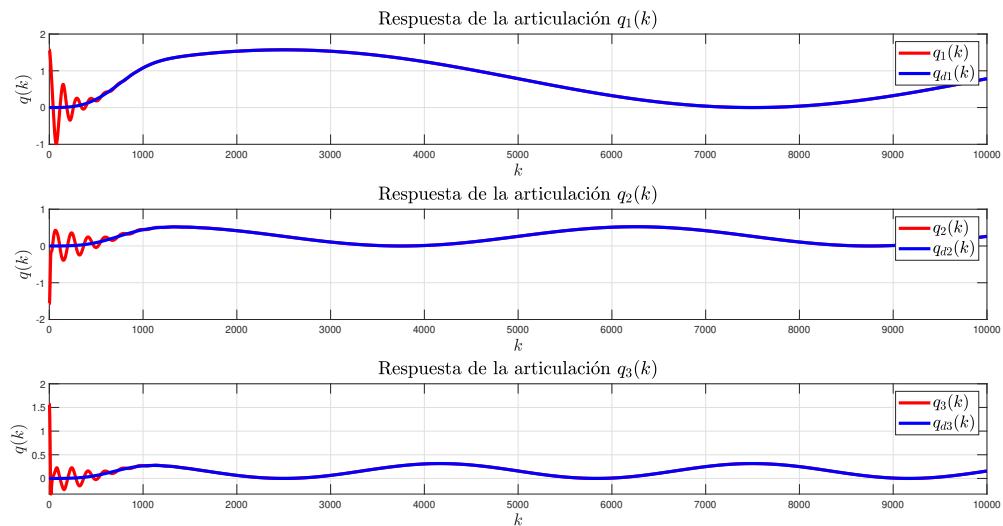


Figura 5.3 Respuesta de las articulaciones en modo discreto

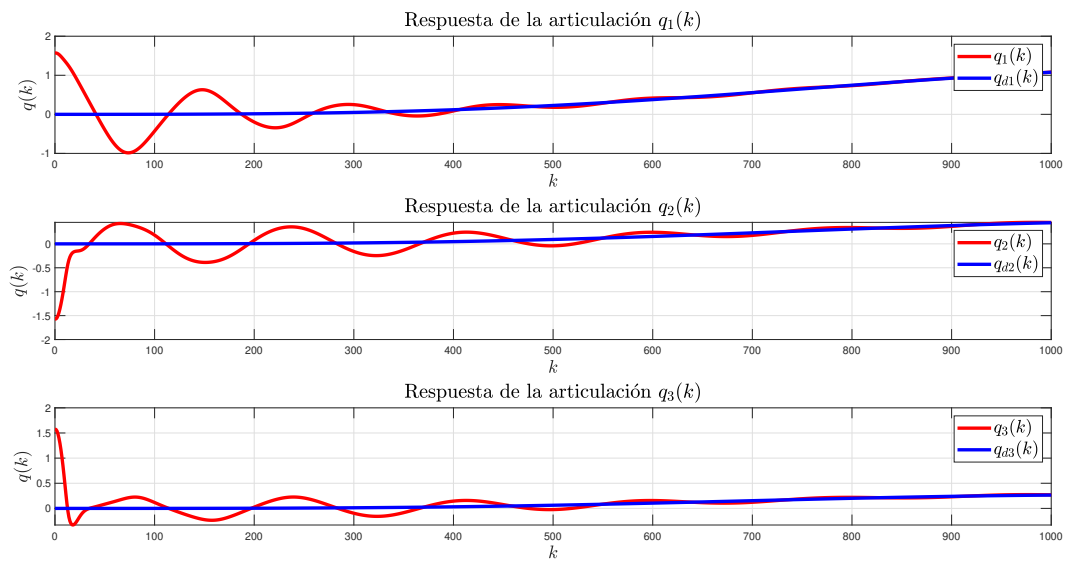
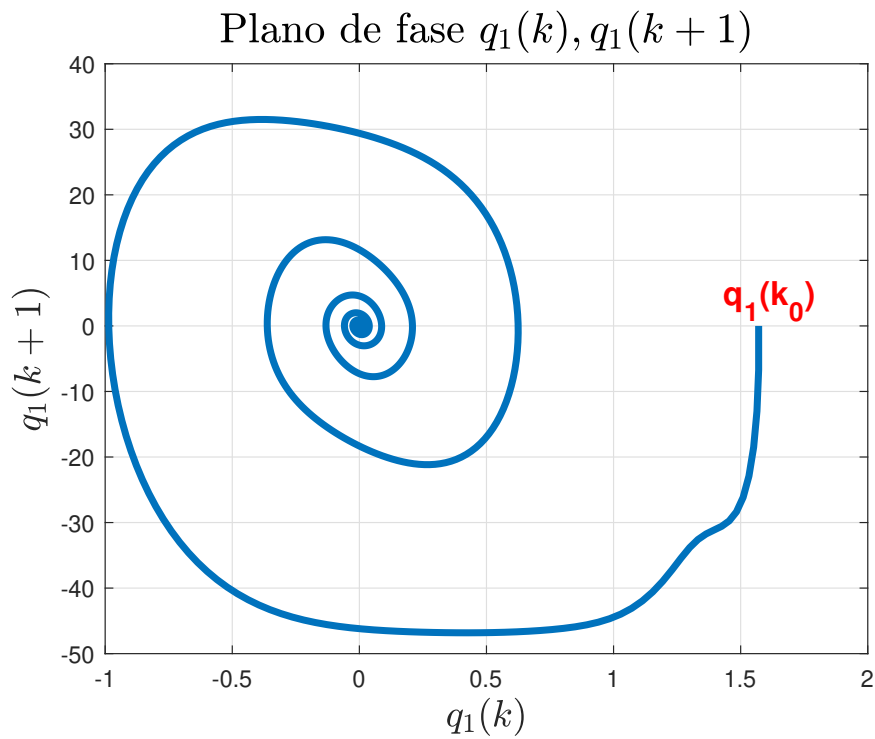


Figura 5.4 Respuesta de las articulaciones en modo discreto

Figura 5.5 Plano de fase de la articulación $q_1(k)$

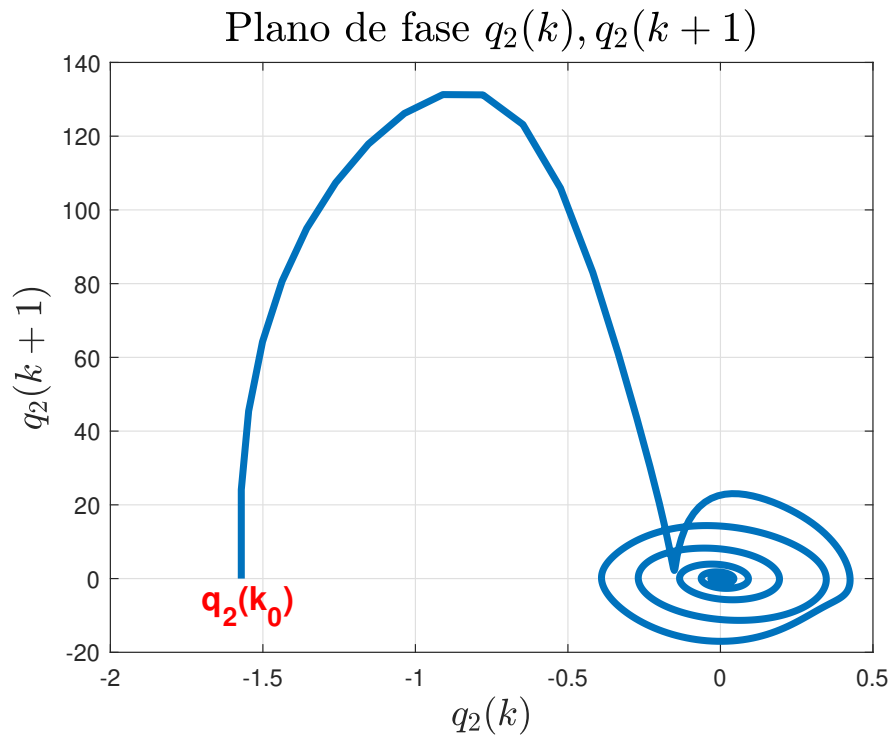


Figura 5.6 Plano de fase de la articulación $q_2(k)$

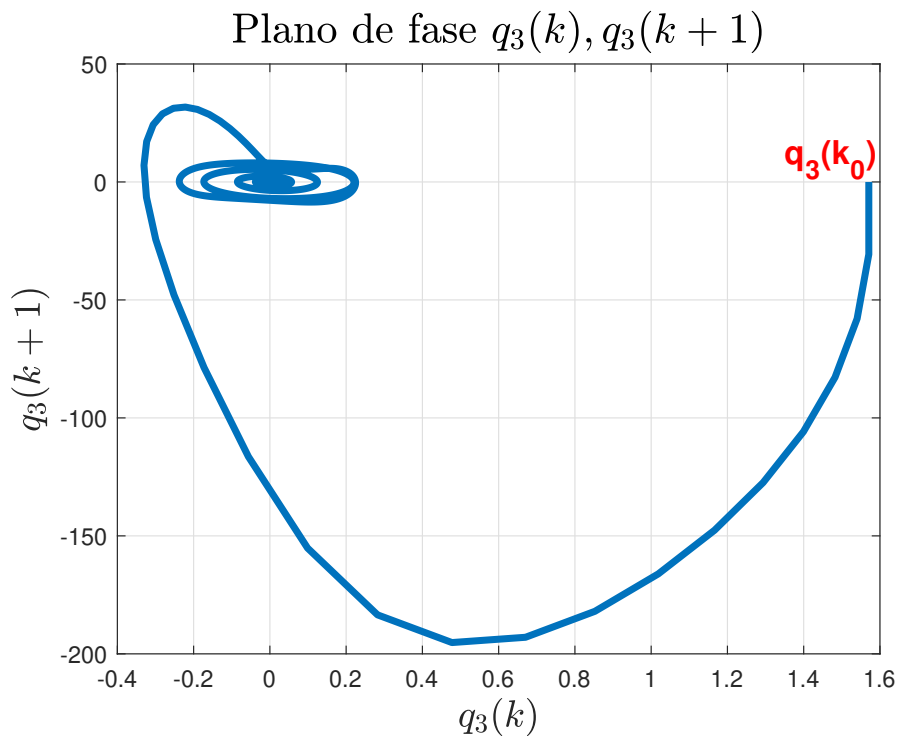


Figura 5.7 Plano de fase de la articulación $q_3(k)$

5.3. Simulación y resultados del sistema en lazo cerrado mediante CoppeliaSim

Otra alternativa de simulación considerada es el software CoppeliaSim de Coppelia Robotics, el cual nos permite simular un robot modelado por completo o por partes, por medio de *scripts* usando diversos lenguajes de programación, aunque de forma predeterminada se usa LUA.

Este software se utiliza para la automatización de procesos, aplicaciones educativas, prototipado y verificación, entre otras. En este caso, se utilizó para la comprobación del controlador PD en lazo cerrado aplicado a un robot manipulador.

El robot utilizado es un Niryo One de 6 GDL, en el cual el fabricante utiliza piezas impresas en 3D y aluminio para la estructura y una tarjeta *Raspberry pi 4* para su control. En la Figura 5.8 se aprecia el robot en CoppeliaSim.

Este robot fue modificado en el software, eliminando 3 GDL para dejarlo con 3, con el fin de poder aplicar el control realizado en esta tesis y cumplir con las restricciones. En la Figura 5.9 se observa el robot utilizado para la simulación, ya con las modificaciones realizadas.

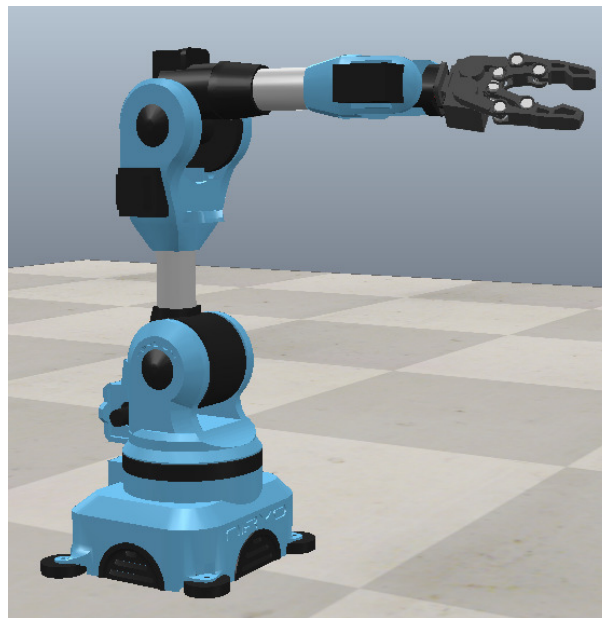


Figura 5.8 Robot Niryo One, originalmente de 6 GDL

Una vez que se eliminaron las articulaciones sobrantes en el modelo, se eliminan también en el árbol del proyecto. Y en el *script* del robot se cambia el código precargado por Coppelia, por el control desarrollado en Matlab, el cual incluye los parámetros del robot de 3 GDL,

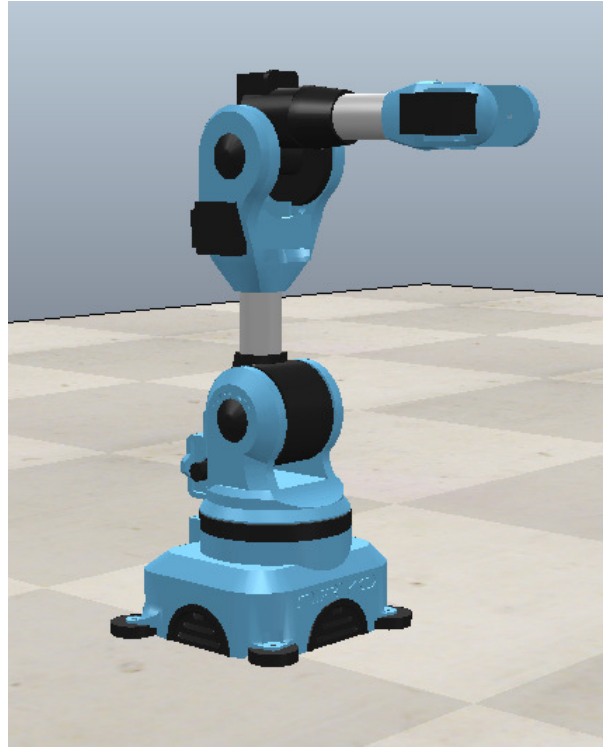


Figura 5.9 Robot Niryo One, originalmente de 6 GDL, modificado para fines de simulación

ecuación dinámica, algoritmo de control y la señal de referencia.

La simulación se realizó en CoppeliaSim Edu, Versión 4.3.0 (rev. 3), para evaluar el desempeño que presenta el sistema discreto en lazo cerrado representado por (5.3), partiendo de las siguientes posiciones iniciales: $q(0) = \left[\frac{\pi}{2}, -\frac{\pi}{2}, \frac{\pi}{2} \right]^T$, $q(1) = [0 \ 0 \ 0]^T$ y respondiendo a los vectores de las señales de trayectorias deseadas, dado por (5.4).

En la Figura 5.10 se aprecia el desempeño del sistema discreto en lazo cerrado (5.3), con los tiempos de asentamientos de 1.5 s, 2.5 ms y 2.2 ms para $q_1(k)$, $q_2(k)$ y $q_3(k)$ respectivamente. El retardo en la convergencia de los estados a la trayectoria deseada se supone a la incertidumbre del modelo respecto a la configuración del par aplicado en los actuadores en la plataforma de CoppeliaSim.

Respecto al máximo sobrepaso se reporta 0% para las tres articulaciones. En lo que se refiere al error en estado estacionario el sistema reporta los siguientes valores: para la articulación $q_1(k)$ el error es de $4 \times 10^{-3} \text{ rad}$, para $q_2(k)$ el error es de $10 \times 10^{-3} \text{ rad}$ y para $q_3(k)$ el error es de $8 \times 10^{-3} \text{ rad}$.

La magnitud del par máximo que se aplica a cada articulación varía dependiendo de lo alejado de las condiciones iniciales al origen, los cuales son: -994 Nm para el par de

$q_1(k)$, 1001 Nm para el par de $q_2(k)$ y -986 Nm para el par de $q_3(k)$, los cuales decrecen asintóticamente conforme se alcanza el objetivo de control.

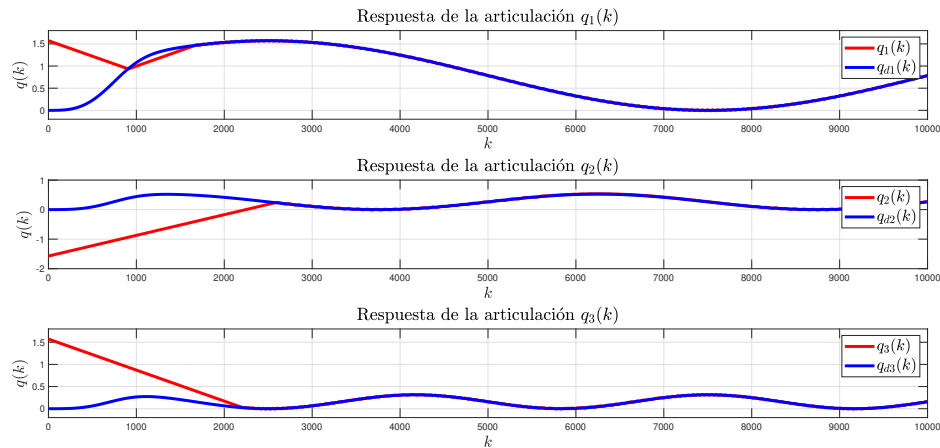


Figura 5.10 Respuesta en el tiempo discreto de las articulaciones $q_1(k)$, $q_2(k)$ y $q_3(k)$ en CoppeliaSim

5.4. Resultados de la Tesis

Se analizó y sintetizó un control PD no lineal con compensación de la dinámica del robot sobre la trayectoria deseada aplicado al robot manipulador Niryo One de tres grados de libertad, en el dominio del tiempo continuo, y se verificó su desempeño en Simulink, comprobando que el punto de equilibrio del sistema en lazo cerrado inicializado por la condición inicial y libre de perturbaciones sea asintóticamente estable en forma global y que la ganancia del sistema en lazo cerrado perturbado atenúan las perturbaciones a la salida.

El paso siguiente fue discretizar el sistema de control en lazo cerrado y comprobar su desempeño; la respuesta obtenida en Simulink fue aproximada a la obtenida por el sistema en lazo cerrado en tiempo continuo (ver Tabla 5.1), la diferencia en los criterios de desempeño radica en el método seleccionado para realizar la discretización, según [23] el algoritmo de Euler presenta un mayor error y tiempo de convergencia en comparación con otros métodos, por ejemplo: Trapezoidal, Runge Kutta, entre otros. Sin embargo, también se verificó su desempeño en Simulink, bajo condiciones similares al sistema en tiempo continuo (como condiciones iniciales, vector de función de perturbación, ganancias del sistema de control) comprobando que el punto de equilibrio del sistema en lazo cerrado inicializado por la condición inicial y libre de perturbaciones sea asintóticamente estable en forma global y que la ganancia del sistema en lazo cerrado perturbado atenúan las perturbaciones a la salida.

Respecto a los resultados obtenidos en la simulación del sistema de control discreto en lazo cerrado realizado en CoppeliaSim se observan resultados hasta tres veces más alejados de los obtenidos en la simulación del sistema continuo, por ejemplo tiempo de asentamiento de hasta 2.5 s, máximo sobrepaso de 0 % y error en estado estable de hasta 10 milésimas de radianes, lo cual es muy obvio en el efecto oscilante de las articulaciones que presentan los resultados gráficos. La hipótesis para explicar estos resultados es la siguiente: derivado de la falta de pericia en la plataforma de CoppeliaSim, específicamente en la configuración del vector de par aplicado del sistema de control con los actuadores del robot Niryo One, esto se sustenta por el máximo sobrepaso y los tiempos de asentamiento obtenidos.

Criterios de desempeño	Continuo - Simulink $q_1(t)$ $q_2(t)$ $q_3(t)$	Discreto - Simulink $q_1(k)$ $q_2(k)$ $q_3(k)$	Discreto - CoppeliaSim $q_1(k)$ $q_2(k)$ $q_3(k)$
Tiempo de asentamiento	500 ms 100 ms 60 ms	500 ms 800 ms 500 ms	1.5 s 2.5 s 2.2 s
Máximo sobrepaso	70 % 60 % 20 %	90 % 50 % 33 %	0 % 0 % 0 %
Error en estado estacionario	$0.3 \times 10^{-3} \text{ rad}$ $0.5 \times 10^{-3} \text{ rad}$ $0.7 \times 10^{-3} \text{ rad}$	$0.5 \times 10^{-3} \text{ rad}$ $0.6 \times 10^{-3} \text{ rad}$ $0.1 \times 10^{-3} \text{ rad}$	$4 \times 10^{-3} \text{ rad}$ $10 \times 10^{-3} \text{ rad}$ $8 \times 10^{-3} \text{ rad}$
Par aplicado	-993 Nm 1001 Nm -986 Nm	-994 Nm 1001 Nm -986 Nm	-994 Nm 1001 Nm -986 Nm

Tabla 5.1 Resultados de la simulación del desempeño del sistema en lazo cerrado

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones generales

En el Capítulo tres se obtiene el modelo dinámico de un robot manipulador de tres grados de libertad, con articulaciones del tipo rotacional y rígidas, se particulariza con los parámetros del robot Niryo One, y se calculan las cotas máximas de la matriz de inercia, la matriz de fuerzas centrífugas y de Coriolis y del vector de fuerzas gravitacionales. Se discretiza el modelo dinámico del robot Niryo One por el método de Euler, y se realiza la validación en simulación, obteniendo resultados similares a los obtenidos por el sistema en tiempo continuo.

En el Capítulo cuatro se analiza y sintetiza en control PD no lineal con compensación de la dinámica del robot sobre la trayectoria deseada en tiempo continuo, en primer lugar se determina la cota mínima de la constante de Lipchitz, que es necesaria y suficiente para garantizar existencia y unicidad del punto de equilibrio, también se establecen las cotas mínimas del sistema de control que son necesarias y suficientes para garantizar estabilidad asintótica en forma global del sistema en lazo cerrado libre de perturbaciones, se valida el desempeño del sistema de control en lazo cerrado con MatLab/Simulink. Se discretiza el sistema en lazo cerrado mediante el método numérico de Euler y se valida el desempeño del sistema de control discreto en MatLab/Simulink y en CoppeliaSim. La propuesta inicial para este trabajo de tesis era discretizar el modelo dinámico del robot manipulador y posteriormente diseñar el sistema de control discreto. En este proceso, se logró establecer la cota mínima que es necesaria y suficiente para garantizar existencia y unicidad del punto de equilibrio del sistema de control en lazo cerrado discreto, el siguiente paso era proponer una función candidata de Lyapunov discreta para el análisis de estabilidad asintótica en forma global del punto de equilibrio, al no encontrar una propuesta viable, se optó por la opción reportada en el Capítulo cinco.

En el Capítulo cinco se realiza la discretización del sistema de control en lazo cerrado mediante el método numérico de Euler y se valida el desempeño en simulación, se realiza el comparativo entre la simulación de los sistemas de control en lazo cerrado en ambos dominios (continuo y discreto), se muestra una variación en los índices de desempeño entre

ambos sistemas, esto debido a que el método de Euler no es la mejor opción para discretizar sistemas según lo demuestra Neuman en [23], por esta razón se optó inicialmente por el método propuesto por Nicosia en [27] el cual aplica técnicas de discretización numérica al problema de minimización de las funciones de Lagrange, sin embargo este método no se pudo aplicar a este trabajo de tesis debido a que se descubrió un error de consistencia en el álgebra matricial que presentaban las ecuaciones del autor, por lo cual se descartó. Sin embargo ambos sistemas cumplen con el objetivo de control establecido.

6.2. Trabajo a futuro

Proponer un modelo discreto del robot manipulador basado en técnicas numéricas de integración simpléctica que preservan la estructura simpléctica del espacio de fase de los sistemas hamiltonianos, respetando así las propiedades inherentes a su dinámica, por lo que son apropiados para traducir el paso de formación de energía a tiempo discreto [18], continuando con esta línea de análisis proponer una función de Lyapunov discreta para el diseño del control PD no lineal discreto.

Capítulo 7

Anexos

7.1. Código de Matlab en tiempo continuo

A continuación se presenta el código de los bloques de Matlab para el sistema en lazo cerrado y tiempo continuo.

7.1.1. Controlador

```
function [T1, T2, T3, x1] = Control_PD(qd1,qd2,qd3,dqd1,dqd2,dqd3,ddqd1,ddqd2,ddqd3,
q1,q2,q3,dq1,dq2,dq3) % Parametros del Niryo One
l1 = 0.16048; % m
l2 = 0.297; % m
m1 = 1; % kg
m2 = 1.631; % kg
J1 = 0.004676; % kgm2
J2 = 0.01468; % kgm2
J3 = 0.007468; % kgm2
g = 9.81; % m/seg2

% Friccion Viscosa
f = [0.1    0    0;    0    0.2    0;    0    0    0.15];

% Ganancia del Control PD
Kv = [4.4889    0    0;    0    4.3889    0;    0    0    4.2889];
Kp = [632.8277    0    0;    0    631.8277    0;    0    0    630.8277];

% Vector de Velocidad Angular Deseada
dqd = [dq1; dq2; dq3];

% Vector de Aceleración Angular Deseada
ddqd = [ddqd1; ddqd2; ddqd3];
```

% Error de Posición

$$x1 = [q1-qd1; q2-qd2; q3-qd3];$$

% Error de Velocidad

$$x2 = [dq1-dqd1; dq2-dqd2; dq3-dqd3];$$

% Obtener la dinámica del sistema

% Matriz de Inercia

$$M11 = m2*l1*l1*cos(qd2)*cos(qd2) + 2*m2*l1*l2*cos(qd2)*cos(qd2+qd3) + m2*l2*l2*cos(qd2+qd3)*cos(qd2+qd3) + m1*l1*l1*cos(qd2)*cos(qd2) + J1;$$

$$M22 = m1*l1*l1 + m2*l1*l1 + 2*m2*l1*l2*cos(qd3) + m2*l2*l2 + J2 + J3;$$

$$M23 = m2*l1*l2*cos(qd3) + m2*l2*l2 + J3;$$

$$M33 = m2*l2*l2 + J3;$$

$$M = [M11, \quad 0, \quad 0; \quad 0, \quad M22, \quad M23; \quad 0, \quad 0, \quad M33]$$

% Matriz de Coriolis

$$C11 = -m2*l1*l1*sin(qd2)*cos(qd2)*dq2 - m2*l1*l2*sin(qd2)*cos(qd2+qd3)*dq2 - m1*l1*l1*sin(qd2)*cos(qd2)*dq2 - m2*l1*l2*cos(qd2)*sin(qd2+qd3)*(dq2+dq3) - m2*l2*l2*sin(qd2+qd3)*cos(qd2+qd3)*(dq2+dq3);$$

$$C12 = -m2*l1*l1*sin(qd2)*cos(qd2)*dq1 - m2*l1*l2*cos(qd2)*sin(qd2+qd3)*dq1 - m2*l1*l2*sin(qd2)*cos(qd2+qd3)*dq1 - m2*l2*l2*sin(qd2+qd3)*cos(qd2+qd3)*dq1 - m1*l1*l1*sin(qd2)*cos(qd2)*dq1;$$

$$C13 = -m2*l1*l2*cos(qd2)*sin(qd2+qd3)*dq1 - m2*l2*l2*sin(qd2+qd3)*cos(qd2+qd3)*dq1;$$

$$C21 = m2*l1*l1*sin(qd2)*cos(qd2)*dq1 + m2*l1*l2*cos(qd2)*sin(qd2+qd3)*dq1 + m2*l1*l2*sin(qd2)*cos(qd2+qd3)*dq1 + m2*l2*l2*sin(qd2+qd3)*cos(qd2+qd3)*dq1 + m1*l1*l1*sin(qd2)*cos(qd2)*dq1;$$

$$C22 = -m2*l1*l2*sin(qd3)*dq3;$$

$$C23 = -m2*l1*l2*sin(qd3)*(dq2+dq3);$$

$$C31 = m2*l1*l2*cos(qd2)*sin(qd2+qd3)*dq1 + m2*l2*l2*sin(qd2+qd3)*cos(qd2+qd3)*dq1;$$

$$C32 = m2*l1*l2*sin(qd3)*dq2;$$


```

C = [C11, C12, C13; C21, C22, C23; C31, C32, 0

% Vector de gravedad
gr1=0.0;

gr2=g*m1*l1*cos(qd2) + g*m2*l1*cos(qd2) + g*m2*l2*cos(qd2+qd3);

gr3=g*m2*l2*cos(qd2+qd3);

gr = [gr1; gr2; gr3];

%Señal de Control PD mas Compensación de la Dinámica de la planta
T = M*ddqd + C*dqd + gr + f*dqd - Kp*x1 - Kv*x2;

T1 = T(1);
T2 = T(2);
T3 = T(3)

```

7.1.2. Planta

```

function [ddq1,ddq2,ddq3] = Robot3gdl(t,T1,T2,T3,q1,q2,q3,dq1,dq2,dq3)

% Parámetros del Niryo
l1 = 0.16048; % m
l2 = 0.297; % m
m1 = 1; % Kg
m2 = 1.631; % Kg
J1 = 0.004676; % Kg m2
J2 = 0.01468; % Kg m2
J3 = 0.007468; % Kg m2
g = 9.81; % m/seg2

% Vector de velocidad angular
dq = [dq1; dq2; dq3];

% Fricción Viscosa
f = [0.1 0 0; 0 0.2 0; 0 0 0.15];

% Obtener la dinámica del sistema
% Matriz de Inercia
M11 = m2*l1*l1*cos(q2)*cos(q2) + 2*m2*l1*l2*cos(q2)*cos(q2+q3)
+ m2*l2*l2*cos(q2+q3)*cos(q2+q3) + m1*l1*l1*cos(q2)*cos(q2) + J1;

```

$$M22 = m1*l1*l1 + m2*l1*l1 + 2*m2*l1*l2*\cos(q3) + m2*l2*l2 + J2 + J3;$$

$$M23 = m2*l1*l2*\cos(q3) + m2*l2*l2 + J3;$$

$$M33 = m2*l2*l2 + J3;$$

$$M = [M11, \quad 0, \quad 0; \quad 0, \quad M22, \quad M23; \quad 0, \quad 0, \quad M33];$$

% Matriz de Coriolis

$$C11 = -m2*l1*l1*\sin(q2)*\cos(q2)*dq2 - m2*l1*l2*\sin(q2)*\cos(q2+q3)*dq2 \\ - m1*l1*l1*\sin(q2)*\cos(q2)*dq2 - m2*l1*l2*\cos(q2)*\sin(q2+q3)*(dq2+dq3) \\ - m2*l2*l2*\sin(q2+q3)*\cos(q2+q3)*(dq2+dq3);$$

$$C12 = -m2*l1*l1*\sin(q2)*\cos(q2)*dq1 - m2*l1*l2*\cos(q2)*\sin(q2+q3)*dq1 \\ - m2*l1*l2*\sin(q2)*\cos(q2+q3)*dq1 - m2*l2*l2*\sin(q2+q3)*\cos(q2+q3)*dq1 \\ - m1*l1*l1*\sin(q2)*\cos(q2)*dq1;$$

$$C13 = -m2*l1*l2*\cos(q2)*\sin(q2+q3)*dq1 - m2*l2*l2*\sin(q2+q3)*\cos(q2+q3)*dq1;$$

$$C21 = m2*l1*l1*\sin(q2)*\cos(q2)*dq1 + m2*l1*l2*\cos(q2)*\sin(q2+q3)*dq1 \\ + m2*l1*l2*\sin(q2)*\cos(q2+q3)*dq1 + m2*l2*l2*\sin(q2+q3)*\cos(q2+q3)*dq1 \\ + m1*l1*l1*\sin(q2)*\cos(q2)*dq1;$$

$$C22 = -m2*l1*l2*\sin(q3)*dq3;$$

$$C23 = -m2*l1*l2*\sin(q3)*(dq2+dq3);$$

$$C31 = m2*l1*l2*\cos(q2)*\sin(q2+q3)*dq1 + m2*l2*l2*\sin(q2+q3)*\cos(q2+q3)*dq1;$$

$$C32 = m2*l1*l2*\sin(q3)*dq2;$$

$$C = [C11, C12, C13; C21, C22, C23; C31, C32, 0];$$

% Vector de gravedad

$$gr1=0.0;$$

$$gr2=g*m1*l1*\cos(q2) + g*m2*l1*\cos(q2) + g*m2*l2*\cos(q2+q3);$$

$$gr3=g*m2*l2*\cos(q2+q3);$$

$$gr = [gr1; gr2; gr3];$$

% Par de entrada

$$T = [T1; T2; T3];$$

% Vector de Perturbación

$$w1 = 0.38*\sin(2*t); \%22 \text{ deg}$$

$$w2 = 0.38*\sin(4*t);$$

$$w3 = 0.38*\sin(8*t);$$

$$w = [w1; w2; w3];$$

% Obtener la inversa de la Matriz M

$$\det M = M11*(M22*M33 - M23*M23);$$

```

Mi11 = (1.0/detM)*(M22*M33-M23*M23);
Mi22 = (1.0/detM)*(M11*M33);
Mi23 = (1.0/detM)*(M11*M23);
Mi33 = (1.0/detM)*(M11*M22);

Mi = [Mi11, 0, 0, 0, Mi22, Mi23, 0, 0, Mi33];

% Dinámica del Sistema
ddq = Mi*( -C*dq - gr -f*dq+ T + w);

ddq1 = ddq(1);
ddq2 = ddq(2);
ddq3 = ddq(3);

```

7.1.3. Señal de referencia

```

function [qd1,qd2,qd3,dqd1,dqd2,dqd3,ddqd1,ddqd2,ddqd3] = referencia(t)
b1 = pi/4; b2 = pi/12; b3 = pi/20;
c1 = pi/4; c2 = pi/12; c3 = pi/20;
f1 = 0.1; f2 = 0.2; f3 = 0.3;
w1 = 2*pi*f1; w2 = 2*pi*f2; w3 = 2*pi*f3;

qd1 = 0*(b1*(1-exp(-2.0*t^3)) + c1*(1-exp(-2.0*t^3))*sin(w1*t));
qd2 = 0*(b2*(1-exp(-2.0*t^3)) + c2*(1-exp(-2.0*t^3))*sin(w2*t));
qd3 = 0*(b3*(1-exp(-2.0*t^3)) + c3*(1-exp(-2.0*t^3))*sin(w3*t));

dqd1 = 0*(6*b1*t^2*exp(-2.0*t^3) + 6*c1*t^2*exp(-2.0*t^3)*sin(w1*t) + c1*(1-exp(-2.0*t^3))*cos(w1*t)*w1);
dqd2 = 0*(6*b2*t^2*exp(-2.0*t^3) + 6*c2*t^2*exp(-2.0*t^3)*sin(w2*t) + c2*(1-exp(-2.0*t^3))*cos(w2*t)*w2);
dqd3 = 0*(6*b3*t^2*exp(-2.0*t^3) + 6*c3*t^2*exp(-2.0*t^3)*sin(w3*t) + c3*(1-exp(-2.0*t^3))*cos(w3*t)*w3);

ddqd1 = 0*(12*b1*t*exp(-2.0*t^3) - 36*b1*t^4*exp(-2.0*t^3) + 12*c1*t*exp(-2.0*t^3)*sin(w1*t) - 36*c1*t^4*exp(-2.0*t^3)*sin(w1*t) + 12*c1*t^2*exp(-2.0*t^3)*cos(w1*t)*w1 - c1*(1-exp(-2.0*t^3))*sin(w1*t)*w1^2);
ddqd2 = 0*(12*b2*t*exp(-2.0*t^3) - 36*b2*t^4*exp(-2.0*t^3) + 12*c2*t*exp(-2.0*t^3)*sin(w2*t) - 36*c2*t^4*exp(-2.0*t^3)*sin(w2*t) + 12*c2*t^2*exp(-2.0*t^3)*cos(w2*t)*w2 - c2*(1-exp(-2.0*t^3))*sin(w2*t)*w2^2);
ddqd3 = 0*(12*b3*t*exp(-2.0*t^3) - 36*b3*t^4*exp(-2.0*t^3) + 12*c3*t*exp(-2.0*t^3)*sin(w3*t) - 36*c3*t^4*exp(-2.0*t^3)*sin(w3*t) + 12*c3*t^2*exp(-2.0*t^3)*cos(w3*t)*w3 - c3*(1-exp(-2.0*t^3))*sin(w3*t)*w3^2);

```

$\cos(w3 * t) * w3 - c3 * (1 - \exp(-2.0 * t^3)) * \sin(w3 * t) * w3^2);$

7.2. Código de Matlab en tiempo discreto

7.2.1. Planta

```
function [dq11,dq12,dq13,dq21,dq22,dq23] =
Robot3gdl(w1,w2,w3,T1,T2,T3,q11,q12,q13,q21,q22,q23)
```

% Parámetros del Niryo

$l1 = 0.16048; \% \text{ m}$

$l2 = 0.297; \% \text{ m}$

$m1 = 1; \% \text{ kg}$

$m2 = 1.631; \% \text{ kg}$

$J1 = 0.004676; \% \text{ kg m}^2$

$J2 = 0.01468; \% \text{ kg m}^2$

$J3 = 0.007468; \% \text{ kg m}^2$

$g = 9.81; \% \text{ m/seg}^2$

$Ts = 0.001;$

% Posiciones angulares

$q1 = q11;$

$q2 = q12;$

$q3 = q13;$

% Derivadas de las posiciones

$dq1 = q21;$

$dq2 = q22;$

$dq3 = q23;$

% Representación en Espacio de Estados del Robot de 3 GDL.

$m11 = m2 * l1^2 * \cos(q2)^2 + 2 * m2 * l1 * l2 * \cos(q2) * \cos(q2 + q3) + m2 * l2^2 * \cos(q2 + q3)^2 + m1 * l1^2 * \cos(q2)^2 + J1;$

$m22 = m1 * l1^2 + m2 * l1^2 + 2 * m2 * l1 * l2 * \cos(q3) + m2 * l2^2 + J2 + J3;$

$m23 = m2 * l1 * l2 * \cos(q3) + m2 * l2^2 + J3;$

$m33 = m2 * l2^2 + J3;$

$M = [m11, \quad 0, \quad 0; \quad 0, \quad m22, \quad m23; \quad 0, \quad m23, \quad m33];$

$c11 = -m2 * l1^2 * \sin(q2) * \cos(q2) * dq2 - m2 * l1 * l2 * \sin(q2) * \cos(q2 + q3) * dq2 - m1 * l1^2 * \sin(q2) * \cos(q2) * dq2 - m2 * l1 * l2 * \cos(q2) * \sin(q2 + q3) * (dq2 + dq3) - m2 * l2^2 * \sin(q2 + q3) * \cos(q2 + q3) * (dq2 + dq3);$

```

c12 = -m2 * l1^2 * sin(q2) * cos(q2) * dq1 - m2 * l1 * l2 * cos(q2) * sin(q2 + q3) * dq1 - m2 *
l1 * l2 * sin(q2) * cos(q2 + q3) * dq1 - m2 * l2^2 * sin(q2 + q3) * cos(q2 + q3) * dq1 - m1 *
l1^2 * sin(q2) * cos(q2) * dq1;
c13 = -m2 * l1 * l2 * cos(q2) * sin(q2 + q3) * dq1 - m2 * l2^2 * sin(q2 + q3) * cos(q2 + q3) *
dq1;
c21 = m2 * l1^2 * sin(q2) * cos(q2) * dq1 + m2 * l1 * l2 * cos(q2) * sin(q2 + q3) * dq1 + m2 *
l1 * l2 * sin(q2) * cos(q2 + q3) * dq1 + m2 * l2^2 * sin(q2 + q3) * cos(q2 + q3) * dq1 + m1 *
l1^2 * sin(q2) * cos(q2) * dq1;
c22 = -m2 * l1 * l2 * sin(q3) * dq3;
c23 = -m2 * l1 * l2 * sin(q3) * (dq2 + dq3);
c31 = m2 * l1 * l2 * cos(q2) * sin(q2 + q3) * dq1 + m2 * l2^2 * sin(q2 + q3) * cos(q2 + q3) * dq1;
c32 = m2 * l1 * l2 * sin(q3) * dq2;
C = [c11, c12, c13; c21, c22, c23; c31, c32, 0];

G = [0; g * m1 * l1 * cos(q2) + g * m2 * l1 * cos(q2) + g * m2 * l2 * cos(q2 + q3); m2 *
g * l2 * cos(q2 + q3)];

%Senaldecontrol.
T = [T1; T2; T3];

%Friccionviscosa
f = [0.1 0 0; 0 0.2 0; 0 0 0.15];

%Vector de Perturbación
w = [w1; w2; w3];

%Estadosdelsistema.
Q1 = [q11; q12; q13];
Q2 = [q21; q22; q23];

%Respuestadelaplantaenlazocerrado.
Dq1 = Q1 + Ts * Q2;
Dq2 = Q2 + Ts * inv(M) * (-C * Q2 - G - f * Q2 + T + w);

dq11 = Dq1(1);
dq12 = Dq1(2);
dq13 = Dq1(3);
dq21 = Dq2(1);
dq22 = Dq2(2);
dq23 = Dq2(3);

```

7.2.2. Controlador

```

function [T1, T2, T3,x1] =
Control_PD(qd1,qd2,qd3,dqd1,dqd2,dqd3,ddqd1,ddqd2,ddqd3,q1,q2,q3,dq1,dq2,dq3)

% Parametros del Niryo
l1 = 0.16048; % m
l2 = 0.297; % m
m1 = 1; % Kg
m2 = 1.631; % Kg
J1 = 0.004676; %  $Kgm^2$ 
J2 = 0.01468; %  $Kgm^2$ 
J3 = 0.007468; %  $Kgm^2$ 
g = 9.81; %  $m/seg^2$ 
Ts = 0.001

% Fricción Viscosa
f = [0.1  0  0;  0  0.2  0;  0  0  0.15];

% Ganancia del Control PD
Kv = [4.4889  0  0;  0  4.3889  0;  0  0  4.2889];
Kp = [632.8277  0  0;  0  631.8277  0;  0  0  630.8277];

% Vector de Velocidad Angular Deseada
dqd = [dq1; dq2; dq3];

% Vector de Aceleración Angular Deseada
ddqd = [ddqd1; ddqd2; ddqd3];

% Error de Posición
x1 = [q1-qd1;  q2-qd2;  q3-qd3];

% Error de Velocidad
x2 = [dq1-dqd1;  dq2-dqd2;  dq3-dqd3];

% Obtener la dinámica del sistema
% Matriz de Inercia
M11 = m2*l1*l1*cos(qd2)*cos(qd2) + 2*m2*l1*l2*cos(qd2)*cos(qd2+qd3)
+ m2*l2*l2*cos(qd2+qd3)*cos(qd2+qd3) + m1*l1*l1*cos(qd2)*cos(qd2) + J1;
M22 = m1*l1*l1 + m2*l1*l1 + 2*m2*l1*l2*cos(qd3) + m2*l2*l2 + J2 + J3;
M23 = m2*l1*l2*cos(qd3) + m2*l2*l2 + J3;
M33 = m2*l2*l2 + J3;

M = [M11,  0,  0;  0,  M22,  M23;  0,  0,  M33];

```

```

% Matriz de Coriolis
C11 = -m2*l1*l1*sin(qd2)*cos(qd2)*dqd2 - m2*l1*l2*sin(qd2)*cos(qd2+qd3)*dqd2
- m1*l1*l1*sin(qd2)*cos(qd2)*dqd2 - m2*l1*l2*cos(qd2)*sin(qd2+qd3)*(dqd2+dqd3) -
m2*l2*l2*sin(qd2+qd3)*cos(qd2+qd3)*(dqd2+dqd3);
C12 = -m2*l1*l1*sin(qd2)*cos(qd2)*dqd1 - m2*l1*l2*cos(qd2)*sin(qd2+qd3)*dqd1
- m2*l1*l2*sin(qd2)*cos(qd2+qd3)*dqd1 - m2*l2*l2*sin(qd2+qd3)*cos(qd2+qd3)*dqd1 -
m1*l1*l1*sin(qd2)*cos(qd2)*dqd1;
C13 = -m2*l1*l2*cos(qd2)*sin(qd2+qd3)*dqd1 - m2*l2*l2*sin(qd2+qd3)*cos(qd2+qd3)*dqd1;
C21 = m2*l1*l1*sin(qd2)*cos(qd2)*dqd1 + m2*l1*l2*cos(qd2)*sin(qd2+qd3)*dqd1 +
m2*l1*l2*sin(qd2)*cos(qd2+qd3)*dqd1 + m2+l2*l2*sin(qd2+qd3)*cos(qd2+qd3)*dqd1
+ m1*l1*l1*sin(qd2)*cos(qd2)*dqd1;
C22 = -m2*l1*l2*sin(qd3)*dqd3;
C23 = -m2*l1*l2*sin(qd3)*(dqd2+dqd3);
C31 = m2*l1*l2*cos(qd2)*sin(qd2+qd3)*dqd1 + m2*l2*l2*sin(qd2+qd3)*cos(qd2+qd3)*dqd1;
C32 = m2*l1*l2*sin(qd3)*dqd2;

C = [C11, C12, C13; C21, C22, C23; C31, C32, 0];

% Vector de gravedad
gr1=0.0;
gr2=g*m1*l1*cos(qd2) + g*m2*l1*cos(qd2) + g*m2*l2*cos(qd2+qd3);
gr3=g*m2*l2*cos(qd2+qd3);

gr = [gr1; gr2; gr3];

% Señal de Control PD mas Compensación de la Dinámica de la planta
T = M*ddqd + C*dqd + gr + f*dqd - Kp*x1 - Kv*x2;

T1 = T(1);
T2 = T(2);
T3 = T(3);

```

7.2.3. Señal de referencia

```

function [qd1,qd2,qd3] = referencia(kT)
b1 = pi/4; b2 = pi/12; b3 = pi/20;
c1 = pi/4; c2 = pi/12; c3 = pi/20;
f1 = 0.1; f2 = 0.2; f3 = 0.3;
w1 = 2*pi*f1; w2 = 2*pi*f2; w3 = 2*pi*f3;

qd1 = b1*(1-exp(-2.0*kT^3)) + c1 * (1 - exp(-2.0*kT^3)) * sin(w1*kT);

```

$$qd2 = b2 * (1 - \exp(-2.0 * kT^3)) + c2 * (1 - \exp(-2.0 * kT^3)) * \sin(w2 * kT);$$

$$qd3 = b3 * (1 - \exp(-2.0 * kT^3)) + c3 * (1 - \exp(-2.0 * kT^3)) * \sin(w3 * kT);$$

7.3. Código de Coppelia

```
function sysCall_init()
file=io.open('prueba2.txt','w+') - crea el archivo de salida y permite la escritura recurrente

- -Handle articulaciones
e1=sim.getObjectHandle('NiryoOneJoint1')
e2=sim.getObjectHandle('NiryoOneJoint2')
e3=sim.getObjectHandle('NiryoOneJoint3')

- -Handle eslabones
link1 = sim.getObjectHandle('NiryoOneLink1')
link2 = sim.getObjectHandle('NiryoOneLink2')
link3 = sim.getObjectHandle('NiryoOneLink3')

- -masas de los eslabones
m1 = sim.getShapeMass(link1);
m2 = sim.getShapeMass(link2);
m3 = sim.getShapeMass(link3);

- -Tensor de inercia eslabones (momentos principales)
Inertia1=sim.getShapeInertia(link1)
Inertia2=sim.getShapeInertia(link2)
Inertia3=sim.getShapeInertia(link3)

- -Inercia Eslabón
I1 = Inertia1[9]; - -Izz   Ixx[1]   Iyy[5]   Izz[9]
I2 = Inertia2[1]; - -Ixx
I3 = Inertia3[1]; - -Ixx

-Longitudes eslabones
L1 = 1.2247e-1;
L2 = 2.9216e-1;
L3 = 1.6048e-1;

- -parámetros varios
tLast = 0;
a11 = 0.01 - -fricciones
a22 = 0.02
```



```

a33 = 0.015
g = 9.81 - -gravedad
uTau = 0.5*math.pi/180 - - umbral para bloquear el motor en grados
vel = 0.7
vel2 = 0.7
Kp1 = 632.8277 - -ganancias utilizables para sintonizar
Kv1 = 4.4889
Kp2 = 631.8277
Kv2 = 4.3889
Kp3 = 630.8277
Kv3 = 4.2889
k = 0;
qdeseadas = 1 - - El selector entre variante en el tiempo y no variante
dt = sim.getSimulationTimeStep()

- - Referencia variante en tiempo
b1 = math.pi/4; b2 = math.pi/12; b3 = math.pi/20;
c1 = math.pi/4; c2 = math.pi/12; c3 = math.pi/20;
f1 = 0.1; f2 = 0.2; f3 = 0.3;
w1 = 2*math.pi*f1; w2 = 2*math.pi*f2; w3 = 2*math.pi*f3;
t=0

if qdeseadas == 1 then

qd1 = b1*(1-math.exp(-2.0*k3)) + c1*(1 - math.exp(-2.0*k3))*math.sin(w1*k);
qd2 = b2*(1 - math.exp(-2.0*k3)) + c2*(1 - math.exp(-2.0*k3))*math.sin(w2*k);
qd3 = b3*(1 - math.exp(-2.0*k3)) + c3*(1 - math.exp(-2.0*k3))*math.sin(w3*k);

qd1_0 = qd1;
qd2_0 = qd2;
qd3_0 = qd3;

dqd1 = 6*b1*k2*math.exp(-2.0*k3) + 6*c1*k2*math.exp(-2.0*k3)*math.sin(w1*k) + c1*(1 - math.exp(-2.0*k3))*math.cos(w1*k)*w1;
dqd2 = 6*b2*k2*math.exp(-2.0*k3) + 6*c2*k2*math.exp(-2.0*k3)*math.sin(w2*k) + c2*(1 - math.exp(-2.0*k3))*math.cos(w2*k)*w2;
dqd3 = 6*b3*k2*math.exp(-2.0*k3) + 6*c3*k2*math.exp(-2.0*k3)*math.sin(w3*k) + c3*(1 - math.exp(-2.0*k3))*math.cos(w3*k)*w3;

dqd1_0 = dqd1;
dqd2_0 = dqd2;
dqd3_0 = dqd3;

```

```

ddqd1 = 12*b1*k*math.exp(-2.0*k3) - 36*b1*k4*math.exp(-2.0*k3) + 12*c1*k*
math.exp(-2.0*k3)*math.sin(w1*k) - 36*c1*k4*math.exp(-2.0*k3)*math.sin(w1*
t) + 12*c1*t2*math.exp(-2.0*t3)*math.cos(w1*t)*w1 - c1*(1 - math.exp(-2.0*
t3))*math.sin(w1*t)*w12;
ddqd2 = 12*b2*k*math.exp(-2.0*k3) - 36*b2*k4*math.exp(-2.0*k3) + 12*c2*k*
math.exp(-2.0*k3)*math.sin(w2*k) - 36*c2*k4*math.exp(-2.0*k3)*math.sin(w2*
t) + 12*c2*t2*math.exp(-2.0*t3)*math.cos(w2*t)*w2 - c2*(1 - math.exp(-2.0*
t3))*math.sin(w2*t)*w22;
ddqd3 = 12*b3*k*math.exp(-2.0*k3) - 36*b3*k4*math.exp(-2.0*k3) + 12*c3*k*
math.exp(-2.0*k3)*math.sin(w3*k) - 36*c3*k4*math.exp(-2.0*k3)*math.sin(w3*
t) + 12*c3*t2*math.exp(-2.0*t3)*math.cos(w3*t)*w3 - c3*(1 - math.exp(-2.0*
t3))*math.sin(w3*t)*w32;

```

```
elseif qdeseadas == 2 then
```

```
qd1 = 60*math.pi/180
```

```
qd2 = -45*math.pi/180
```

```
qd3 = 45*math.pi/180
```

```
dqd1 = 0
```

```
dqd2 = 0
```

```
dqd3 = 0
```

```
ddqd1 = 0
```

```
ddqd2 = 0
```

```
ddqd3 = 0
```

```
end
```

```
-- orientaciones iniciales
```

```
-- Calculando orientaciones iniciales
```

```
quat = sim.GetObjectQuaternion(e1, -1) -- Obtiene orientacion a traves de cuaterniones
```

```
cuatq1 = quat[1]; -- x
```

```
cuatq3 = quat[3]; -- y
```

```
cuatq2 = quat[2]; -- z
```

```
cuatq0 = quat[4]; -- w
```

```
q1_0 = sim.getJointPosition(e1)
```

```
qq1 = 0
```

```
quat2 = sim.GetObjectQuaternion(e2, -1) -- Obtiene orientacion a traves de cuaterniones
```

```
cuatq21 = quat2[1]; -- x
```

```
cuatq23 = quat2[3]; -- y
```

```
cuatq22 = quat2[2]; -- z
```

```
cuatq20 = quat2[4]; -- w
```

```
q2_0 = sim.getJointPosition(e2)
```

```
qq2 = 0
```

```
quat3 = sim.getObjectQuaternion(e3, -1) -- Obtiene orientacion a traves de cuaterniones
```

```
cuatq31 = quat3[1]; -- x
```

```
cuatq33 = quat3[3]; -- y
```

```
cuatq32 = quat3[2]; -- z
```

```
cuatq30 = quat3[4]; -- w
```

```
q3_0 = sim.getJointPosition(e3)
```

```
qq3 = 0
```

```
-- errores iniciales
```

```
eqd1 = q1_0 - qd1 -- Error qd1
```

```
eqd2 = q2_0 - qd2 -- Error qd2
```

```
eqd3 = q3_0 - qd3 -- Error qd3
```

```
deqd1 = qq1 - dqd1 -- Derivada del error qd1
```

```
deqd2 = qq2 - dqd2 -- Derivada del error qd2
```

```
deqd3 = qq3 - dqd3 -- Derivada del error qd3
```

```
eqd1_0 = eqd1;
```

```
eqd2_0 = eqd2;
```

```
eqd3_0 = eqd3;
```

```
q1 = q1_0;
```

```
q2 = q2_0;
```

```
q3 = q3_0;
```

```
if qdeseadas == 1 then
```

```
qd1 = b1*(1-math.exp(-2.0*(k*dt)3))+c1*(1-math.exp(-2.0*(k*dt)3))*math.sin(w1*(k*dt));
```

```
qd2 = b2*(1-math.exp(-2.0*(k*dt)3))+c2*(1-math.exp(-2.0*(k*dt)3))*math.sin(w2*(k*dt));
```

```
qd3 = b3*(1-math.exp(-2.0*(k*dt)3))+c3*(1-math.exp(-2.0*(k*dt)3))*math.sin(w3*(k*dt));
```

```
dqd1 = (qd1 - qd1_0)/dt;
```

```
qd1_0 = qd1;
```

```
dqd2 = (qd2 - qd2_0)/dt;
```

```
qd2_0 = qd2;
```

$$dq3 = (q3 - q3_0)/dt;$$

$$q3_0 = q3;$$

$$ddq1 = (dq1 - dq1_0)/dt;$$

$$dq1_0 = dq1;$$

$$ddq2 = (dq2 - dq2_0)/dt;$$

$$dq2_0 = dq2;$$

$$ddq3 = (dq3 - dq3_0)/dt;$$

$$dq3_0 = dq3;$$

```
elseif qdeseadas == 2 then
dq1 = 0 - qd1 punto
dq2 = 0 - qd2 punto
dq3 = 0 - qd3 punto
```

```
ddq1 = 0 - qd1 2 puntos
ddq2 = 0 - qd2 2 puntos
ddq3 = 0 - qd3 2 puntos
end
```

```
sim.setFloatSignal("qd1",qd1)
sim.setFloatSignal("qd2",qd2)
sim.setFloatSignal("qd3",qd3)
```

```
eqd1 = q1 - qd1 - Error qd1
eqd2 = q2 - qd2 - Error qd2
eqd3 = q3 - qd3 - Error qd3
```

- derivada del error discretizada

```
deqd1 = (eqd1 - eqd1_0)/dt - Derivada del error qd1
eqd1_0 = eqd1;
```

```
deqd2 = (eqd2 - eqd2_0)/dt - Derivada del error qd2
eqd2_0 = eqd2;
```

```
deqd3 = (eqd3 - eqd3_0)/dt - Derivada del error qd3
eqd3_0 = eqd3;
```

- - Calculo de elementos de la matriz m

```
m11 = m1*L1*L1*math.cos(qd2)* math.cos(qd2) + I1 + m2* (L1*math.cos(qd2)
```

```

+L2*math.cos(qd2+qd3));
m12 = 0;
m13 = 0;

m21 = 0;
m22 = m1*L1*L1 + m2*L1*L1 + 2*m2*L1*L2*math.cos(qd3) + I2 + I3;
m23 = m2*L2*L2 + m2*L1*L2*math.cos(qd3) + I3;

m31 = 0;
m32 = m2*L2*L2 + m2*L1*L2*math.cos(qd3) + I3;
m33 = m2*L2*L2 + I3;

- Calculo de elementos matriz C
c11 = -2*m2*(L1*math.cos(qd2)+L2*math.cos(qd2+qd3))*(L1*math.sin(qd2)*dqd2
+ L2*math.sin(qd2+qd3)*(dqd2+dqd3));
c12 = -2*m1*L1*L1*math.sin(qd2)*math.cos(qd2)*dqd1;
c13 = 0;

c21 = m1*L1*L1*math.sin(qd2)*math.cos(qd2)*dqd1 - m2*(L1*math.cos(qd2)
+L2*math.cos(qd2+qd3))*(-L1*math.sin(qd2)-L2*math.sin(qd2+qd3))*dqd1;
c22 = -2*m2*L1*L2*math.sin(qd3)*dqd3;
c23 = -m2*L1*L2*math.sin(qd3)*dqd3;

c31 = m2*(L1*math.cos(qd2)+L2*math.cos(qd2+qd3))*(L2*math.sin(qd2+qd3))*dqd1;
c32 = m2*L1*L2*math.sin(qd3)*(dqd2+dqd3);
c33 = 0;

- Calculo de elementos vector g
g1 = 0;
g2 = m1*L1*math.cos(qd2) + m2*L1*math.cos(qd2) + m2*L2*math.cos(qd2+qd3);
g3 = m2*L2*math.cos(qd2+qd3);

Tau1 = m11*ddqd1 + c11*dqd1 + c12*dqd2 + a11*dqd1 - Kp1*eqd1 - Kv1*deqd1;
Tau2 = m22*ddqd2 + m23*ddqd3 + c21*dqd1 + c22*dqd2 + c23*dqd3 + g*qd2 + a22*dqd2
- Kp2*eqd2 - Kv2*deqd2;
Tau3 = m32*ddqd2 + m33*ddqd3 + c31*dqd1 + c32*dqd2 + g*qd2 + a33*dqd3 - Kp3*eqd3
- Kv3*deqd3;

xx= m22*ddqd2 + m23*ddqd3 + c21*dqd1 + c22*dqd2 + c23*dqd3 + g*qd2 + a22*dqd2 -
Kp2*eqd2

if math.abs(eq1)<uTau then

```

```

else
if (Tau1 < 0) then
sim.setJointTargetVelocity(e1,-vel)
else
sim.setJointTargetVelocity(e1,vel)
end
end

if (math.abs(eqd2) < uTau) then
else
if Tau2 < 0 then
sim.setJointTargetVelocity(e2,-vel)
else
sim.setJointTargetVelocity(e2,vel)
end
end

if (math.abs(eqd3) < uTau) then
else
if Tau3 < -uTau then
sim.setJointTargetVelocity(e3,-vel2)
else
sim.setJointTargetVelocity(e3,vel2)
end
end
sim.setJointMaxForce(e1,math.abs(Tau1))
sim.setJointMaxForce(e2,math.abs(Tau2))
sim.setJointMaxForce(e3,math.abs(Tau3))

dt = sim.getSimulationTimeStep()

sim.addLog(sim.verbosity_scriptinfos,"q1=")
sim.addLog(sim.verbosity_scriptinfos,q1*180/math.pi)

    file:write(string.format(' %.1f%3.2f%3.3f%3.3f%3.3f%3.3f%3.3f%3.3f%3.3f%3.3f
n',k,dt*k, q1,q2,q3,qd1, qd2, qd3, Tau1, Tau2, Tau3)) - actualiza el archivo
end

function sysCall_cleanup()
file:close() - cierra edición del archivo
end

function sysCall_actuation()

```

```

t = sim.getSimulationTime()
k = k + 1 -- siguiente periodo
dt = sim.getSimulationTimeStep()
-- Calculando orientaciones
quat = sim.getObjectQuaternion(e1, -1) -- Obtiene orientacion a traves de cuaterniones
cuatq1 = quat[1]; -- x
cuatq3 = quat[3]; -- y
cuatq2 = quat[2]; -- z
cuatq0 = quat[4]; -- w
q1 = sim.getJointPosition(e1)
qq1 = (q1 - q1_0)/dt; -- derivada de q1 metodo de Euler
q1_0 = q1; -- q1 anterior

quat2 = sim.getObjectQuaternion(e2, -1) -- Obtiene orientacion a traves de cuaterniones
cuatq21 = quat2[1]; -- x
cuatq23 = quat2[3]; -- y
cuatq22 = quat2[2]; -- z
cuatq20 = quat2[4]; -- w
q2 = sim.getJointPosition(e2)
qq2 = (q2 - q2_0)/dt; -- derivada de q2 metodo de Euler
q2_0 = q2;

quat3 = sim.getObjectQuaternion(e3, -1) -- Obtiene orientacion a traves de cuaterniones
cuatq31 = quat3[1]; -- x
cuatq33 = quat3[3]; -- y
cuatq32 = quat3[2]; -- z
cuatq30 = quat3[4]; -- w
q3 = sim.getJointPosition(e3)
qq3 = (q3 - q3_0)/dt; -- derivada de q3 metodo de Euler
q3_0 = q3;

sim.setFloatSignal("q1", q1)
sim.setFloatSignal("q2", q2)
sim.setFloatSignal("q3", q3)

-- Cálculo de la trayectoria deseada
if qdeseadas == 1 then
qd1 = b1*(1-math.exp(-2.0*(k*dt)^3))+c1*(1-math.exp(-2.0*(k*dt)^3))*math.sin(w1*(k*dt));
qd2 = b2*(1-math.exp(-2.0*(k*dt)^3))+c2*(1-math.exp(-2.0*(k*dt)^3))*math.sin(w2*(k*dt));
qd3 = b3*(1-math.exp(-2.0*(k*dt)^3))+c3*(1-math.exp(-2.0*(k*dt)^3))*math.sin(w3*(k*dt));

```

```

(k * dt));
dqd1 = (qd1 - qd1_0)/dt;
qd1_0 = qd1;

dqd2 = (qd2 - qd2_0)/dt;
qd2_0 = qd2;

dqd3 = (qd3 - qd3_0)/dt;
qd3_0 = qd3;

ddq1 = (dqd1 - dqd1_0)/dt;
dqd1_0 = dqd1;

ddq2 = (dqd2 - dqd2_0)/dt;
dqd2_0 = dqd2;

ddq3 = (dqd3 - dqd3_0)/dt;
dqd3_0 = dqd3;

elseif qdeseadas == 2 then
dqd1 = 0 - -qd1 punto
dqd2 = 0 - -qd2 punto
dqd3 = 0 - -qd3 punto

ddqd1 = 0 - -qd1 2 puntos
ddqd2 = 0 - -qd2 2 puntos
ddqd3 = 0 - -qd3 2 puntos
end
sim.setFloatSignal("qd1",qd1)
sim.setFloatSignal("qd2",qd2)
sim.setFloatSignal("qd3",qd3)

eqd1 = q1 - qd1 - -Errorqd1
eqd2 = q2 - qd2 - -Errorqd2
eqd3 = q3 - qd3 - -Errorqd3

```

- -derivada del error discretizada deqd1 = (eqd1 - eqd1_0)/dt - -Derivada del error qd1
eqd1_0 = eqd1;

deqd2 = (eqd2 - eqd2_0)/dt - -Derivada del error qd2
eqd2_0 = eqd2;

- [30] Ortega, R., Praly, L., and Landau, I. (1985). Robustness of discrete-time direct adaptive controllers. *IEEE transactions on automatic control*, 30(12):1179–1187.
- [31] Osuna-Ibarra, L., Caballero-Barragán, H., Loukianov, A. G., and Bayro-Corrochano, E. (2019). Tracking control using optimal discrete-time h for mechanical systems: Applied to robotics. *Robotics and Autonomous Systems*, 119:201–208.
- [32] Paul, R. P. (1981). *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. Richard Paul.
- [33] Sage, H., De Mathelin, M., and Ostertag, E. (1999). Robust control of robot manipulators: a survey. *International Journal of control*, 72(16):1498–1522.
- [34] SLOTINE, J. J. and SASTRY, S. S. (1983). Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators†. *International Journal of Control*, 38(2):465–492.
- [35] Soltanpour, M. R., Moattari, M., et al. (2019). Voltage based sliding mode control of flexible joint robot manipulators in presence of uncertainties. *Robotics and Autonomous Systems*, 118:204–219.
- [36] Spong, M., Khorasani, K., and Kokotovic, P. (1987). An integral manifold approach to the feedback control of flexible joint robots. *IEEE Journal on Robotics and Automation*, 3(4):291–300.
- [37] Spong, M. W. and Vidyasagar, M. (1991). *Robot Dynamics and Control*. Wiley.
- [38] Spong, M. W. and Vidyasagar, M. (2008). *Robot dynamics and control*. John Wiley & Sons.
- [39] Tomizuka, M., Horowitz, R., and Landau, Y. (1982). On the use of model reference adaptive control techniques for mechanical manipulators. In *2nd IASTED Symposium on Identification, Control and Robotics, Davos, Switzerland*.
- [40] Utkin, V., Guldner, J., and Shi, J. (2017). *Sliding mode control in electro-mechanical systems*. CRC press.
- [41] Utkin, V. I. (2013). *Sliding modes in control and optimization*. Springer Science & Business Media.
- [42] Williams, B. (2014). An introduction to robotics. *Mech. and Cont. of Robotic Manipulators*.
- [43] Yates, D. R., Vaessen, C., and Roupret, M. (2011). From leonardo to da vinci: the history of robot-assisted surgery in urology. *BJU international*, 108(11):1708–1713.

- [13] Kelly, R., Davila, V. S., and Perez, J. A. L. (2005b). *Control of Robot Manipulators in Joint Space*. 1 edition.
- [14] Kelly, R. and Monroy, C. (2018). A brief tour on exotic control objectives in robotics. *Journal of Mechanics Engineering and Automation*, 8(2):53–59.
- [15] Kelly, R. and Monroy, C. (2019). On robustness of pd control with gravity compensation of torque-driven robot manipulators. In *2019 International Conference on Control, Automation and Diagnosis (ICCAD)*, pages 1–5. IEEE.
- [16] Kim, Y. and Lewis, F. (1999). Neural network output feedback control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 15(2):301–309.
- [17] Kotyczka, P. and Lefèvre, L. (2020). Discrete-time control design based on symplectic integration: Linear systems. *IFAC-PapersOnLine*, 53(2):7563–7568.
- [18] Kotyczka, P. and Thoma, T. (2021). Symplectic discrete-time energy-based control for nonlinear mechanical systems. *Automatica*, 133:109842.
- [19] Kumar, N., Porwal, A., Singh, A. R., Naskar, R., and Purwar, S. (2019). Event Triggered Control of Robot Manipulator. In *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 362–366.
- [20] Landau, I. (1985). Adaptive control techniques for robotic manipulators-the status of the art. *IFAC Proceedings Volumes*, 18(16):17–25.
- [21] Luh, J. Y., Walker, M. W., and Paul, R. P. (1980). On-line computational scheme for mechanical manipulators.
- [22] Monaco, S. and Normand-Cyrot, D. (1985). On the sampling of a linear analytic control system. In *1985 24th IEEE Conference on Decision and Control*, pages 1457–1462. IEEE.
- [23] Neuman, C. P. and Tourassis, V. D. (1985). Discrete dynamic robot models. *IEEE transactions on systems, man, and cybernetics*, (2):193–204.
- [24] Nicosia, S. and Tomei, P. (1985). A discrete-time mras control for industrial robots. In *IASTED International Symposium on Robotics and Automation*, pages 83–89.
- [25] Nicosia, S., Tomei, P., and Tornambe, A. (1986). Dynamic modelling of flexible robot manipulators. In *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, volume 3, pages 365–372. IEEE.
- [26] Nicosia, S., Tomei, P., and Tornambè, A. (1989a). Discrete-time modeling and control of robotic manipulators. *Journal of Intelligent and Robotic Systems*, 2(4):411–423.
- [27] Nicosia, S., Tomei, P., and Tornambè, A. (1989b). Discrete-time modeling and control of robotic manipulators. *Journal of Intelligent and Robotic Systems*, 2(4):411–423.
- [28] Ogata, K. (2003). *Ingeniería de control moderna*. Pearson Educación.
- [29] Orlov, Y. V. and Aguilar, L. T. (2014). *Advanced H control: Towards nonsmooth theory and applications*. Springer Science & Business Media.

Bibliografía

- [1] Aoues, S., Matignon, D., and Alazard, D. (2015). Control of a flexible spacecraft using discrete ida-pbc design. *IFAC-PapersOnLine*, 48(13):188–193.
- [2] Arimoto, S. (1996). Control theory of nonlinear mechanical systems. *A Passivity-based and Circuit-theoretic Approach*.
- [3] Cao, W.-J. and Xu, J.-X. (2004). Nonlinear integral-type sliding surface for both matched and unmatched uncertain systems. *IEEE Transactions on Automatic Control*, 49(8):1355–1360.
- [4] Cervantes, I. and Alvarez-Ramirez, J. (2001). On the PID tracking control of robot manipulators. *Systems Control Letters*, 42(1):37–46.
- [5] Chen, H.-T. and Rossi, P. H. (1987). The theory-driven approach to validity. *Evaluation and program planning*, 10(1):95–103.
- [6] Dai, Y., Shi, S., and Zheng, N. (1999). Adaptive control of robot manipulators with only position measurements. *IFAC Proceedings Volumes*, 32(2):665–670.
- [7] Darajat, A. U. and Istiqphara, S. (2020). Control of Two-Link Robot Manipulator with Uncertainty Parameter using Self-Tuning Sliding Mode Control. In *2020 3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT)*, pages 376–380.
- [8] Elices, J. (2020). Medieval robots? They were just one of this Muslim inventor’s creations.
- [9] Estrada, M. A., Fridman, L., and Moreno, J. A. (2020). Control of fully actuated mechanical systems via super-twisting based lyapunov redesign. *IFAC-PapersOnLine*, 53(2):5117–5121.
- [10] Hairer, E., Wanner, G., and Lubich, C. (2006). Symplectic integration of hamiltonian systems. In *Geometric Numerical Integration*, pages 179–236. Springer.
- [11] Hollerbach, J. M. (1980). A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity. *IEEE Transactions on Systems, Man, and Cybernetics*, 10(11):730–736.
- [12] Kelly, R., Davila, V. S., and Perez, J. A. L. (2005a). *Control of robot manipulators in joint space*. Springer Science & Business Media.

7.4.6. Cálculo de k_{h_1}

La constante k_{h_1} debe cumplir con la desigualdad

$$k_{h_1} \geq k_{C_1} \|\dot{q}_d\|_M$$

por lo tanto

$$k_{h_1} \geq 3.1004 \frac{\text{kg} \cdot \text{m}^2}{\text{s}}$$

7.4.7. Cálculo de k_{h_2}

Este parámetro está dado por la fórmula

$$k_{h_2} \geq \frac{s_2}{\tanh \frac{s_2}{s_1}}, \text{ donde}$$

$$s_1 = [k_g + k_M \|\ddot{q}_d\|_M + k_{C_2} \|\dot{q}_d\|_M^2]$$

$$s_2 = 2[k' + k'_M \|\ddot{q}_d\|_M + k_{C_1} \|\dot{q}_d\|_M^2]$$

Es así como k_{h_2} debe cumplir con la siguiente condición

$$k_{h_2} \geq 106.1685 \text{ Nm}$$

$$\frac{\partial C_{211}(q)}{\partial q_2} = (-\sin^2(q_2) + \cos^2(q_2))(m_1 l_1^2 + m_2 l_1^2 + 2m_2 l_1 l_2 [\cos(q_2) \cos(q_2 + q + 3) - \sin(q_2) \sin(q_2 + q + 3)])$$

$$\frac{\partial C_{211}(q)}{\partial q_3} = m_2 l_1 l_2 [\cos(q_2) \cos(q_2 + q_3) - \sin(q_2) \sin(q_2 + q_3) + m_2 l_2^2 [\cos^2(q_2 + q_3) - \sin^2(q_2 + q_3)]]$$

El cálculo de este parámetro está dado por la fórmula

$$k_{C_2} \geq n^3 \left(\max_{i,j,k,l,q} \left| \frac{\partial C_{k_{i,j}}(q)}{\partial q_l} \right| \right)$$

donde $\max_{i,j,k,l,q} \left| \frac{\partial C_{k_{i,j}}(q)}{\partial q_l} \right|$ resultó ser $\frac{\partial C_{121}(q)}{\partial q_2}$ por lo tanto, la constante k_{C_2} se obtendrá con:

$$k_{C_2} \geq n^3 \left[-2m_2 [l_1^2 (\cos^2(q_2) - \sin^2(q_2)) + l_1 l_2 [\cos(q_2) \cos(q_2 + q_3) - \sin(q_2) \sin(q_2 + q_3) + \cos(q_2) \cos(q_2 + q_3) - \sin(q_2) \sin(q_2 + q_3)] + l_2^2 (\cos^2(q_2 + q_3) - \sin^2(q_2 + q_3))] \right],$$

resultando de esta manera:

$$k_{C_2} \geq 3.5457 \text{ kg} \cdot \text{m}^2$$

7.4.5. Cálculo de k_g

Para el cálculo de esta constante, se toma en cuenta el vector de fuerzas gravitacionales y se obtiene la derivada parcial con respecto al vector de articulaciones $q(t)$, y utilizando la fórmula que se ve en seguida:

$$k_g \geq n \left(\max_{i,j,q} \left| \frac{\partial g_i(q)}{\partial q_j} \right| \right),$$

entonces la constante k_g se obtendrá de la siguiente manera:

$$\begin{array}{l} \frac{\partial g_2(q)}{\partial q_1} = 0 \quad \left| \quad \frac{\partial g_2(q)}{\partial q_2} = [-m_1 l_1 \sin(q_2) - m_2 l_1 \sin(q_2) - m_2 l_2 \sin(q_2 + q_3)]g \quad \right| \quad \frac{\partial g_2(q)}{\partial q_3} = -m_2 l_2 g \sin(q_2 + q_3) \\ \frac{\partial g_3(q)}{\partial q_1} = 0 \quad \left| \quad \frac{\partial g_3(q)}{\partial q_2} = -m_2 l_2 g \sin(q_2 + q_3) \quad \right| \quad \frac{\partial g_3(q)}{\partial q_3} = -m_2 l_2 g \sin(q_2 + q_3) \end{array}$$

$$k_g \geq n \left[\frac{\partial g_2(q)}{\partial q_2} \right],$$

$$k_g \geq n [-m_1 l_1 \sin(q_2) - m_2 l_1 \sin(q_2) - m_2 l_2 \sin(q_2 + q_3)]g,$$

resultando la constante con la siguiente desigualdad:

$$k_g \geq 15.9025 \frac{\text{kg} \cdot \text{m}^2}{\text{s}^2}$$

7.4.4. Cálculo de K_{C_2}

Para el cálculo de esta constante, se toma en cuenta $C(q, \dot{q})\dot{q}$ y la matriz Jacobiana:

$$\begin{array}{lll}
 \frac{\partial C_{111}(q)}{\partial q_1} = 0 & \frac{\partial C_{111}(q)}{\partial q_2} = 0 & \frac{\partial C_{111}(q)}{\partial q_3} = 0 \\
 \frac{\partial C_{112}(q)}{\partial q_1} = 0 & \frac{\partial C_{112}(q)}{\partial q_2} = -2m_1 l_1^2 \cos^2(q_2) & \frac{\partial C_{112}(q)}{\partial q_3} = 0 \\
 \frac{\partial C_{113}(q)}{\partial q_1} = 0 & \frac{\partial C_{113}(q)}{\partial q_2} = 0 & \frac{\partial C_{113}(q)}{\partial q_3} = 0 \\
 \frac{\partial C_{121}(q)}{\partial q_1} = 0 & \frac{\partial C_{121}(q)}{\partial q_2} = \text{Abajo} & \frac{\partial C_{121}(q)}{\partial q_3} = \text{Abajo} \\
 \frac{\partial C_{122}(q)}{\partial q_1} = 0 & \frac{\partial C_{122}(q)}{\partial q_2} = 0 & \frac{\partial C_{122}(q)}{\partial q_3} = 0 \\
 \frac{\partial C_{123}(q)}{\partial q_1} = 0 & \frac{\partial C_{123}(q)}{\partial q_2} = 0 & \frac{\partial C_{123}(q)}{\partial q_3} = 0 \\
 \frac{\partial C_{131}(q)}{\partial q_1} = 0 & \frac{\partial C_{131}(q)}{\partial q_2} = \text{Abajo} & \frac{\partial C_{131}(q)}{\partial q_3} = \text{Abajo} \\
 \frac{\partial C_{132}(q)}{\partial q_1} = 0 & \frac{\partial C_{132}(q)}{\partial q_2} = 0 & \frac{\partial C_{132}(q)}{\partial q_3} = 0 \\
 \frac{\partial C_{133}(q)}{\partial q_1} = 0 & \frac{\partial C_{133}(q)}{\partial q_2} = 0 & \frac{\partial C_{133}(q)}{\partial q_3} = 0 \\
 \frac{\partial C_{211}(q)}{\partial q_1} = 0 & \frac{\partial C_{211}(q)}{\partial q_2} = \text{Abajo} & \frac{\partial C_{211}(q)}{\partial q_3} = \text{Abajo} \\
 \frac{\partial C_{232}(q)}{\partial q_1} = 0 & \frac{\partial C_{232}(q)}{\partial q_2} = 0 & \frac{\partial C_{232}(q)}{\partial q_3} = -2m_2 l_1 l_2 \cos(q_3) \\
 \frac{\partial C_{233}(q)}{\partial q_1} = 0 & \frac{\partial C_{233}(q)}{\partial q_2} = 0 & \frac{\partial C_{233}(q)}{\partial q_3} = -m_2 l_1 l_2 \cos(q_3) \\
 \frac{\partial C_{121}(q)}{\partial q_2} = -2m_2 [l_1^2 (\cos^2(q_2) - \sin^2(q_2)) + l_1 l_2 [\cos(q_2) \cos(q_2 + q_3) \\
 - \sin(q_2) \sin(q_2 + q_3) + \cos(q_2) \cos(q_2 + q_3) - \sin(q_2) \sin(q_2 + q_3)] \\
 + l_2^2 (\cos^2(q_2 + q_3) - \sin^2(q_2 + q_3))] \\
 \frac{\partial C_{121}(q)}{\partial q_3} = -2m_2 [l_1 l_2 \cos(q_2) \cos(q_2 + q_3) - l_1 l_2 \sin(q_2) \sin(q_2 + q_3) \\
 - l_2^2 \cos^2(q_2 + q_3) - l_2^2 \sin^2(q_2 + q_3)] \\
 \frac{\partial C_{131}(q)}{\partial q_2} = -2m_2 [l_1 l_2 \cos(q_2) \cos(q_2 + q_3) - l_1 l_2 \sin(q_2) \sin(q_2 + q_3) \\
 + l_2^2 \cos^2(q_2 + q_3) - l_2^2 \sin^2(q_2 + q_3)] \\
 \frac{\partial C_{131}(q)}{\partial q_3} = -2m_2 [l_1 l_2 \cos(q_2) \cos(q_2 + q_3) + l_2^2 \cos^2(q_2 + q_3) - l_2^2 \sin^2(q_2 + q_3)]
 \end{array}$$

$$C(q, \dot{q})\dot{q} = \begin{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}^T & C_1(q) & \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \\ \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}^T & C_2(q) & \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \\ \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}^T & C_3(q) & \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \end{bmatrix}$$

donde $C_1(q)$ es el siguiente producto de vectores y matriz

$$C(q, \dot{q})\dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}^T \begin{bmatrix} 0 & -2m_1 l_1^2 \sin(q_2) \cos(q_2) & 0 \\ C_{21} & 0 & 0 \\ C_{31} & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$C_{21} = -2m_2(l_1^2 \cos(q_2) \sin(q_2) + l_1 l_2 \cos(q_2) \sin(q_2 + q_3) + l_1 l_2 \cos(q_2 + q_3) \sin(q_2) + l_2^2 \cos(q_2 + q_3) \sin(q_2 + q_3))$$

$$C_{31} = -2m_2(l_1 l_2 \cos(q_2) \sin(q_2 + q_3) + l_2^2 \cos(q_2 + q_3) \sin(q_2 + q_3))$$

$C_2(q)$ es el siguiente producto de vectores y matriz

$$C(q, \dot{q})\dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}^T \begin{bmatrix} C_{11} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -2m_2 l_1 l_2 \sin(q_3) & -m_2 l_1 l_2 \sin(q_3) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$C_{11} = m_1 l_1^2 \sin(q_2) \cos(q_2) + m_2 l_1^2 \sin(q_2) \cos(q_2) + m_2 l_1 l_2 \cos(q_2) \sin(q_2 + q_3) + m_2 l_1 l_2 \sin(q_2) \cos(q_2 + q_3) + m_2 l_2^2 \sin(q_2 + q_3) \cos(q_2 + q_3)$$

Y $C_3(q)$ es el siguiente producto de vectores y matriz

$$C(q, \dot{q})\dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}^T \begin{bmatrix} C_{11} & 0 & 0 \\ 0 & m_2 l_1 l_2 \sin(q_3) & 0 \\ 0 & m_2 l_1 l_2 \sin(q_3) & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$C_{11} = m_2 l_1 l_2 \cos(q_2) \sin(q_2 + q_3) + m_2 l_2^2 \sin(q_2 + q_3) \cos(q_2 + q_3)$$

La constante k_{C_1} se determina de la siguiente manera:

$$k_{C_1} \geq n^2 \left(\max_{i,j,k,\mathbf{q}} |C_{kij}(q)| \right)$$

la constante k_{C_1} resultará de calcular lo siguiente:

$$k_{C_1} \geq n^2 |-2m_2 l_1 l_2 \cos(q_2) \sin(q_2 + q_3) - 2m_2 l_2^2 \cos(q_2 + q_3) \sin(q_2 + q_3)|$$

por lo tanto, la constante k_{C_1} está determinada como:

$$k_{C_1} \geq 0.7751 \text{ kg} \cdot \text{m}^2 \quad (7.1)$$


```
xx= m22*ddqd2 + m23*ddqd3 + c21*dqd1 + c22*dqd2 + c23*dqd3 + g*qd2 + a22*dqd2 -
Kp2*eqd2
```

```
if math.abs(eqd1)<uTau then
else
```

```
if (Tau1 < 0) then
sim.setJointTargetVelocity(e1,-vel)
else
sim.setJointTargetVelocity(e1,vel)
end
end
```

```
if (math.abs(eqd2) < uTau) then
else
if Tau2 < 0 then
sim.setJointTargetVelocity(e2,-vel)
else
sim.setJointTargetVelocity(e2,vel)
end
end
```

```
if (math.abs(eqd3) < uTau) then
else
if Tau3 < -uTau then
sim.setJointTargetVelocity(e3,-vel2)
else
sim.setJointTargetVelocity(e3,vel2)
end
end
```

```
sim.setJointMaxForce(e1,math.abs(Tau1))
sim.setJointMaxForce(e2,math.abs(Tau2))
sim.setJointMaxForce(e3,math.abs(Tau3))
```

```
t_text=sim.getSimulationTime()
if t_text>tLast+1 then
tau=sim.getObjectSizeValues(link1)
sim.addLog(sim.verbosity_scriptinfos,"tau 3=")
sim.addLog(sim.verbosity_scriptinfos,q3*180/math.pi)- -q1*180/math.pi)
sim.addLog(sim.verbosity_scriptinfos,"tau 2=")
sim.addLog(sim.verbosity_scriptinfos,math.deg(q2))- -q2*180/math.pi)
tLast=t_text
end
```

deqd3 = (eqd3 - eqd3_0)/dt - -Derivada del error qd3

eqd3_0 = eqd3;

- - Cálculo de elementos de la matriz m

m11 = m1*L1*L1*math.cos(qd2)* math.cos(qd2) + I1 + m2* (L1*math.cos(qd2)
+L2*math.cos(qd2+qd3));

m12 = 0;

m13 = 0;

m21 = 0;

m22 = m1*L1*L1 + m2*L1*L1 + 2*m2*L1*L2*math.cos(qd3) + I2 + I3;

m23 = m2*L2*L2 + m2*L1*L2*math.cos(qd3) + I3;

m31 = 0;

m32 = m2*L2*L2 + m2*L1*L2*math.cos(qd3) + I3;

m33 = m2*L2*L2 + I3;

- -Calculo de elementos matriz C

c11 = -2*m2*(L1*math.cos(qd2)+L2*math.cos(qd2+qd3))*(L1*math.sin(qd2)*dqd2
+ L2*math.sin(qd2+qd3)*(dqd2+dqd3));

c12 = -2*m1*L1*L1*math.sin(qd2)*math.cos(qd2)*dqd1;

c13 = 0;

c21 = m1*L1*L1*math.sin(qd2)*math.cos(qd2)*dqd1 - m2*(L1*math.cos(qd2)
+L2*math.cos(qd2+qd3))*(-L1*math.sin(qd2)-L2*math.sin(qd2+qd3))*dqd1;

c22 = -2*m2*L1*L2*math.sin(qd3)*dqd3;

c23 = -m2*L1*L2*math.sin(qd3)*dqd3;

c31 = m2*(L1*math.cos(qd2)+L2*math.cos(qd2+qd3))*(L2*math.sin(qd2+qd3))*dqd1;

c32 = m2*L1*L2*math.sin(qd3)*(dqd2+dqd3);

c33 = 0;

- -Calculo de elementos vector g

g1 = 0;

g2 = m1*L1*math.cos(qd2) + m2*L1*math.cos(qd2) + m2*L2*math.cos(qd2+qd3);

g3 = m2*L2*math.cos(qd2+qd3);

Tau1 = m11*ddqd1 + c11*dqd1 + c12*dqd2 + a11*dqd1 - Kp1*eqd1 - Kv1*deqd1;

Tau2 = m22*ddqd2 + m23*ddqd3 + c21*dqd1 + c22*dqd2 + c23*dqd3 + g*qd2 + a22*dqd2
- Kp2*eqd2 - Kv2*deqd2;

Tau3 = m32*ddqd2 + m33*ddqd3 + c31*dqd1 + c32*dqd2 + g*qd2 + a33*dqd3 - Kp3*eqd3
- Kv3*deqd3;