

Universidad Católica de Santa María
Facultad de Ciencias e Ingenierías Físicas y
Formales
Escuela de Profesional de Ingeniería Mecánica,
Eléctrica y Mecatrónica



**ANÁLISIS DE DESEMPEÑO EN SIMULACIÓN DE UN
CONTROLADOR ADAPTATIVO EN EL CONTROL DE
TRAYECTORIA DE UN MANIPULADOR ROBOTICO DE 2 GRADOS
DE LIBERTAD**

Tesis presentada por el Bachiller:

**Chavez Cornejo, Rodrigo
Javier Antonio**

para optar el Título Profesional de
Ingeniero Mecatrónico

Asesor (a):

**Ing. Quispe Ccachuco, Marcelo
Jaime**

Arequipa – Perú

2022

UCSM-ERP

UNIVERSIDAD CATÓLICA DE SANTA MARÍA
INGENIERIA MECANICA, MECANICA-ELECTRICA Y MECATRONICA
TITULACIÓN CON TESIS
DICTAMEN APROBACIÓN DE BORRADOR

Arequipa, 02 de Junio del 2022

Dictamen: 002987-C-EPIMMEM-2022

Visto el borrador del expediente 002987, presentado por:

2015220781 - CHAVEZ CORNEJO RODRIGO JAVIER ANTONIO

Titulado:

**ANALISIS DE DESEMPEÑO EN SIMULACIÓN DE UN CONTROLADOR ADAPTATIVO EN EL
CONTROL DE TRAYECTORIA DE UN MANIPULADOR ROBOTICO DE 2 GRADOS DE LIBERTAD**

Nuestro dictamen es:

APROBADO

**1936 - MESTAS RAMOS SERGIO ORLANDO
DICTAMINADOR**



**2397 - CUADROS MACHUCA JUAN CARLOS
DICTAMINADOR**



**2776 - COLLADO OPORTO CHRISTIAM GUILLERMO
DICTAMINADOR**



Dedicatoria

La presente Tesis está dedicada a mi madre Lilian por ser un gran modelo a seguir, un gran ser humano y que siempre me brindó su apoyo incondicionalmente.

A mi padre Jorge y mi hermana Carolina por el cariño y tiempo, por los consejos y por acompañarme en cada momento.

A mis familiares, amigos y todos aquellos que me impulsaron a continuar y superarme cada día.



Agradecimiento



A mi familia y amigos que siempre estuvieron presentes en cada etapa, apoyando aconsejando y motivando a ser mejor cada día.

Agradezco a mis maestros, compañeros, y a la Universidad por todo lo anterior y por el conjunto de conocimientos brindados que fueron esenciales para seguir creciendo.

RESUMEN

La presente tesis abarca el diseño de un controlador adaptativo relacionado a la inercia para su posterior análisis y comparación respecto a los tipos de controladores como PD, PD aumentado y Torque computado. Siendo evaluados en diferentes tipos de trayectoria, una curva, una circunferencia y una trayectoria con un cambio de dirección; para poder comparar cómo se comportan cada uno de estos controladores con las distintas trayectorias y cuál de ellos presenta la mejor opción para el control de trayectorias para un robot de 2 DOF, tomando como principal indicador la cuantificación del error.

Para el diseño del controlador adaptativo se utiliza la propiedad de la linealidad de los parámetros dinámicos del robot, de la cual obtenemos las ecuaciones de control adaptativo relacionado a la inercia y colocando la variable desconocida como la masa del link número 2, de la misma manera se supone una incertidumbre en el link número 2 del robot de un 50% el cuál utilizaremos como variable para el control adaptativo. De esta manera se propone un controlador que es capaz de asumir incertidumbres en el link número 2 del robot y es capaz de alcanzar la trayectoria en un tiempo mucho menor que los controladores con los cuales se compara sin la necesidad de contar con un modelamiento exacto del robot, que es lo que suponen los controladores PD, Torque computado o un controlador PID para el correcto funcionamiento de estos. El cuál se evalúa respecto a una trayectoria deseada y el error de posición de las juntas.

Palabras claves:

Adaptativo, Seguimiento de trayectoria, No lineal

ABSTRACT

The present thesis covers the design of an adaptive controller related to inertia for its subsequent analysis and comparison with respect to the types of controllers such as PD, increased PD, computed torque. Being evaluated in different types of trajectory, a curve, a circumference and a trajectory with a change of direction; to be able to compare how each of these controllers behave with the different trajectories and which one of them presents the best option for trajectory control for a 2 DOF robot. For the design of the adaptive controller, the property of linearity of the dynamic parameters of the robot is used, from which we obtain the adaptive control equations related to inertia and placing the unknown variable as the mass of link number 2, in the same way An uncertainty in the link number 2 of the robot of 50% is assumed, which we will use as a variable for adaptive control. In this way, a controller is proposed that is capable of assuming uncertainties in the robot's link number 2 and is capable of reaching the trajectory in a much shorter time than the controllers with which it is compared without the need for an exact modeling of the robot. robot, which is what PD controllers, computed torque or a PID controller suppose for their correct operation. Which is evaluated with respect to a desired trajectory and the position error of the joints

Key words:

Adaptative, Trajectory tracking, Non linear

ÍNDICE

RESUMEN.....	v
ABSTRACT.....	vi
ÍNDICE DE TABLAS.....	10
ÍNDICE DE FIGURAS.....	11
INTRODUCCIÓN.....	14
CAPITULO I.....	17
MARCO METODOLOGICO.....	17
1. IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA.....	17
1.1 ENUNCIADO DEL PROBLEMA.....	17
1.2 DESCRIPCIÓN DEL PROBLEMA.....	17
1.3 JUSTIFICACIÓN DE LA INVESTIGACIÓN.....	18
1.4 ANTECEDENTES.....	19
2. OBJETIVOS.....	23
2.1 OBJETIVO PRINCIPAL.....	23
2.2 OBJETIVOS ESPECIFICOS.....	23
3. HIPOTESIS.....	23
4. VARIABLES.....	24
4.1 VARIABLES DEPENDIENTE.....	24
4.2 VARIABLE INDEPENDIENTE.....	24
5. INDICADORES.....	24
6. ALCANCES Y LIMITACIONES.....	24
CAPITULO II.....	26
MARCO TEORICO.....	26
1. MODELAMIENTO DEL MANIPULADOR ROBOTICO DE 2 GRADOS DE LIBERTAD.....	26
1.1 MANIPULADOR ROBÓTICO.....	26
1.2 CONFIGURACIONES CLÁSICAS DE UN MANIPULADOR ROBÓTICO.....	26
1.2.1 ROBOT CARTESIANO.....	26
1.2.2 ROBOT CILÍNDRICO.....	26
1.2.3 ROBOT ESFÉRICO O POLAR.....	27
1.2.4 ROBOT DE BRAZO ARTICULADO O SCARA.....	27
1.2.5 ROBOT ANTROPOMÓRFICO.....	27
3. DEFINICIONES.....	28

4. CINEMATICA.....	29
4.1 CINEMATICA DIRECTA	29
4.1.1 CONVENCIÓN DENAVIT-HARTENBERG (DH).....	30
4.2 CINETICA INVERSA.....	34
5. DINÁMICA	35
5.1 MODELAMIENTO DINAMICO.....	35
5.1.1 METODO DE EULER- LAGRANGE	36
5.2 MODELAMIENTO DINÁMICO DE UN MANIPULADOR ROBÓTICO DE DOS GRADOS DE LIBERTAD	39
5.2.1 PROPIEDADES NOTABLES DEL MODELO DINÁMICO	41
6. SISTEMAS DE CONTROL	44
6.1 SISTEMA DE CONTROL EN LAZO CERRADO	44
7. CONTROL DE MOVIMIENTO.....	45
7.1 CONTROL DE SEGUIMIENTO DE TRAYECTORIAS.....	46
7.2 CONTROLADOR PD	47
7.3 CONTROLADOR PD+ o PD AUMENTADO.....	48
7.4 CONTROLADOR POR TORQUE-COMPUTADO.....	49
7.5 CONTROL ADAPTATIVO	50
8. ¿PORQUÉ CONTROL ADAPTATIVO?.....	52
9. EL PROBLEMA DEL CONTROL ADAPTATIVO	53
10. ÍNDICES DE DESEMPEÑO	53
10.1 ISE	54
10.2 IAE	54
10.3 ITAE.....	55
CAPITULO III	57
DESARROLLO DE LA INGENIERIA.....	57
1. DISEÑO DE CONTROLADORES.....	57
1.1 PROPORCIONAL+ DERIVATIVO PD	59
1.2 PD AUMENTADO	62
1.3 TORQUE COMPUTADO.....	63
1.4 CONTROL ADAPTATIVO	64
2 CONSIDERACIONES PREVIAS PARA LA SIMULACIÓN	67
3 DIAGRAMA DE FLUJO DEL CÓDIGO MATLAB	70
CAPITULO IV.....	73
RESULTADOS Y DISCUSION.....	73

1. TRAYECTORIA CURVA SUAVIZADA	73
1.1 TRAYECTORIA.....	73
1.2 ERRORES DE POSICIÓN	75
1.3 ERRORES DE VELOCIDAD.....	78
1.4 TORQUES APLICADOS.....	81
2. TRAYECTORIA CIRCULAR CON RADIO REDUCIDO	83
2.1 TRAYECTORIA.....	83
2.2 ERRORES DE POSICIÓN	86
2.3 ERRORES DE VELOCIDAD.....	88
2.4 TORQUES APLICADOS.....	92
3. TRAYECTORIA LINEAL CON CAMBIO DE DIRECCIÓN	95
3.1 TRAYECTORIA.....	95
3.2 ERRORES DE POSICIÓN	98
3.3 ERRORES DE VELOCIDAD.....	100
3.4 TORQUES APLICADOS.....	104
DISCUSIÓN.....	107
CONCLUSIONES.....	109
RECOMENDACIONES Y TRABAJO FUTURO.....	111
REFERENCIAS.....	112

ÍNDICE DE TABLAS

Tabla 1: Resumen de parámetros DH.....	33
Tabla 2: Indicadores de rendimiento para la trayectoria suavizada.....	82
Tabla 3: Indicadores de rendimiento para la trayectoria circular	93
Tabla 4: Indicadores de rendimiento para la trayectoria lineal	105
Tabla 4: Índices de error par el control adaptativo.....	110



ÍNDICE DE FIGURAS

Figura 1. Respuesta de un sistema de segundo orden al escalón unitario con diferentes valores de amortiguamiento.....	20
Figura 2. Respuesta al escalón unitario de un sistema de control	21
Figura 3: Configuraciones básicas de robots industriales.....	28
Figura 4: Manipulador de dos grados de libertad.....	30
Figura 5: Convención introducida por Denavit y Hartenberg para la localización en un eje de coordenadas.....	31
Figura 6: Robot de dos grados de libertad.....	32
Figura 7: Diagrama de relación entre cinemática directa e inversa	35
Figura 8: Manipulador robótico con vector de coordenadas generalizadas.....	39
Figura 10: Sistema de control en lazo cerrado	44
Figura 11: Controlador de reducción de error.	46
Figura 12: Sistema de control PD	48
Figura 13: Diagrama de bloques del control adaptativo	52
Figura 14: Manipulador robótico de dos grados de libertad.	57
Figura 15: Diagrama de Flujo del algoritmo usado	71
Figura 16. Trayectoria suavizada con controlador PD	73
Figura 17. Trayectoria suavizada con controlador PD aumentado	73
Figura 18. Trayectoria suavizada con controlador Torque Computado	74
Figura 19. Trayectoria suavizada con controlador adaptativo	74
Figura 20. Error de posición con trayectoria suavizada con controlador PD	75
Figura 21. Error de posición con trayectoria suavizada con controlador PD aumentado	76
Figura 22. Error de posición con trayectoria suavizada con controlador Torque Computado.....	76
Figura 23. Error de posición con trayectoria suavizada con controlador adaptativo.....	77
Figura 24. Error de velocidad con trayectoria suavizada con controlador PD ..	78
Figura 25. Error de velocidad con trayectoria suavizada con controlador PD aumentado	78
Figura 26. Error de velocidad con trayectoria suavizada con controlador Torque computado	79
Figura 27. Error de velocidad con trayectoria suavizada con controlador adaptativo.....	79

Figura 28. Torque aplicado con trayectoria suavizada con controlador PD	81
Figura 29. Torque aplicado con trayectoria suavizada con controlador PD aumentado	81
Figura 30. Torque aplicado con trayectoria con controlador Torque computado	82
Figura 31. Torque aplicado con trayectoria suavizada con controlador adaptativo	82
Figura 32 Trayectoria circular con controlador PD	83
Figura 33. Trayectoria circular con controlador PD aumentado	84
Figura 34 Trayectoria circular con controlador Torque computado	84
Figura 35. Trayectoria circular con controlador adaptativo	85
Figura 36. Error de posición con trayectoria circular con controlador PD	86
Figura 37. Error de posición con trayectoria circular con controlador PD aumentado	86
Figura 38. Error de posición con trayectoria circular con controlador Torque computado	87
Figura 39. Error de posición con trayectoria circular con controlador adaptativo	87
Figura 40. Error de velocidad con trayectoria circular con controlador PD.....	88
Figura 41. Error de velocidad con trayectoria circular con controlador PD aumentado	89
Figura 42. Error de velocidad con trayectoria circular con controlador Torque computado	89
Figura 43. Error de velocidad con trayectoria circular con controlador adaptativo	90
Figura 44. Torque con trayectoria circular con controlador PD.....	92
Figura 45. Torque con trayectoria circular con controlador PD +.....	93
Figura 46. Torque con trayectoria circular con controlador Torque computado	93
Figura 47. Torque con trayectoria circular con controlador adaptativo	94
Figura 48. Trayectoria lineal con cambio de dirección con controlador PD	95
Figura 49. Trayectoria lineal con cambio de dirección con controlador PD aumentado	95
Figura 50. Trayectoria lineal con cambio de dirección con controlador Torque Computado.....	96
Figura 51. Trayectoria lineal con cambio de dirección con controlador adaptativo	96
Figura 52. Error de posición en Trayectoria lineal con cambio de dirección con controlador PD.....	98

Figura 53. Error de posición en Trayectoria lineal con cambio de dirección con controlador PD aumentado.....	98
Figura 54. Error de posición en Trayectoria lineal con cambio de dirección con controlador Torque computado	99
Figura 55. Error de posición en Trayectoria lineal con cambio de dirección con adaptativo.....	99
Figura 56. Error de velocidad en Trayectoria lineal con cambio de dirección con controlador PD.....	100
Figura 57. Error de velocidad en Trayectoria lineal con cambio de dirección con controlador PD aumentado.....	101
Figura 58. Error de velocidad en Trayectoria lineal con cambio de dirección con controlador Torque computado	101
Figura 59. Error de velocidad en Trayectoria lineal con cambio de dirección con controlador adaptativo.....	102
Figura 60. Torque en Trayectoria lineal con cambio de dirección con controlador PD	104
Figura 61. Torque en Trayectoria lineal con cambio de dirección con controlador PD aumentado	105
Figura 62. Torque en Trayectoria lineal con cambio de dirección con controlador Torque Computado	105
Figura 63. Torque en Trayectoria lineal con cambio de dirección con controlador adaptativo.....	106

INTRODUCCIÓN

Hoy en día podemos encontrar sistemas robóticos en todas partes, desde industrias o centros de investigación hasta hogares, los cuales pueden ser de diferentes tipos y clases, tales como robots móviles, de exploración, de servicio, brazos robóticos, etc. Dentro de los cuales, los más utilizados son los manipuladores robóticos, especialmente en la industria, donde son ampliamente requeridos para automatizar diferentes tareas debido a la gran capacidad de adaptación en tareas repetitivas realizándolas de forma precisa y eficiente, automatizando, de esta forma, diversos procesos de manufactura en empresas.

Sin embargo, muchas de estas tareas realizadas por los manipuladores robóticos, dependen de un óptimo control de la trayectoria, debido a que podría afectar el desempeño del proceso y/o involucrar dispositivos cercanos que se pudiesen ver afectados. Para ello es necesaria la implementación de controladores capaces de lidiar con este tipo de desafíos y que además puedan tener en consideración las incertezas y perturbaciones que pudiesen surgir durante la realización de las tareas.

Durante las últimas décadas, diferentes tipos de control han sido utilizadas para el control de trayectoria de manipuladores robóticos, uno de los cuales es el clásico control tipo PID que es utilizado normalmente en diferentes tareas. Sin embargo, es bien sabido que los manipuladores poseen una dinámica con componentes altamente no lineales que podrían afectar el desempeño de este tipo de control clásico, complicando de esta forma, la utilización del controlador PID para cierto tipo de escenarios donde es requerido un control óptimo del manipulador. Por otra parte, otras estrategias de control, tales como el de PID aumentado o torque computado, han sido propuestas para compensar las no linealidades del modelo de robot convirtiéndose, de esta forma en controladores mucho más robustos y confiables. No obstante, el éxito del funcionamiento de este tipo de controladores depende en gran medida del modelamiento exacto del manipulador robótico, lo cual implica que las incertidumbres existentes debido a parámetros mal estimados o errores en el modelo del robot podrían provocar resultados no deseados. Por este motivo, diversos controladores no lineales han sido desarrollados especialmente para lidiar con este tipo de situaciones de incertidumbres paramétricas, tal es el caso del control adaptativo, control robusto y el control por modo deslizante.

Por todo lo expuesto, este trabajo de investigación presenta un análisis comparativo de diversos controladores lineales basados en modelo y de igual manera se centra en el desarrollo de un controlador no lineal de tipo adaptativo para el control de trayectoria en un manipulador robótico de dos grados de libertad que sea capaz de tener en consideración incertidumbres paramétricas en el modelo del robot.





CAPITULO I

MARCO METODOLOGICO

CAPITULO I

MARCO METODOLOGICO

1. IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA

1.1 ENUNCIADO DEL PROBLEMA

En la industria existen distintos tipos de procesos los cuales son no lineales, para realizar el control de estos se hace una linealización o una aproximación, la cual es funcional dependiendo del tipo de proceso, pero existen aquellos que requieren de altas precisiones o de altas velocidades de actuación o ambas. Por ello es necesario de implementar nuevos tipos de control a los convencionales, aplicando un control no lineal como el control robusto o control adaptativo el cual responderá a las distintas variaciones de la planta y a su no linealidad.

Debido a esta problemática es necesario el diseño y el análisis de desempeño de un controlador adaptativo en el control de distintos tipos de trayectoria de un manipulador robótico de 2 grados de libertad, donde se evaluará el desempeño de cada uno de ellos y sus distintas consideraciones para un proceso altamente no lineal con la posibilidad de realizarse cambios en la planta.

1.2 DESCRIPCIÓN DEL PROBLEMA

Los manipuladores robóticos son sistemas dinámicos compuestos de parámetros altamente variables convirtiéndose de esta forma en sistemas muy complejos, algunos de estos parámetros incluyen: la inercia, tamaño de los links, masa de los links entre otros, esta variabilidad se debe principalmente debido a la interacción del robot con el ambiente de trabajo en el cual se encuentra realizando las diferentes tareas.

Adicionalmente, para realizar el control de estos sistemas complejos, es necesario contar con un modelo matemático que permita describir con alta fidelidad la dinámica de este sistema a fin de modelar las diferentes estrategias de control para las tareas que debe realizar el manipulador robótico. Este modelamiento puede ser realizado de diferentes formas, de las cuales, el enfoque más utilizado es aquel que aprovecha las ecuaciones de dinámica del robot conocido como la formulación de Euler-Lagrange. Por lo tanto, lo que se busca al momento de diseñar las estrategias de control del sistema es lograr una estabilidad, robustez frente a

perturbaciones y el menor error posible frente a un setpoint predeterminado. A pesar de que diferentes tareas realizadas por los manipuladores robóticos pueden ser completadas exitosamente utilizando controladores lineales, como por ejemplo tareas de regulación, es necesario saber que existen otras que difícilmente podrán ser ejecutadas utilizando controladores de este tipo, especialmente cuando el robot realiza tareas de seguimiento de trayectorias o cuando existen cambios en sus parámetros debido a factores externos, por ejemplo cuando el robot tiene una carga con masa incierta. En este contexto, es necesario diseñar una estrategia de control considerando las no linealidades del sistema y compararlo con distintas estrategias de control clásicas.

1.3 JUSTIFICACIÓN DE LA INVESTIGACIÓN

Actualmente el análisis y desarrollo de controladores no lineales es un objeto de estudio en diversos laboratorios y empresas debido al desafío que representa la gran diversidad de tareas en los que pueden ser utilizados los manipuladores robóticos. Este trabajo de investigación se justifica porque enfrenta el problema de control de trayectoria de controladores robóticos que es vagamente abordado en la bibliografía, especialmente ante la presencia de incertezas paramétricas, puesto a que se aborda mayormente el control de posición en este tipo de sistemas. De igual manera, este trabajo presenta una comparativa del desempeño de diferentes controladores lineales descritos en la literatura y diferentes trayectorias, diseñados para compensar las no linealidades de los manipuladores robóticos, en contraposición con el controlador no lineal diseñado en esta investigación de tipo adaptativo utilizado para el control de trayectoria.

En este contexto, la justificación principal del presente trabajo de investigación se enfoca principalmente en el diseño y desarrollo de un controlador no lineal para el área de control robótico y sus implicancias en el funcionamiento ante diferentes tipos de trayectoria (línea suavizada, una circunferencia de radio reducido y una lineal con un cambio de dirección) para demostrar de este modo el desempeño superior de este tipo de controlador en las situaciones antes mencionadas y así pueda ser aplicado en dispositivos reales para realizar tareas que requieran de seguimiento de trayectorias de alta precisión. (Duhart, 2014)

1.4 ANTECEDENTES

Los manipuladores robóticos al ser dispositivos ampliamente utilizados en manufactura o procesos industriales, requieren de controladores que puedan afrontar los diferentes desafíos que se puedan presentar al momento de la realización de diferentes tareas, especialmente en aquellas que requieren de un seguimiento óptimo de trayectoria a diferentes velocidades o en tiempos específicos, además de tener en consideración la presencia de incertezas paramétricas y/o perturbaciones debido a un mal modelamiento del manipulador o por la existencia de condiciones que no fueron consideradas al momento de la implementación del controlador tal como A. Ishiguro, T. Furuhashi, S. Okuma, Y. Uchikawa (1992) A pesar de que diferentes controladores de tipo lineal han sido desarrollados enfocados especialmente en lidiar con estas incertezas compensándolas en el modelamiento de la ley de control, Poved (2012), la implementación de controladores no lineales como el control adaptativo, control robusto y el control por modo deslizante que según Liu Fan and E. M. Joo (2009), han presentado un mejor desempeño.

El problema de seguimiento de trayectoria en manipuladores robóticos no es reciente, diferentes trabajos han sido desarrollados abordando este problema tal como Nguyen, Kim D., Dankowicz, Harry (2014) que desarrolla un controlador adaptativo para manipuladores robóticos con alta tasa de estimación de los parámetros garantizando la robustez del sistema demostrando la habilidad del controlador para lidiar con las incertezas paramétricas y la presencia de ruido en el sistema.

O por ejemplo, B. Rahmani, M. Belkheiri (2016) que presenta un controlador adaptativo híbrido PD-red neuronal para el problema del seguimiento de trayectoria con perturbaciones externas y ruido de medición. Este trabajo demuestra la estabilidad del controlador mediante el uso de funciones de Lyapunov y su efectividad para realizar la tarea propuesta mediante la simulación de un manipulador robótico de 2 grados de libertad.

De igual forma, Bin Ren, Yao Wang, Lilan Liu, X. Tu, Rogelio Lozano (2018) que estudia el problema de seguimiento de trayectorias en manipuladores robóticos de 6 DOF con parámetros desconocidos, este trabajo de investigación demostró que la aplicación de algoritmos de control robusto adaptativo en este tipo de manipuladores mejora significativamente el desempeño del sistema.

Debido a que la presente investigación se centra en el diseño y la comparación de un controlador no lineal adaptativo, se usaron valores para el factor de amortiguamiento, así como de frecuencia natural, los cuales son mencionados en la bibliografía existente. Teniendo como base el concepto principal que la aproximación de polos puede simplificar el análisis de sistemas, como lo dice Ogata (1997), es frecuente que la ganancia de un sistema de orden superior se ajuste para que exista un par de polos dominantes complejos conjugados en lazo cerrado, la presencia de tales polos en un sistema estable reduce el efecto de las no linealidades, tales como la zona muerta, el huelgo (backlash) y la fricción de Coulomb.

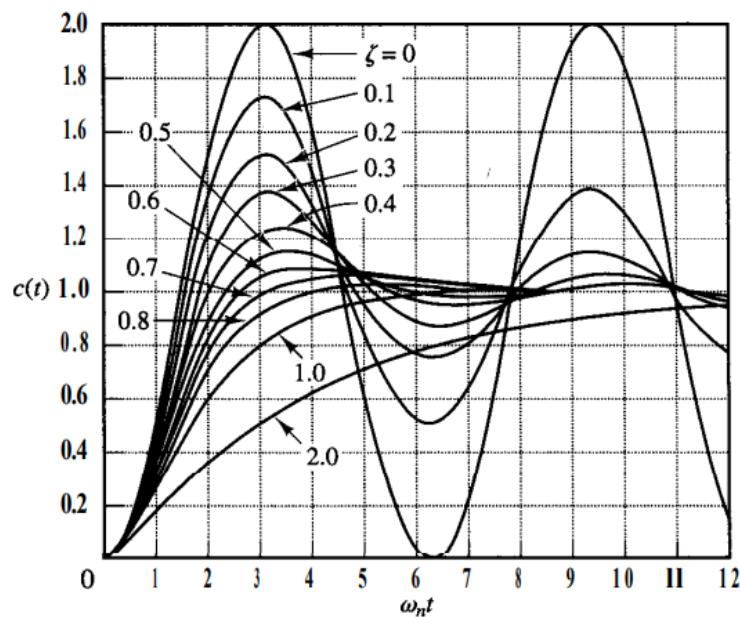


Figura 1. Respuesta de un sistema de segundo orden al escalón unitario con diferentes valores de amortiguamiento.

Fuente: Ogata (1997)

Por ello los valores escogidos atienden a la selección de los polos dominantes para asegurar una estabilidad en la ecuación en lazo cerrado del sistema de control utilizando el método de aproximación de polos, normalmente se podrían escoger otros valores de amortiguación y frecuencia natural, haciendo uso de distintos métodos como un podría ser utilizando la integral del error, los cuales harían que los polos del sistema se vayan más hacia la izquierda volviendo el sistema más estable, o hacia la derecha haciéndolo más inestable, pero esto acarrearía consecuencias como afectar el tiempo de respuesta y los sobrepicos del sistema.

Por esta razón se tomó el valor de $\zeta = 0.5$, porque representa un intermedio dentro de los valores de amortiguación el cuál nos dará una respuesta con un pico reducido y una velocidad de respuesta aceptable ya que se encuentra en el rango de $0.4 < \zeta < 0.8$ lo cual asegura un sistema que se acerca al valor final con mayor rapidez que un sistema críticamente amortiguado o sobreamortiguado.

Observe que dos sistemas de segundo orden que tienen el mismo ζ pero diferente ω_n , rebasarán en la misma medida el límite máximo y mostrarán el mismo patrón oscilatorio. Se dice que tales sistemas tienen la misma estabilidad relativa, Ogata (1997) por la misma razón se escoge un valor intermedio y en un punto de inflexión con un valor $\omega_n = 5$.

La medición de los parámetros de desempeño del controlador se hace tal cual, Daniel (2017) en función de los parámetros que caracterizan una respuesta transitoria del sistema.

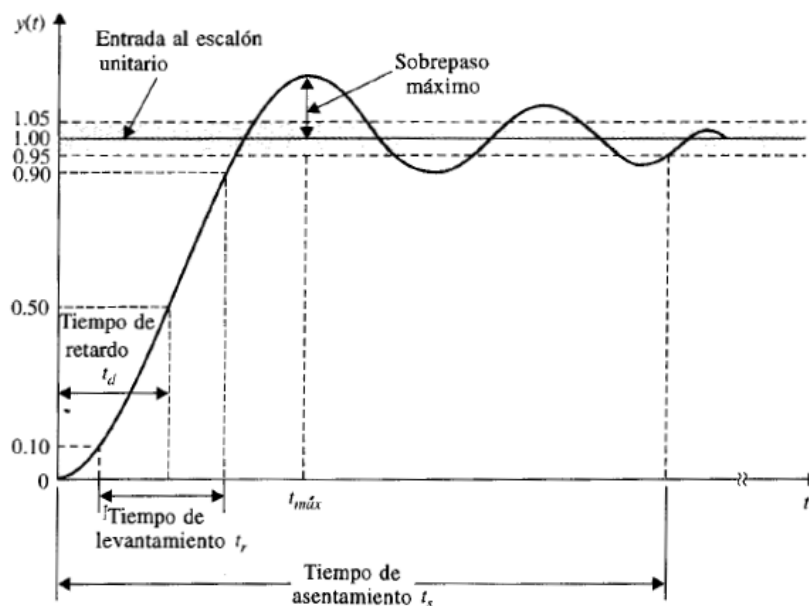


Figura 2. Respuesta al escalón unitario de un sistema de control

Fuente: Kuo (1996)

Sobreimpulso máximo.- Definido en base al valor de salida máximo alcanzado Y_{max} y al valor en estado estable Y_{ss} como $SP_{max} = Y_{max} - Y_{ss}$. Es normalmente representado como un porcentaje del valor en estado estable y para el presente trabajo se consideró un valor aceptable del 20%. Generalmente este valor es determinado en función a los parámetros de calidad del producto final obtenido mediante el proceso controlado.

Tiempo de levantamiento t_r . - Es el tiempo de respuesta del sistema ante una señal, para que la salida se eleve del 10 al 90% del calor de estado estable Y_{ss} .

Tiempo de asentamiento t_s . - Es el tiempo para que la respuesta al escalón se mantenga dentro de un determinado porcentaje alrededor del valor de estado estable.

De la misma manera en distintos trabajos se toma en cuenta otro tipo de índice para la medición del desempeño, como es el caso de O. Arrieta, V. Alfaro (2003) que estudia un tipo de sintonización PID usando como una forma de cuantificar el comportamiento de los lazos de control basado en la señal del error $e(t)$, la cuál es la diferencia entre el valor deseado de la variable controlada y su valor real.

Los más conocidos son los llamados criterios integrales definidos como:

Integral del error cuadrático

$$ISE = \int_0^{\infty} e(t)^2 dt$$

Integral del tiempo por el error cuadrático

$$ITSE = \int_0^{\infty} te(t)^2 dt$$

Integral del error absoluto

$$IAE = \int_0^{\infty} |e(t)| dt$$

Integral del tiempo por el error absoluto

$$ITAE = \int_0^{\infty} t|e(t)| dt$$

Para la presente investigación se tomarán en cuenta los índices de desempeño ISE, IAE e ITAE, por la razón que con estos 3 índices se puede obtener una imagen clara del desempeño del controlador al considerar velocidad, y tiempo en la medición del error.

El índice ITSE será obviado en esta oportunidad ya que su interpretación nos indica los valores de velocidad a lo largo del tiempo, siendo estos ya considerados por el ITAE con la diferencia que del ITSE obtendremos valores aumentados según la cantidad de error lo que nos desviará del porcentaje de error en un estado avanzado del tiempo de simulación

2. OBJETIVOS

2.1 OBJETIVO PRINCIPAL

Desarrollar y analizar el desempeño de un controlador no lineal (adaptativo) para el control de diferentes trayectorias y compararlo mediante la medición del error frente a 3 estrategias de control lineal, en un manipulador robótico de 2DOF.

2.2 OBJETIVOS ESPECIFICOS

- Modelar un manipulador robótico de 2DOF
- Diseñar controladores de tipo lineal para el control de trayectoria.
- Diseñar un controlador no lineal, adaptativo, para el control de trayectoria.
- Evaluar el desempeño del controlador adaptativo mediante simulación en el seguimiento de determinadas trayectorias, haciendo uso de indicadores de desempeño.
- Comparar el desempeño de los distintos controladores.

3. HIPOTESIS

Utilizando un controlador no lineal de tipo adaptativo se logrará un control de trayectoria más preciso que otros tipos de control lineal, considerando la existencia de incerteza paramétrica en el modelo dinámico del robot.

4. VARIABLES

4.1 VARIABLES DEPENDIENTE

Controlador adaptativo: Controlador diseñado para el seguimiento de trayectoria.

4.2 VARIABLE INDEPENDIENTE

Posición del end-effector del manipulador robótico: Posición actual del end-effector durante el seguimiento de trayectoria

5. INDICADORES

Medición del error: Diferencia entre la posición actual y la posición deseada del end-effector.

6. ALCANCES Y LIMITACIONES

Mediante el presente trabajo de investigación los alcances que se pretenden cumplir son los siguientes: modelamiento de un manipulador robótico de dos grados de libertad, diseño de un controlador adaptativo para el control de trayectoria y su posterior evaluación de desempeño en comparación con controladores de tipo lineal: PD, PD aumentado y torque computado, para el seguimiento de 3 trayectorias: una curva suavizada, una circunferencia de diámetro reducido y una trayectoria lineal con cambio de dirección y de esta forma demostrar el desempeño superior del controlador adaptativo ante la presencia de incertezas paramétricas de modelamiento del manipulador.

Debido a la naturaleza de este trabajo, la principal limitación encontrada es la no existencia de un manipulador robótico físico, esto es, se simulará un manipulador de forma numérica utilizando algunos parámetros escogidos sobre el cual se evaluarán los diferentes controladores desarrollados.

Se consideró como único parámetro desconocido o perturbación la masa m_2 del link 2 del brazo robótico, para facilitar el modelamiento y diseño del controlador adaptativo, por ello una de las limitaciones del trabajo es diseñar el controlador en base a un solo parámetro desconocido.

Por otro lado, ya que el principal objetivo no es el análisis de estabilidad de los controladores desarrollados, no se entrará en detalle en este tipo de análisis, sin embargo, se tendrá en cuenta como una recomendación para una implementación futura.



CAPITULO II

MARCO TEÓRICO

CAPITULO II

MARCO TEORICO

1. MODELAMIENTO DEL MANIPULADOR ROBOTICO DE 2 GRADOS DE LIBERTAD

1.1 MANIPULADOR ROBÓTICO

Un manipulador robótico consiste en una secuencia de cuerpos rígidos llamados elementos, conectados mediante articulaciones prismáticas (deslizante, traslacional) si un eslabón se desliza sobre un eje solidario al anterior o de revolución en caso que un eslabón gire en torno a un eje solidario al eslabón anterior, este conjunto de eslabones y articulaciones se denomina cadena cinemática, cada par articulación-elemento constituye un grado de libertad, denominado así a cada una de las coordenadas independientes que son necesarias para describir el estado del sistema mecánico del robot en función de la posición y orientación en el espacio de sus elementos.

Equipos programables y multifuncionales diseñados para mover materiales, partes, herramientas o dispositivos específicos mediante movimientos programados para realizar diferentes tareas. (González.V, 2006)

1.2 CONFIGURACIONES CLÁSICAS DE UN MANIPULADOR ROBÓTICO

Un brazo manipulador puede presentar cinco configuraciones clásicas.

1.2.1 ROBOT CARTESIANO

Utiliza tres dispositivos deslizantes perpendiculares entre sí, para generar movimientos de acuerdo a los tres ejes cartesianos X, Y, Z. González.V (2006). Aplicado generalmente a máquinas con control numérico, fresado o dibujo.

1.2.2 ROBOT CILÍNDRICO

Se basa en una columna vertical que gira sobre la base, también cuenta con dos dispositivos deslizantes que pueden generar movimientos sobre los ejes Z e Y. Aplicado a traslados de materiales, montaje (González.V, 2006).

1.2.3 ROBOT ESFÉRICO O POLAR

Esta configuración utiliza un brazo telescópico que puede bascular en torno a un eje horizontal. Este eje está montado sobre una base giratoria. Las articulaciones proporcionan al robot la posibilidad de desplazarse en una zona esférica (González.V, 2006).

Aplicado en fundición a presión, soldadura por gas y arco eléctrico.

1.2.4 ROBOT DE BRAZO ARTICULADO O SCARA.

Se trata de una columna que gira sobre la base, el brazo contiene una articulación más solo realiza movimientos en un plano. En el extremo del brazo contiene un eje deslizante que se desplaza en un eje Z.

Aplicaciones como pintado y ensamblaje. (González.V, 2006)

1.2.5 ROBOT ANTROPOMÓRFICO

Configuración constituida por dos componentes rectos que simulan el brazo o antebrazo humano, sobre una columna giratoria. Estos antebrazos están conectados mediante articulaciones que se asemejan al hombro y al codo.

Aplicaciones como en exoesqueletos, soldadura de alta precisión, traslado de materiales con alta fidelidad, entre otros. (González.V, 2006)

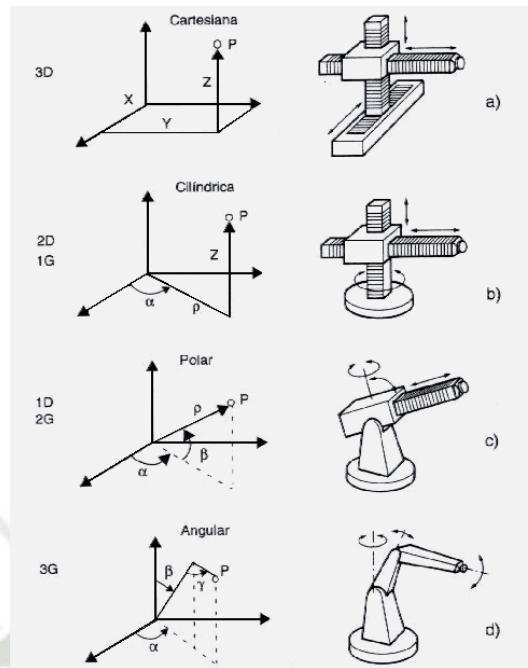


Figura 3: Configuraciones básicas de robots industriales

Fuente: Ferraté et al. (1986)

3. DEFINICIONES

- Zona de trabajo: **Campo de trabajo del robot.**
- Eslabón o junta: **Elemento físico que existe entre las articulaciones.**
- Articulación: **Es aquella que conecta los eslabones, puede ser de revolución, prismáticas o esféricas.**
- Efecto Final: **Elemento que se encuentra en el extremo del robot, mayormente es la herramienta que se va a usar en el trabajo.**
- Sistema de coordenadas: **Espacio en el que se ejecuta el movimiento de cada grado de libertad, expresado en coordenadas articulares y cartesianas.**
- Coordenadas articulares: **Coordenadas para la rotación dadas en un tiempo específico.**
- Coordenadas cartesianas: **Determina la posición del efector final del robot en un plano X, Y.**
- Repetividad: **Precisión en la repetición de movimientos.**

- Entorno de simulación: **Programa que permite la manipulación de variables e interactuar con los parámetros del robot, el tipo de control y el tipo de trayectoria a utilizar. Visualizándose en un espacio 2D mediante gráficas realistas del desempeño.**
- Algoritmo: **Conjunto de pasos definidos para la solución o realización de un trabajo en un número finito de pasos.**

4. CINEMATICA

La cinemática es el campo de estudio referido al movimiento de un cuerpo sin considerar las fuerzas que lo causan, dentro de esta área, se estudia la posición, velocidad, aceleración, o la posición y sus respectivas variables derivativas.

En este contexto, el estudio de la cinemática en manipuladores robóticos es referido a todas las propiedades del movimiento del mecanismo robótico sin tomar en cuenta las fuerzas o torques que producen este movimiento, de esta forma, la cinemática constituye una parte primordial en el diseño de controladores para manipuladores robóticos debido a las aplicaciones que realizan. Por otro lado, el estudio de la cinemática se divide en dos ramas principales: la cinemática directa y la cinemática inversa. (Spong, 2005)

4.1 CINEMATICA DIRECTA

La cinemática directa se utiliza para poder determinar la posición y orientación del end-effector del manipulador robótico con respecto al sistema de coordenadas base del robot dados los valores de posición de todas las juntas y los valores de los parámetros de los links.

La cinemática directa es muy importante en el diseño de algoritmos de control de manipuladores robóticos ya que todos los valores antes mencionados son típicamente medidos a través del uso de sensores instalados en la estructura del robot. (Siciliano, 2009)

Para resolver el problema de la cinemática directa del manipulador se calcula la transformación entre un eje de coordenadas fijo ubicado en el end-effector del manipulador y otro eje de coordenadas que se encuentra fijo en la base. En los manipuladores robóticos este procedimiento es sencillo ya que puede ser obtenido concatenando las transformaciones entre

las diferentes juntas del robot, obteniendo de esta forma una matriz de transformación homogénea de tamaño 4x4 que contiene la información de posición y rotación del end-effector del robot con respecto a su base.

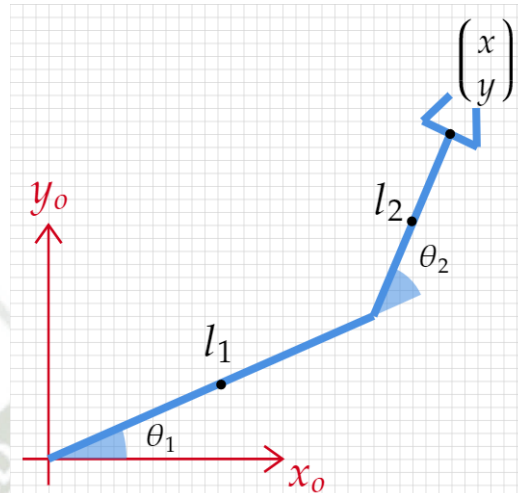


Figura 4: Manipulador de dos grados de libertad.

Fuente: Propia

Sea el manipulador robótico de dos grados de libertad mostrado en la figura anterior, la cinemática directa de este manipulador, para encontrar la posición $\mathbf{P} = [x, y]$ del end-effector, puede ser calculado a través de las siguientes ecuaciones:

4.1.1 CONVENCION DENAVIT-HARTENBERG (DH)

Como se menciona en el apartado anterior, en cada link se encuentra un eje de coordenadas que pueden ser utilizados para describir completamente la geometría de un manipulador robótico. Debido a que estos ejes pueden ser localizados arbitrariamente, es necesario contar con una convención que facilite este cálculo, para ello, es posible utilizar la convención introducida por Denavit y Hartenberg. (Siciliano, 2009)

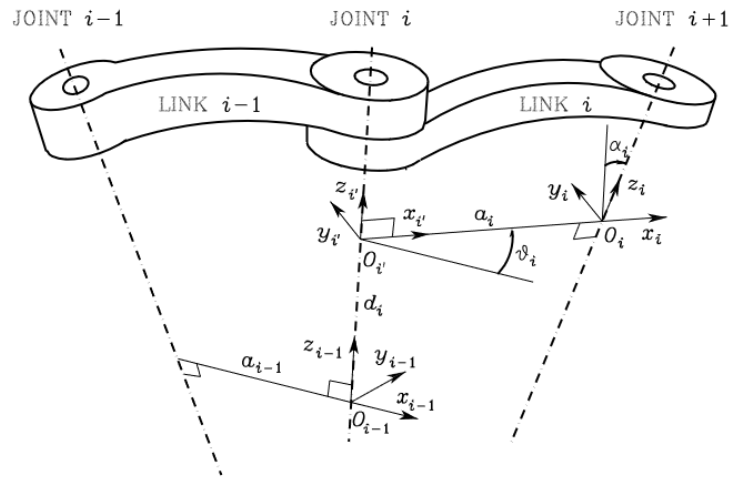


Figura 5: Convención introducida por Denavit y Hartenberg para la localización en un eje de coordenadas.

Fuente: Siciliano (2009)

Tradicionalmente, para localizar un eje de coordenadas con respecto a otro es necesario contar con 6 parámetros, 3 de posición (x, y, z) y 3 de rotación (roll, pitch, yaw). Sin embargo, lo que permite la convención de Denavit-Hartenberg (DH) es el poder utilizar solo 4 parámetros en lugar de 6, dos parámetros que describen a los links: la longitud (a_i) y el giro (α_i) del link, y dos parámetros que describen a las juntas: el desfase (d_i) y el ángulo (θ_i) de la junta.

De esta forma, la relación cinemática entre un par de links adyacentes ($i - 1$ e i) de un manipulador conectados a través de una junta puede ser completamente descrita a través de estos cuatro parámetros ($a_i, \alpha_i, d_i, \theta_i$). Los cuales son definidos como:

a_i : Longitud de la normal común

d_i : Distancia entre los ejes X de x_{i-1} y x_i en el sentido de Z_{i-1}

θ_i : Angulo entre x_{i-1} y x_i en el sentido de Z_{i-1}

α_i : Angulo entre Z_{i-1} y Z_i en el sentido de x_i

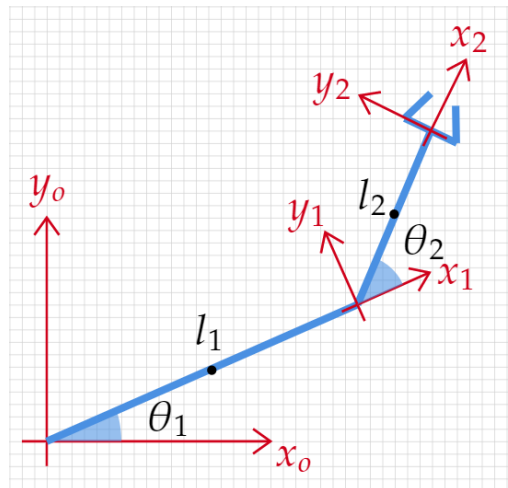


Figura 6: Robot de dos grados de libertad.

Fuente: Propia

Para un robot de dos grados de libertad, como el considerado en este trabajo de investigación, la tabla que resume los parámetros DH sería el siguiente:

Link	a_i	d_i	θ_i	α_i
0	l_1	0	θ_1	0
1	l_2	0	θ_2	0

Tabla 1: Resumen de parámetros DH

Fuente: propia

Mediante esta convención la localización del eje de coordenadas relativas i con respecto al eje $i - 1$ se realiza realizando una rotación a través del ángulo α_i sobre el eje x_{i-1} , una rotación a través del ángulo θ_i alrededor del eje Z_i , y una traslación d_i a través de Z_i .

Matemáticamente esto se puede definir mediante la matriz de transformación siguiente:

$$T_i^{i-1} = \begin{pmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

De esta forma, la matriz de transformación presentada, es una función que depende solamente de los valores de los ángulos y distancias de la junta i (Spong, 2005).

Por otro lado, utilizando la matriz de transformación, fácilmente se puede hallar la transformación de coordenadas que del eje i con respecto al eje 0 , esto se puede realizar mediante una multiplicación en cadena de la forma siguiente:

$$T_i^0 = T_1^0 T_2^1 T_3^2 \dots T_i^{i-1} \quad (2)$$

Considerando el manipulador robótico presentado anteriormente, podemos hallar la transformación de coordenadas que describen la posición y orientación del end-effector utilizando la ecuación recursiva anterior de la siguiente forma:

$$T_{\text{end-effector}}^{\text{base}} = T_1^0 T_2^1 \quad (3)$$

Encontrándose la base en el eje coordinado 0 y el end-effector en el eje coordinado 2, de este modo, la matriz resultante tendrá la siguiente forma:

$$T_{\text{end-effector}}^{\text{base}} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

En la cual, podemos encontrar la información de la rotación y posición del end-effector, siendo la matriz de rotación y el vector posición los siguientes respectivamente:

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (5)$$

$$\mathbf{P} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (6)$$

4.2 CINÉTICA INVERSA

Mientras la cinemática directa establece una relación entre el espacio de juntas y el sistema Cartesiano, la cinemática inversa determina toda una serie de posibilidades o conjuntos posibles de variables que podrán llevar el end-effector del robot hacia una posición y orientación deseada. En otras palabras, mediante la cinemática inversa se encuentran los valores de las posiciones de las juntas dados los valores de posición y orientación del end-effector del manipulador.

Para calcular la cinemática inversa existen dos enfoques de solución, el geométrico y el algebraico, los cuales son utilizados para poder realizar el cálculo de la cinemática inversa de manera analítica. Para los manipuladores robóticos que poseen más de 3 links y que además realicen movimientos en las 3 dimensiones este cálculo se vuelve mucho más complicado. Por este motivo, en este tipo de manipuladores con varios links, se elige el enfoque algebraico para poder encontrar la solución de la cinemática inversa. (Craig, 2009)

$$q_1 = \tan^{-1} \left(\frac{P_y(a_1 + a_1 c_2) - P_x a_2 s_2}{P_y a_2 s_2 + P_x(a_1 + a_2 c_2)} \right) \quad (7)$$

$$q_2 = \cos^{-1} \left(\frac{P_y^2 + P_x^2 - a_1^2 - a_2^2}{2a_1 a_2} \right) \quad (8)$$

$$q_3 = -p_z \quad (9)$$

$$\mathbf{q}_4 = \mathbf{q}_1 - \mathbf{q}_2 - \arctan2(Ay, Ax) \quad (10)$$

La resolución de la cinemática inversa es computacionalmente costosa y generalmente toma mucho tiempo en ser utilizado en control en tiempo real en manipuladores. Matemáticamente puede ser expresado como:

$$\theta_k = f(x, y, z, \alpha, \gamma, \phi) \quad (11)$$

Donde $k=1,2,\dots,i$, θ es el ángulo de la junta y $(x, y, z, \alpha, \gamma, \phi)$ representa la posición y la orientación.

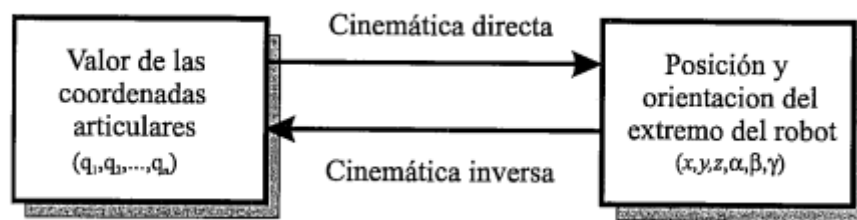


Figura 7: Diagrama de relación entre cinemática directa e inversa

Fuente: Fundamentos de Robótica (1997)

5. DINÁMICA

La dinámica del robot relaciona el movimiento del robot y las fuerzas implicadas en el mismo. El modelo dinámico establece relaciones matemáticas entre las coordenadas articulares (o las coordenadas del extremo del robot), sus derivadas (velocidad y aceleración), las fuerzas y pares aplicados en las articulaciones (o en el extremo) y los parámetros del robot (masas de los eslabones, inercias, etc). (A. Barrientos, 1997)

5.1 MODELAMIENTO DINAMICO

La dinámica se ocupa de la relación entre las fuerzas que actúan sobre un cuerpo y el movimiento que se origina. Por lo tanto, el modelo dinámico del robot tiene por objetivo conocer la relación entre el movimiento del robot y las fuerzas implicadas en el mismo. Dentro

del mismo existen dos problemas relacionados con la dinámica de un manipulador. El primero relacionado a la dinámica directa, en el cual el torque es calculado a partir de datos de desplazamiento angular, velocidad angular y aceleración, mientras que el segundo es referido a la dinámica indirecta donde los datos de velocidad angular, el desplazamiento angular y la aceleración son calculados a partir de un torque dado.

El modelamiento dinámico de un robot puede ser realizado por diferentes métodos tales como el método de Newton-Euler, Lagrange-Euler, etc. El método de Newton-Euler está basado en el análisis de fuerzas y momentos de cada link del robot, para ello requiere de un conjunto de ecuaciones recursivas que van hacia adelante y hacia atrás, de este modo las ecuaciones recursivas que van hacia adelante propagan la información cinemática como la velocidad y la aceleración en el centro de masa de cada link. Mientras que las ecuaciones que van hacia atrás propagan las fuerzas y los momentos ejercidos en el end-effector hacia la base del robot. Por esta razón se podría decir que el método de Newton-Euler utiliza un enfoque de equilibrio de fuerzas utilizado para calcular la dinámica.

Por otro lado, el método de Euler-Lagrange depende de calcular la energía cinética y potencial total del manipulador para poder calcular el Lagrangiano del sistema entero y de esta forma poder calcular el torque de cada junta del robot. A continuación, veremos este método con más detalle. (A. Barrientos, 1997)

5.1.1 METODO DE EULER- LAGRANGE

Como es mencionado anteriormente, el método de Euler-Lagrange es un enfoque basado en cálculos de energía para así poder calcular la dinámica del robot y de esta forma describir el comportamiento de la dinámica del sistema en términos de trabajo y energía almacenado en el sistema en lugar de fuerzas y momento individuales de cada link. De esta forma las ecuaciones dinámicas en forma cerrada pueden derivarse sistemáticamente en cualquier sistema de coordenadas. (Siciliano, 2009)

Mediante este método, el lagrangiano de un manipulador robótico, más generalmente de un sistema dinámico, es definido como:

$$L = T - U \quad (12)$$

Donde T representa la energía cinética y U representa la energía potencial. La energía cinética puede ser calculada mediante la siguiente ecuación

$$T = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{H} \dot{\mathbf{q}} \quad (13)$$

Donde H es la matriz de inercia y \mathbf{q} es el vector de posiciones de las juntas. Por otro lado, la energía potencial es calculada de manera general como:

$$U = \sum_{i=1}^n \mathbf{g}^T \mathbf{P}_{ci} \mathbf{m}_i \quad (14)$$

Siendo \mathbf{g}^T el vector gravitacional, \mathbf{P}_{ci} es el centro de masa del link i y \mathbf{m}_i es la masa del link.

Las ecuaciones dinámicas de movimiento pueden ser desarrolladas usando la ecuación de Lagrange por cada coordenada generalizada de la siguiente manera:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i \quad (15)$$

Siendo τ_i el vector de fuerza generalizada asociada con la coordenada generalizada q_i .

Considerando la ecuación anterior, y tomando las derivadas de las ecuaciones de energía, la ecuación resultante puede ser escrita de forma escalar:

$$\sum_{j=1}^n \mathbf{H}_{ij} \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n \mathbf{C}_{ijk} \dot{q}_j \dot{q}_k + \mathbf{g}_i(\mathbf{q}) = \tau_i \quad (16)$$

El cual describe la dinámica del manipulador robótico en términos de inercia, centrífugos, Coriolis y gravitacionales. Los términos C_{ijk} son conocidos como los terminos de Christoffel y se pueden calcular como:

$$C_{ijk} = \frac{1}{2} \left(\frac{\partial H_{ij}}{\partial q_k} - \frac{1}{2} \frac{\partial H_{jk}}{\partial q_i} \right) \quad (17)$$

Los cuales dependen de las variables de posición de las juntas.

Mediante la elección de la matriz q_i es posible demostrar que una matriz N simétrica se puede calcular como:

$$N(\mathbf{q}, \dot{\mathbf{q}}) = \dot{H}(\mathbf{q}) - 2C(\mathbf{q}, \dot{\mathbf{q}}) \quad (18)$$

Finalmente, todas las ecuaciones del modelo dinámico del robot pueden ser expresadas en su forma compacta como:

$$H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (19)$$

La cual describe el movimiento de un manipulador robótico en el espacio de juntas.

Siendo: $\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}$, en primer lugar, los vectores de aceleración, velocidad y posición respectivamente, por otro lado, también se define el vector de fuerza $\boldsymbol{\tau}$ para cada uno de los grados de libertad del manipulador robótico. Además, es definida una matriz H simétrica positiva-definida de tamaño $n \times n$ (donde n representa el numero de grados de libertad del robot) conocida como la matriz de inercia. El segundo término de la ecuación C representa una matriz de tamaño $n \times n$ que contiene los terminos de coriolis y centrífugos. Finalmente, el tercer término \mathbf{g} es una vector que contiene los términos de gravedad.

5.2 MODELAMIENTO DINÁMICO DE UN MANIPULADOR ROBÓTICO DE DOS GRADOS DE LIBERTAD

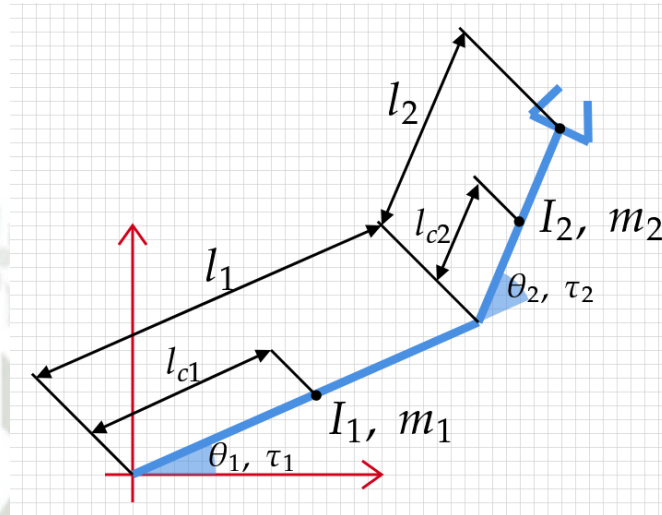


Figura 8: Manipulador robótico con vector de coordenadas generalizadas.

Fuente: propia

Considerando el manipulador robótico de la figura anterior, cuyo vector de coordenadas generalizadas es $\mathbf{q} = [\theta_1, \theta_2]^T$, l_{c1} y l_{c2} son las distancias de los centros de masa con respecto a las juntas del robot, m_1 y m_2 son las masas de los links, l_1 y l_2 son las longitudes de los links, y finalmente I_1 e I_2 son los momentos de inercia relativos a los centros de masa de cada link. De acuerdo con las ecuaciones de movimiento de la dinámica del manipulador descritas en la sección anterior, la matriz de inercia puede escribirse como:

$$\mathbf{H}(\mathbf{q}) = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \quad (20)$$

Donde:

$$\mathbf{H}_{11} = m_1 l_{c1}^2 + I_1 + m_2 [l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(\theta_2)] + I_2 \quad (21)$$

$$\mathbf{H}_{22} = m_2 l_{c2}^2 + I_2 \quad (22)$$

$$\mathbf{H}_{12} = \mathbf{H}_{21} = m_2 l_1 l_{c2} \cos(\theta_2) + m_2 l_{c2}^2 + I_2 \quad (23)$$

Por otro lado, la matriz \mathbf{C} puede ser calculada como:

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} \quad (24)$$

Donde:

$$\mathbf{C}_{11} = m_2 l_1 l_{c2} \sin(\theta_2) \dot{\theta}_2 \quad (25)$$

$$\mathbf{C}_{12} = -m_2 l_1 l_{c2} \sin(\theta_2) (\dot{\theta}_1 + \dot{\theta}_2) \quad (26)$$

$$\mathbf{C}_{21} = m_2 l_1 l_{c2} \sin(\theta_2) \dot{\theta}_1 \quad (27)$$

$$\mathbf{C}_{22} = 0 \quad (28)$$

Finalmente, el vector con los términos gravitacionales se define como:

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix} \quad (29)$$

Donde:

$$\mathbf{g}_1 = m_1 l_1 l_{c1} \mathbf{g} \cos(\theta_1) + m_2 \mathbf{g} [l_{c2} \cos(\theta_1 + \theta_2) + l_1 \cos(\theta_1)] \quad (30)$$

$$\mathbf{g}_2 = m_2 \mathbf{g} l_{c2} \cos(\theta_1 + \theta_2) \quad (31)$$

Asumiendo además la ausencia de fricción y demás fuerzas, las ecuaciones de movimiento resultante para un manipulador robótico de dos grados de libertad (2 DOF) son:

$$\begin{aligned} \tau_1 = & (m_1 l_{c1}^2 + I_1 + m_2 [l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(\theta_2)] + I_2) \ddot{\theta}_1 + \\ & (m_2 l_1 l_{c2} \cos(\theta_2) + m_2 l_{c2}^2 + I_2) \ddot{\theta}_2 - \\ & 2m_2 l_1 l_{c2} \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2 - m_2 l_1 l_{c2} \sin(\theta_2) \dot{\theta}_2^2 + \\ & m_1 l_1 l_{c1} g \cos(\theta_1) + m_2 g [l_{c2} \cos(\theta_1 + \theta_2) + l_1 \cos(\theta_1)] \end{aligned} \quad (32)$$

Y

$$\begin{aligned} \tau_2 = & (m_2 l_1 l_{c2} \cos(\theta_2) + m_2 l_{c2}^2 + I_2) \ddot{\theta}_1 + (m_2 l_{c2}^2 + I_2) \ddot{\theta}_2 \\ & + m_2 l_1 l_{c2} \sin(\theta_2) \dot{\theta}_1^2 + m_2 g l_{c2} \cos(\theta_1 + \theta_2) \end{aligned} \quad (33)$$

Donde: τ_1 y τ_2 son los torques aplicados a las juntas.

5.2.1 PROPIEDADES NOTABLES DEL MODELO DINÁMICO

SIMETRÍA DIAGONAL DE LA MATRIZ N

Esta propiedad nos indica que la matriz N, obtenida mediante la siguiente ecuación:

$$N(q, \dot{q}) = \dot{H}(q) - 2C(q, \dot{q}) \quad (34)$$

Es simétrica diagonalmente de acuerdo con la selección de la matriz C (q, \dot{q}), lo cual también indica que la selección de la matriz C (q, \dot{q}) no es única (Siciliano, 2009).

Esta propiedad se puede comprobar mediante la ecuación siguiente:

$$w^T N(q, \dot{q}) w = 0 \quad (35)$$

Para cualquier vector w de tamaño $n \times 1$.

LINEALIDAD DE LOS PARÁMETROS DINÁMICOS

Esta propiedad nos indica que el modelo dinámico es “lineal” con respecto a los parámetros dinámicos que caracterizan el manipulador robótico, Siciliano (2009). En otras palabras, existe un vector π de tamaño $p \times 1$ con parámetros constantes, y una matriz Y , usualmente llamada como “regresor”, de tamaño $n \times p$ el cual es función de las posiciones, velocidades y aceleraciones de las juntas. De modo que se cumple la siguiente relación:

$$H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = Y(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\pi} \quad (36)$$

PARAMETRIZACIÓN DEL MODELO DINÁMICO

Tomando en consideración la propiedad notable de linealidad de los parámetros dinámicos del manipulador robótico, el modelo dinámico antes presentado también puede ser representado de forma parametrizada de acuerdo con la ecuación siguiente:

$$\boldsymbol{\tau} = Y(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\pi} \quad (37)$$

Donde $\boldsymbol{\pi}$ es el vector de parámetros de tamaño $p \times 1$ y Y es una matriz de regresión en función de las posiciones de las juntas, las velocidades y las aceleraciones de tamaño $n \times p$. De este modo, el manipulador considerado en este trabajo de investigación podría ser parametrizado de la forma siguiente:

$$\boldsymbol{\pi} = [\pi_1 \ \pi_2 \ \pi_3 \ \pi_4 \ \pi_5]^T \quad (38)$$

Con:

$$\pi_1 = m_2 \quad (39)$$

$$\pi_2 = m_2 l_{c2} \quad (40)$$

$$\pi_3 = I_1 + m_1 l_{c1}^2 \quad (41)$$

$$\pi_4 = I_2 + m_2 l_{c2}^2 \quad (42)$$

$$\pi_5 = m_1 l_{c1} \quad (43)$$

Por otro lado, el regresor puede ser considerado de la forma siguiente:

$$Y = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} & y_{15} \\ y_{21} & y_{22} & y_{23} & y_{24} & y_{25} \end{bmatrix} \quad (44)$$

Con:

$$y_{11} = l_1^2 \ddot{\theta}_1 + l_1 \cos(\theta_1) g \quad (45)$$

$$y_{12} = 2l_1 \cos(\theta_2) \ddot{\theta}_1 + l_1 \cos(\theta_2) \ddot{\theta}_2 - l_1 \sin(\theta_2) \dot{\theta}_2^2 + g \cos(\theta_1 + \theta_2) \quad (46)$$

$$y_{13} = \ddot{\theta}_1 \quad (47)$$

$$y_{14} = \ddot{\theta}_1 + \ddot{\theta}_2 \quad (48)$$

$$y_{15} = g \cos(\theta_1) \quad (49)$$

$$y_{21} = 0 \quad (50)$$

$$y_{22} = l_1 \cos(\theta_2) \ddot{\theta}_1 + l_1 \sin(\theta_2) \dot{\theta}_1^2 + g \cos(\theta_1 + \theta_2) \quad (51)$$

$$y_{23} = 0 \quad (52)$$

$$y_{24} = \ddot{\theta}_1 + \ddot{\theta}_2 \quad (53)$$

$$y_{25} = 0 \quad (54)$$

6. SISTEMAS DE CONTROL

6.1 SISTEMA DE CONTROL EN LAZO CERRADO

En diferencia con el control en lazo abierto, existen situaciones en las cuales se necesita una relación entre la entrada y la salida.

Un sistema de control en lazo cerrado es el cuál presenta una relación entre la entrada y la salida, es decir, la acción de control es dependiente de la salida. De esta manera la salida influye en la entrada y para lograr esto es necesario que exista una modificación. Para ello se usa lo que llamamos realimentación. (W, 2013)

La realimentación es una propiedad de los sistemas de control en lazo cerrado mediante la cual una variable del sistema que esté siendo controlada se compara con la entrada o una de las entradas del sistema de manera que se genere una acción de control que sea una función de ambas.

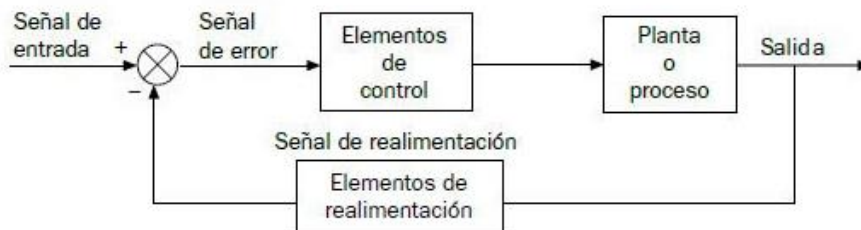


Figura 10: Sistema de control en lazo cerrado

Fuente: Wordpress (2018)

En la Figura 8 se observa la existencia de una realimentación de la salida hacia la entrada. Se realiza una comparación o una resta entre la entrada que es la señal deseada o consigna, y el valor de la salida, dando origen a una señal de error.

La señal que entrega el controlador es llamada señal de control o manipulada y la entregada por la salida, señal controlada.

Esta salida del sistema de regulación se realimenta mediante un elemento de realimentación o captador, es aquel que recibe la información de la salida y se compara con la señal de consigna:

$$\text{Error} = \text{entrada} - \text{salida}$$

Si la señal de error es nula entonces la salida tendrá exactamente el valor deseado; mientras que si la señal de Error no es nula esta señal actuará sobre el elemento de control el cual proporcionará una señal, que a través de un elemento accionador o un actuador, influirá en la planta o proceso para que la salida alcance el valor deseado y de esta manera el Error se anule. (W, 2013)

7. CONTROL DE MOVIMIENTO

Como es posible ver en apartados anteriores, los manipuladores robóticos presentan un gran grado de complejidad e incertezas en su dinámica que resultan en un gran desafío para el control de movimiento.

Para el diseño de controladores que nos permitan alcanzar trayectorias deseadas vamos a considerar el modelo dinámico del manipulador robótico que fue descrito en la sección anterior mediante la formulación de las ecuaciones de Lagrange, esto es:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (55)$$

Donde: \mathbf{H} es la matriz de inercia de tamaño $\mathbf{n} \times \mathbf{n}$, para un manipulador de \mathbf{n} grados de libertad. \mathbf{C} que es la matriz que describe las fuerzas centrífugas y de Coriolis, y finalmente el vector \mathbf{g} de tamaño $\mathbf{n} \times \mathbf{1}$ que describe las fuerzas de gravedad.

Por razones de simplicidad, las fuerzas de fricción y otras perturbaciones no serán consideradas en el modelo dinámico del manipulador robótico.

Por otro lado, el control de movimiento puede ser realizado mediante dos enfoques, mediante el control en espacio de juntas o mediante el control en el espacio operacional, este trabajo de investigación se enfocará en el primero de ellos.

Mediante este enfoque, el objetivo es el de poder diseñar un controlador que permita que las coordenadas de las juntas puedan seguir un movimiento deseado de tal modo que el error sea el mínimo posible:

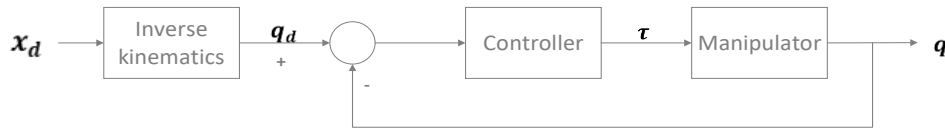


Figura 11: Controlador de reducción de error.

Fuente: Siciliano (2009)

Mediante el gráfico anterior se puede entender de mejor manera esta estrategia de control, como se puede observar, la trayectoria deseada estará expresada en términos de las coordenadas del end-effector, para ello se calcula la cinemática inversa del manipulador robótico, utilizando estos valores el controlador calculará el torque necesario para mover el manipulador robótico hacia la posición deseada.

7.1 CONTROL DE SEGUIMIENTO DE TRAYECTORIAS

A diferencia de los problemas tradicionales de regulación, donde existe una posición fija deseada a la cual la extremidad del manipulador robótico debe posicionarse, existen otro tipo de tareas tales como: corte de precisión, CNC, etc. que requieren de otro tipo de capacidades como el rastreo o seguimiento de trayectoria a través del tiempo.

En este tipo de tareas el objetivo principal es el de seguir o rastrear una trayectoria deseada que varía con respecto al tiempo $q_d(t)$ y de igual manera sus respectivas derivadas: \dot{q}_d, \ddot{q}_d los cuales denotan las velocidades y aceleraciones deseadas respectivamente.

De este modo, el controlador diseñado debe atender estos requerimientos para poder obtener un buen desempeño en términos de velocidad de respuesta, precisión y mínimo error para poder ser utilizado en este tipo de tareas. Diferentes estrategias de control han sido diseñadas para poder atender este tipo de tareas, las cuales pueden ser encontradas en la literatura existente como Ogata (1997), Siciliano (2009), Slotine (1991), Fan & Joo (2009), Adhikary & Mahanta (2015), Fareh, Khadraoui, Baziyad, & Bettayeb (2020), Ma, Zhao, & Di (2020), Ma, Zhao, & Di (2020), algunas de ellas que incluyen técnicas de linealización tal como los controladores

PD o PID, Nicosia & Tomei (1991), algunas otras que consideran la dinámica no lineal del manipulador tal como el controlador PD aumentado, Shang, Cong, Li, & Jiang (2009) o el controlador de torque computado, Wang & Hou (2018), Song, Yi, Zhao, & Li (2005), entre otros, Deotalu & Chiddarwar, (2020).

Por esta razón es que abordamos la comparación del control adaptativo frente a estas estrategias de control (PD, PD aumentado y Torque Computado), las cuales son mencionadas en la bibliografía y son más utilizadas en los procesos reales, en las próximas secciones se presentarán algunos de estos controladores y su aplicación en la tarea de seguimiento de trayectoria.

7.2 CONTROLADOR PD

Debido a que muchos sistemas, especialmente en la industria, pueden ser controlados por compensadores de tipo proporcional (P), derivativo (D) o hasta incluso de tipo integral (I), es que en diferentes tareas se escogen este tipo de controladores para realizar el control de movimiento en manipuladores robóticos. La principal razón de ello es porque el objetivo es el de controlar la posición y velocidad de los diferentes motores en un manipulador robótico, el cual puede ser realizado utilizando un controlador con buena respuesta en lazo cerrado tal como lo es el controlador PD específicamente en puntos de operación donde el manipulador pueda ser considerado como un sistema lineal. (González.V, 2006)

Un diseño interesante de este controlador puede ser encontrado en Siciliano (2009), el cual es un controlador PD con compensación de gravedad. En este controlador el término gravitacional actúa como un corrector que compensa las fuerzas que pueden crear un sobreimpulso y un comportamiento transitorio asimétrico.

La ecuación matemática de este controlador es:

$$\boldsymbol{\tau} = \mathbf{K}_d \dot{\mathbf{e}}_q + \mathbf{K}_p \mathbf{e}_q + \mathbf{g}(\mathbf{q}) \quad (56)$$

Donde K_d y K_p son matrices positivas-definidas de tamaño $n \times n$ siendo n el número de grados de libertad del manipulador robótico.

De este modo, este tipo de controladores pueden ser exitosamente utilizados en problemas de regulación. Sin embargo, a pesar de su gran desempeño para este tipo de tareas, su desempeño se ve reducido en tareas de seguimiento de trayectoria en las cuales son requeridas una gran capacidad de respuesta y gran precisión.

Por este motivo, para mejorar el desempeño del seguimiento de la trayectoria es necesario tomar en consideración los diferentes aspectos no lineales de la dinámica del manipulador tal como lo presenta el controlador PD aumentado.

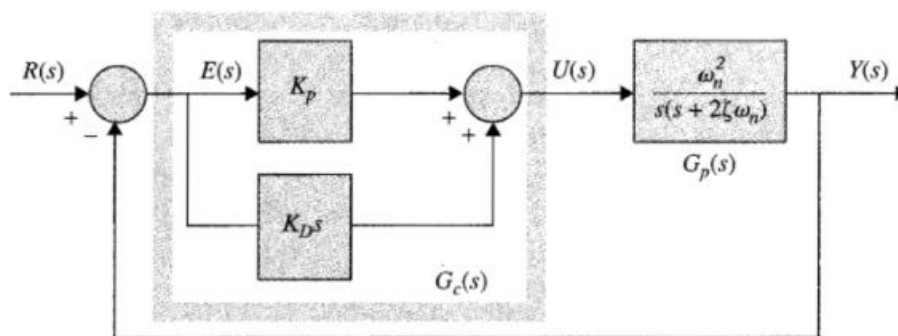


Figura 12: Sistema de control PD

Fuente: dademuchconnection (2018)

7.3 CONTROLADOR PD+ o PD AUMENTADO

A diferencia del controlador PD normal presentado en la sección anterior, el controlador PD+ o PD aumentado toma en consideración los términos no lineales de la dinámica del manipulador robótico, esto es debido a que el controlador PD es un controlador prácticamente lineal, si bien es cierto que puede ser utilizado en tareas como regulación, el desempeño de este controlador se verá reducido en tareas de seguimiento de trayectoria debido a las no linealidades existentes. (Perez Hidalgo, 2018)

Esta estrategia de control responde a la formulación matemática siguiente:

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}) + \mathbf{K}_d\dot{\mathbf{e}}_q + \mathbf{K}_p\mathbf{e}_q \quad (57)$$

En la ecuación anterior se puede observar claramente que además de los términos proporcional y derivativo, se toma en consideración la dinámica no lineal del manipulador robótico, los cuales son expresados en las matrices $\mathbf{H}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ y $\mathbf{g}(\mathbf{q})$ respectivamente. De este modo, este controlador intenta compensar las no linealidades del sistema para realizar un seguimiento de trayectoria más preciso y con menos error que la anterior estrategia de control.

7.4 CONTROLADOR POR TORQUE-COMPUTADO

Como fue presentado en la sección anterior, es necesario tener en consideración la no linealidad de la dinámica del manipulador para poder diseñar una estrategia de control que pueda atender a las especificaciones de la tarea a ser realizada. Similar al PD aumentado, el control por torque computado también toma en consideración las no linealidades del sistema, sin embargo, permite desacoplar la parte no lineal de una manera más sencilla, la cual se realiza mediante una señal de control externa a la cual es añadida una compensación interna con la dinámica no lineal del manipulador. De este modo, la ecuación en lazo cerrado resulta en una ecuación que depende únicamente de elementos lineales, por lo tanto, se puede decir que esta estrategia de control desacopla todas las no linealidades del sistema convirtiéndolo en un sistema lineal fácilmente controlable.

Matemáticamente podemos describirla como:

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\mathbf{v} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (58)$$

Siendo \mathbf{v} la ley de control externa. Para nuestro caso podemos considerar una ley de control PD similar a la revisada en secciones anteriores:

$$\mathbf{v} = \ddot{\mathbf{q}}_d + \mathbf{K}_d\dot{\mathbf{e}}_q + \mathbf{K}_p\mathbf{e}_q \quad (59)$$

Reemplazando la ecuación anterior en la ley de control presentada obtendremos la siguiente ecuación:

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})(\ddot{\mathbf{q}}_d + \mathbf{K}_d\dot{\mathbf{e}}_q + \mathbf{K}_p\mathbf{e}_q) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (60)$$

7.5 CONTROL ADAPTATIVO

Existen muchas definiciones para el control adaptativo o adaptable, el término adaptativo significa cambiar el comportamiento conforme a las circunstancias, por ello un controlador adaptativo es aquel que puede modificar su comportamiento en respuesta a cambios en la dinámica del sistema y a las perturbaciones.

Además de tratarse de un tipo de control no lineal en el que el estado del proceso puede ser separado en dos escalas de tiempo que evolucionan a diferente velocidad. La escala lenta corresponde a los cambios de los parámetros y por consiguiente a la velocidad con la cual los parámetros del regulador son modificados, y la escala rápida corresponde a la dinámica del bucle ordinario de realimentación. (Rodríguez. F, 1996)

Como se pudo observar en secciones anteriores, mediante el desacoplamiento de las no linealidades de la dinámica del manipulador se pueden diseñar estrategias de control con buen desempeño. Sin embargo, todos aquellos controladores requieren que se tenga completo conocimiento de la dinámica del controlador. Esto quiere decir que el error en el modelamiento del manipulador tiene que ser prácticamente nulo. Esto en un escenario real es prácticamente imposible, ya que se pueden encontrar diferentes incertezas que pueden ocasionar que el modelo calculado varíe con respecto al modelo real.

Para este tipo de situaciones es necesario diseñar una estrategia de control que tome en consideración estas incertezas, siendo una de las estrategias de control más utilizadas para este tipo de situaciones el del control adaptativo.

Este controlador, a diferencia de los presentados anteriormente, toma en consideración las incertezas realizando un ajuste paramétrico en tiempo real utilizando las señales en lazo cerrado

del sistema, Slotine (1991) a pesar de las incertezas existentes. Existen diferentes tipos de controladores adaptativos, de los cuales uno de los más populares es el controlador adaptativo relacionado a la inercia ya que permite diseñar de manera muy sencilla la estrategia de control y la compensación paramétrica. (Siciliano, 2009)

Considerando la siguiente ley de control siguiente:

$$\tau = \hat{H}(q)\ddot{q}_r + \hat{C}(q, \dot{q})\dot{q}_r + \hat{g}(q) + K_D\sigma \quad (61)$$

Con:

$$\ddot{q}_r = \ddot{q}_d + \Lambda\dot{e}_q \quad (62)$$

$$\dot{q}_r = \dot{q}_d + \Lambda e_q \quad (63)$$

$$\sigma = \dot{q}_r - \dot{q} \quad (64)$$

Donde: $\hat{H}(q)$, $\hat{C}(q, \dot{q})$ y $\hat{g}(q)$ son matrices estimadas, \ddot{q}_r y \dot{q}_r son señales auxiliares y Λ es una matriz positiva definida. De este modo se puede decir que el termino $K_D\sigma$ es equivalente al termino PD. Para ello es necesario considerar que la ley de adaptación de este controlador responde a la propiedad de linealidad de parámetros del modelo dinámico del manipulador robótico, Siciliano (2009). Tomando en consideración esta propiedad es posible expresar el modelo dinámico del robot en su forma paramétrica siguiente:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = Y(q, \dot{q}, \ddot{q})\pi = \tau \quad (65)$$

Por lo tanto, debido a las incertezas paramétricas, podemos reescribir la ley de control anterior de la forma siguiente:

$$\hat{H}(q)\ddot{q}_r + \hat{C}(q, \dot{q})\dot{q}_r + \hat{g}(q) = Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\hat{a} \quad (66)$$

Donde $\hat{\mathbf{a}}$ es un vector que representa los parámetros estimados.

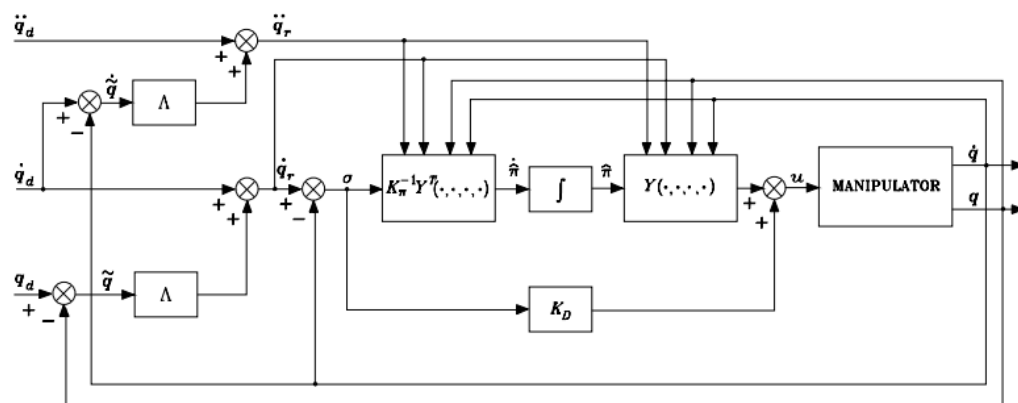


Figura 13: Diagrama de bloques del control adaptativo

Fuente: Siciliano (2009)

8. ¿PORQUÉ CONTROL ADAPTATIVO?

Debido a que en los procesos industriales reales no se cuenta con un modelamiento exacto de la planta, esto deriva a errores en el desempeño de distintos controladores, dado que un controlador adaptativo es un sistema no lineal en el que se necesita ajustar parámetros se tiene que realizar una evaluación para los casos que requieran o no un tipo de controlador adaptativo o no lineal.

El control adaptativo está pensado para interactuar cuando no se conocen el tipo de perturbaciones que afectarán al sistema o cuando estas no puedan medirse a tiempo. Un controlador convencional está pensado para controlar sistemas la mayor parte del tiempo lineales, cuyos parámetros permanecen constantes. Esto es una aproximación buena para la mayor parte de los casos cuando se pretende regular un sistema que presente un punto fijo de operación.

Cuando existen perturbaciones, si estas son pequeñas, dicha aproximación continúa siendo suficiente para obtener un buen control, sin embargo, la aproximación en torno a un punto de

funcionamiento no suele seguir siendo buena si el punto de funcionamiento cambia (Rodríguez. F, 1996).

9. EL PROBLEMA DEL CONTROL ADAPTATIVO

Se pone en manifiesto la necesidad de usar el control adaptativo para procesos industriales que sean bastante complejos y la variación de los parámetros no se pueda determinar de primer momento, por esta razón puede ser ventajoso emplear un esfuerzo en desarrollar controladores más inteligentes.

Si bien un controlador más complejo puede utilizarse para distintos procesos es muy importante recordar que la utilización de un controlador adaptativo no sustituye el buen conocimiento del proceso, que es necesario para elegir las especificaciones, la estructura del controlador y el método de diseño. (Rodríguez. F, 1996)

10. ÍNDICES DE DESEMPEÑO

Se refiere a índice de desempeño como una medida cuantitativa del desempeño en un sistema y este se escoge de manera que se adapte a las especificaciones o requerimientos importantes para el sistema.

Un sistema de control se considera óptimo si los parámetros se escogen de tal forma que minimicen o maximicen estos índices de desempeño (dependiendo de la situación), mayormente se consideran aquellos que minimicen los índices de desempeño.

En otras palabras, este índice es un número que nos indicará que tan efectivo o bueno es el desempeño del sistema de control. (S. Castaño, 2021)

10.1 ISE

Este indicador discrimina entre sistemas que se encuentren con una excesiva amortiguación o aquellos que sean excesivamente subamortiguados. Al presentar el cuadrado del error esto significa que ofrece una penalización considerable cuando el error es mayor y suave cuando el error es pequeño. (S. Castaño, 2021)

Esto significa que este indicador se centra en la velocidad con la cual el error es capaz de disminuir, mayormente presenta niveles adecuados en sistemas con una respuesta rápida, pero con oscilaciones, lo que puede llevar a la poca estabilidad.

Por esta razón un sistema se puede considerar óptimo si presenta un mínimo de este índice, ya que a nivel práctico minimizar el índice ISE implica minimizar el consumo de energía para algunos tipos de sistemas como por ejemplo sistemas aeroespaciales. Asimismo, presenta facilidad de implementación tanto digital como analíticamente.

Aproximación trapezoidal para ISE:

$$\sum_{k=0}^M \frac{e^2(k) + e^2(k-1)}{2} T_s$$

10.2 IAE

Es un índice que es difícil de ser evaluado con medio analíticos, es uno de los más sencillos en ser implementados y supone en la obtención clara de la integral del error, este número supondrá una medida que se podrá comparar respecto a los demás sistemas o controladores para realizar una comparación orientada al desempeño. (S. Castaño, 2021)

Aproximación trapezoidal para IAE:

$$\sum_{k=0}^M \left| \frac{e(k) + e(k-1)}{2} T_s \right|$$

10.3 ITAE

Al igual que el otro criterio que toma en cuenta al tiempo, un gran error inicial es poco penalizado, pero a medida que el tiempo aumenta la penalización es mayor, usado principalmente en sistemas con un pequeño sobreimpulso en la respuesta transitoria y las oscilaciones del sistema son adecuadamente amortiguadas.

El índice de rendimiento ITAE proporciona la mejor selectividad de los índices de rendimiento; es decir, el valor mínimo de la integral es fácilmente discernible a medida que se varían los parámetros del sistema. (S. Castaño, 2021)

Aproximación trapezoidal para el ITAE

$$\sum_{k=0}^M k \left| \frac{e(k) + e(k-1)}{2} T_s \right|$$



CAPITULO III
DESARROLLO DE LA INGENIERIA

CAPITULO III DESARROLLO DE LA INGENIERIA

1. DISEÑO DE CONTROLADORES

Sea un manipulador robótico de dos grados de libertad como se muestra en la figura siguiente:

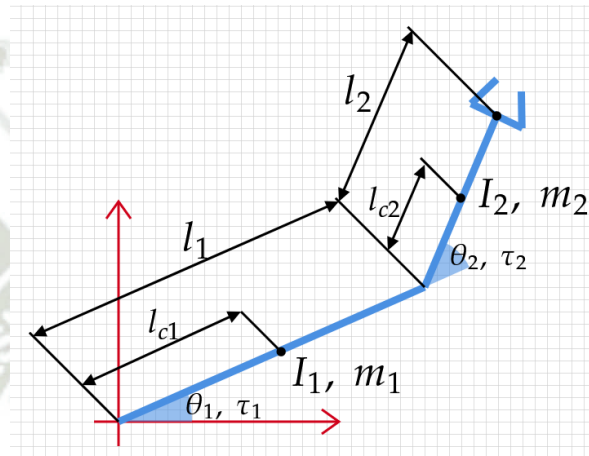


Figura 14: Manipulador robótico de dos grados de libertad.

Fuente: Propia

Considerando el modelo dinámico explicado en secciones anteriores, a partir de las ecuaciones de Euler-Lagrange es posible demostrar que la ecuación dinámica del robot está dada por:

$$\begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_1 \\ \boldsymbol{\tau}_2 \end{bmatrix} \quad (67)$$

Donde:

$$\mathbf{H}_{11} = m_1 l_{c1}^2 + I_1 + m_2 [l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(\theta_2)] + I_2 \quad (68)$$

$$\mathbf{H}_{22} = m_2 l_{c2}^2 + I_2 \quad (69)$$

$$\mathbf{H}_{12} = \mathbf{H}_{21} = m_2 l_1 l_{c2} \cos(\theta_2) + m_2 l_{c2}^2 + I_2 \quad (70)$$

$$\mathbf{C}_{11} = m_2 l_1 l_{c2} \sin(\theta_2) \dot{\theta}_2 \quad (71)$$

$$C_{12} = -m_2 l_1 l_{c2} \sin(\theta_2) (\dot{\theta}_1 + \dot{\theta}_2) \quad (72)$$

$$C_{21} = m_2 l_1 l_{c2} \sin(\theta_2) \dot{\theta}_1 \quad (73)$$

$$C_{22} = 0 \quad (74)$$

$$g_1 = m_1 l_1 l_{c1} g \cos(\theta_1) + m_2 g [l_{c2} \cos(\theta_1 + \theta_2) + l_1 \cos(\theta_1)] \quad (75)$$

$$g_2 = m_2 g l_{c2} \cos(\theta_1 + \theta_2) \quad (76)$$

O en una forma más compacta:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (77)$$

Donde H denota la matriz de inercia del manipulador, C denota la matriz de fuerzas centrífugas y de Coriolis, g representa el vector de fuerzas gravitacionales y τ es la señal de control de torque en las juntas.

Ya que nuestro objetivo de control es el de alcanzar una trayectoria deseada $q_d(t)$ a partir de la configuración actual del robot $q(t)$, es matemáticamente definido como:

$$q \rightarrow q_d(t) \quad (78)$$

$$e_q = q_d(t) - q \rightarrow 0 \quad (79)$$

Donde e_q es el error de configuración en el espacio de las juntas, consideraremos las siguientes trayectorias deseadas para el diseño de los controladores:

Trayectoria curva suavizada:

$$q_d(t) = [-R \cos(\omega_i t) - R, -R \cos(\omega_i t) + 5R] \quad (80)$$

Donde $R=0.5$ m es el radio de la trayectoria.

Trayectoria circular con radio de 0.5:

$$P(t) = [0.5 + 0.5 * \cos(t), 0.5 + 0.5 * \sin(t)] \quad (81)$$

Trayectoria lineal con cambio de dirección:

$$P1(t) = [t, 0.7 * t + 1] \quad (82)$$

$$P2(t) = [t, 0.4 * t + 4.1] \quad (83)$$

De acuerdo con todo lo presentado, se proyectarán leyes de control dinámicas de acuerdo a las estrategias de control siguientes:

1.1 PROPORCIONAL+ DERIVATIVO PD

$$\tau = K_d \dot{e}_q + K_p e_q \quad (84)$$

Donde K_d y K_p son matrices simétricas positiva-definidas y $e_q = q_d(t) - q$ denota el error en el seguimiento en el instante t .

En primer lugar, es necesario definir el valor de las ganancias K_d y K_p , de acuerdo a Ogata (1997), Slotine (1991), es posible escoger el valor de estas ganancias como: $K_p = \text{diag}\{2\zeta_1 \omega_{n1}^2, \dots, 2\zeta_n \omega_{nn}^2\}$ y $K_d = \{\omega_{n1}^2, \dots, \omega_{nn}^2\}$.

Sin embargo, es necesario considerar que el controlador de tipo PD es un controlador lineal. De modo que primero es necesario desacoplar la parte lineal o constante de la ecuación dinámica del robot a fin de dimensionar las ganancias del controlador. Para realizar eso es posible reescribir la matriz de inercia de la siguiente forma:

$$\mathbf{H}(\mathbf{q}) = \mathbf{H}_o + \bar{\mathbf{H}}(\mathbf{q}) \quad (85)$$

Donde \mathbf{H}_o es la parte constante correspondiente a los valores de los links y $\bar{\mathbf{H}}(\mathbf{q})$ es la parte variante en el tiempo correspondiente a los valores de las juntas y del motor respectivamente.

De esta forma, reemplazando la ecuación de la matriz de inercia y la ley de control en la ecuación de dinámica del robot obtenemos:

$$[\mathbf{H}_o + \bar{\mathbf{H}}(\mathbf{q})]\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{K}_d\dot{\mathbf{e}}_q + \mathbf{K}_p\mathbf{e}_q \quad (86)$$

Resolviendo la ecuación anterior:

$$\begin{aligned} \mathbf{H}_o\ddot{\mathbf{q}} + \bar{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) &= \mathbf{K}_d\dot{\mathbf{e}}_q + \mathbf{K}_p\mathbf{e}_q \\ \mathbf{H}_o\ddot{\mathbf{q}} + \bar{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) &= \mathbf{K}_d\dot{\mathbf{e}}_q + \mathbf{K}_p\mathbf{e}_q + \mathbf{H}_o\ddot{\mathbf{q}}_d - \mathbf{H}_o\ddot{\mathbf{q}}_d \\ \mathbf{H}_o\ddot{\mathbf{q}}_d + \bar{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) &= \mathbf{K}_d\dot{\mathbf{e}}_q + \mathbf{K}_p\mathbf{e}_q + \mathbf{H}_o\ddot{\mathbf{q}}_d - \mathbf{H}_o\ddot{\mathbf{q}} \\ \mathbf{H}_o\ddot{\mathbf{q}}_d + \bar{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) &= \mathbf{H}_o\ddot{\mathbf{e}}_q + \mathbf{K}_d\dot{\mathbf{e}}_q + \mathbf{K}_p\mathbf{e}_q \\ \mathbf{H}_o[\ddot{\mathbf{e}}_q + \mathbf{H}_o^{-1}\mathbf{K}_d\dot{\mathbf{e}}_q + \mathbf{H}_o^{-1}\mathbf{K}_p\mathbf{e}_q] &= \mathbf{d}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \ddot{\mathbf{q}}_d) \end{aligned} \quad (87)$$

Donde $\mathbf{d}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \ddot{\mathbf{q}}_d)$ puede ser considerado como una perturbación. Por otro lado, el término derecho de la igualdad podría ser igualado a:

$$\ddot{\mathbf{e}}_q + \mathbf{H}_o^{-1}\mathbf{K}_d\dot{\mathbf{e}}_q + \mathbf{H}_o^{-1}\mathbf{K}_p\mathbf{e}_q = \ddot{\mathbf{e}}_q + 2\zeta_1\omega_n^2\dot{\mathbf{e}}_q + \omega_n^2\mathbf{e}_q \quad (88)$$

A partir de esta ecuación podemos despejar los siguientes términos:

$$\mathbf{H}_o^{-1}\mathbf{K}_d = 2\zeta_1\omega_n^2 \quad (89)$$

$$\mathbf{H}_o^{-1}\mathbf{K}_p = \omega_n^2 \quad (90)$$

Resolviendo las ecuaciones anteriores obtenemos:

$$\mathbf{K}_d = 2\zeta_1\omega_n^2\mathbf{H}_o \quad (91)$$

$$\mathbf{K}_p = \omega_n^2\mathbf{H}_o \quad (92)$$

De este modo obtenemos dimensionamos las ganancias \mathbf{K}_d y \mathbf{K}_p considerando la parte lineal del controlador. Reemplazando los valores de las ganancias en la ley de control obtenemos:

$$\boldsymbol{\tau} = 2\zeta_1\omega_n^2\mathbf{H}_o\dot{\mathbf{e}}_q + \omega_n^2\mathbf{H}_o\mathbf{e}_q \quad (93)$$

Igualando la ecuación anterior con la ecuación de dinámica del controlador se obtiene:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = 2\zeta_1\omega_n^2\mathbf{H}_o\dot{\mathbf{e}}_q + \omega_n^2\mathbf{H}_o\mathbf{e}_q \quad (94)$$

Resolviendo la ecuación anterior es posible obtener la ecuación en lazo cerrado para el sistema utilizando el controlador PD:

$$\ddot{\mathbf{q}} = \mathbf{H}^{-1}(\mathbf{q})[2\zeta_1\omega_n^2\mathbf{H}_o\dot{\mathbf{e}}_q + \omega_n^2\mathbf{H}_o\mathbf{e}_q - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})] \quad (95)$$

Con $\zeta = 0.5$, $\omega_n = 5 \text{ rad s}^{-1}$ para $n = 1, 2$ y:

$$\mathbf{H}_0 = \begin{bmatrix} \mathbf{m}_1 l_{c1}^2 + I_1 + \mathbf{m}_2 [l_1^2 + l_{c2}^2] + I_2 & \mathbf{m}_2 l_{c2}^2 + I_2 \\ \mathbf{m}_2 l_{c2}^2 + I_2 & \mathbf{m}_2 l_{c2}^2 + I_2 \end{bmatrix} \quad (96)$$

Y

$$\bar{\mathbf{H}}(\mathbf{q}) = \begin{bmatrix} 2l_1 l_{c2} \cos(\theta_2) \mathbf{m}_2 & \mathbf{m}_2 l_1 l_{c2} \cos(\theta_2) \\ \mathbf{m}_2 l_1 l_{c2} \cos(\theta_2) & 0 \end{bmatrix} \quad (97)$$

1.2 PD AUMENTADO

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}) + \mathbf{K}_d \dot{\mathbf{e}}_q + \mathbf{K}_p \mathbf{e}_q \quad (98)$$

Donde \mathbf{K}_d y \mathbf{K}_p son matrices simétricas positiva-definidas y $\mathbf{e}_q = \mathbf{q}_d(\mathbf{t}) - \mathbf{q}$ denota el error en el seguimiento en el instante \mathbf{t} .

Sustituyendo la ley de control en la ecuación de dinámica del controlador obtenemos:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}) + \mathbf{K}_d \dot{\mathbf{e}}_q + \mathbf{K}_p \mathbf{e}_q \quad (99)$$

Resolviendo la ecuación anterior:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \mathbf{K}_d \dot{\mathbf{e}}_q + \mathbf{K}_p \mathbf{e}_q$$

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{K}_d \dot{\mathbf{e}}_q + \mathbf{K}_p \mathbf{e}_q$$

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_d \dot{\mathbf{e}}_q + \mathbf{K}_p \mathbf{e}_q$$

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_d(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q})$$

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_d + (\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{K}_d)(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q})$$

$$\ddot{\mathbf{q}} = \mathbf{H}^{-1}(\mathbf{q})[\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_d + (\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{K}_d)(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q})] \quad (100)$$

De esta forma se obtiene la ecuación en lazo cerrado de la dinámica del manipulador robótico utilizando el controlador PD aumentado escogiendo $K_p = \text{diag}\{2\zeta_1\omega_{n1}^2, \dots, 2\zeta_1\omega_{nn}^2\}$ y $K_d = \{\omega_{n1}^2, \dots, \omega_{nn}^2\}$ con $\zeta = 0.5$, $\omega_n = 5 \text{ rad s}^{-1}$ para $n = 1, 2$.

1.3 TORQUE COMPUTADO

$$\tau = H(q)[\ddot{q}_d + K_d\dot{e}_q + K_p e_q] + C(q, \dot{q})\dot{q}_d + g(q) \quad (101)$$

Donde K_d y K_p son matrices simétricas positiva-definidas y $e_q = q_d(t) - q$ denota el error en el seguimiento en el instante t .

Sustituyendo la ley de control en la ecuación de dinámica del controlador obtenemos:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = H(q)[\ddot{q}_d + K_d\dot{e}_q + K_p e_q] + C(q, \dot{q})\dot{q}_d + g(q) \quad (102)$$

Resolviendo la ecuación anterior:

$$H(q)\ddot{q} = H(q)[\ddot{q}_d + K_d\dot{e}_q + K_p e_q]$$

$$\ddot{q} = \ddot{q}_d + K_d\dot{e}_q + K_p e_q \quad (103)$$

Escogiendo $K_p = \text{diag}\{2\zeta_1\omega_{n1}^2, \dots, 2\zeta_1\omega_{nn}^2\}$ y $K_d = \{\omega_{n1}^2, \dots, \omega_{nn}^2\}$ con $\zeta = 0.5$, $\omega_n = 5 \text{ rad s}^{-1}$ para $n = 1, 2$, obtenemos la ecuación en lazo cerrado para el sistema utilizando el controlador de torque computado.

$$\ddot{q} = \ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q) \quad (104)$$

1.4 CONTROL ADAPTATIVO

Para casos en los que se sabe de la existencia de una posible incertidumbre paramétrica en el modelo dinámico del robot, por otro lado, sabiendo que el modelo dinámico puede ser parametrizado linealmente, es posible diseñar una ley de control adaptativa para compensar la incertidumbre paramétrica. En este caso, asumiendo que existe una incertidumbre del 50% en el link 2 en su valor nominal.

Para este caso diseñaremos una ley de control adaptativa relacionada a la inercia, de acuerdo con Siciliano (2009), podemos considerar la siguiente ley de control:

$$\boldsymbol{\tau} = \hat{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}}_r + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \hat{\mathbf{g}}(\mathbf{q}) + \mathbf{K}_D\boldsymbol{\sigma} \quad (105)$$

Con:

$$\ddot{\mathbf{q}}_r = \ddot{\mathbf{q}}_d + \boldsymbol{\Lambda}\dot{\mathbf{e}}_q \quad (106)$$

$$\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d + \boldsymbol{\Lambda}\mathbf{e}_q \quad (107)$$

$$\boldsymbol{\sigma} = \dot{\mathbf{q}}_r - \dot{\mathbf{q}} \quad (108)$$

$$\mathbf{e}_q = \mathbf{q}_d(t) - \mathbf{q} \quad (109)$$

Donde $\hat{\mathbf{H}}(\mathbf{q})$, $\hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})$ y $\hat{\mathbf{g}}(\mathbf{q})$ son matrices estimadas las cuales dependen de los parámetros desconocidos y $\boldsymbol{\Lambda}$ es una matriz positiva definida. Por otro lado, podemos escribir:

$$\hat{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}}_r + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \hat{\mathbf{g}}(\mathbf{q}) = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\hat{\mathbf{a}} \quad (110)$$

Donde \mathbf{Y} es una matriz de funciones conocidas y $\hat{\mathbf{a}}$ es un vector de parámetros desconocidos.

En este caso, el parámetro desconocido es la masa m_2 , por lo tanto:

$$\hat{\mathbf{a}} = [\mathbf{m}_2] \quad (111)$$

De modo que, de acuerdo con Siciliano. (2009) podemos realizar la parametrización y representar la ecuación (77) de la siguiente forma:

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r) \hat{\mathbf{a}} = \mathbf{H}_d(\mathbf{q}) \ddot{\mathbf{q}}_r + \mathbf{H}_o(\mathbf{q}) \ddot{\mathbf{q}}_r + \mathbf{C}_d(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}_r + \mathbf{C}_o(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}_r + \mathbf{g}_d(\mathbf{q}) + \mathbf{g}_o(\mathbf{q})$$

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r) \hat{\mathbf{a}} = \bar{\mathbf{Y}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r) \hat{\mathbf{a}} + \mathbf{H}_o(\mathbf{q}) \ddot{\mathbf{q}}_r + \mathbf{C}_o(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}_r + \mathbf{g}_o(\mathbf{q}) \quad (112)$$

Donde \mathbf{H}_d , \mathbf{C}_d y \mathbf{g}_d son matrices del modelo dinámico cuyos elementos dependen directamente del parámetro de interés el cual es parametrizado como $\bar{\mathbf{Y}}$ y \mathbf{H}_o , \mathbf{C}_o y \mathbf{g}_o son matrices del modelo dinámico cuyos elementos no dependen directamente de $\hat{\mathbf{m}}_2$ la cual es la masa estimada del link. De modo que en este caso vamos a obtener:

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r) \hat{\mathbf{a}} = \begin{bmatrix} \bar{\mathbf{y}}_1 \\ \bar{\mathbf{y}}_2 \end{bmatrix} [\mathbf{m}_2] + \mathbf{H}_o(\mathbf{q}) \ddot{\mathbf{q}}_r + \mathbf{C}_o(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}_r + \mathbf{g}_o(\mathbf{q}) \quad (113)$$

Donde:

$$\begin{aligned} \bar{\mathbf{y}}_1 = & (\mathbf{l}_1^2 + \mathbf{l}_{c2}^2 + 2\mathbf{l}_1\mathbf{l}_{c2} \cos \theta_1) \ddot{\mathbf{q}}_{r1} + (\mathbf{l}_1\mathbf{l}_{c2} \cos \theta_2 + \mathbf{l}_{c2}^2) \ddot{\mathbf{q}}_{r2} + \\ & (\mathbf{l}_1\mathbf{l}_{c2} \sin \theta_2 \dot{\mathbf{q}}_2) \dot{\mathbf{q}}_{r1} - (\mathbf{l}_1\mathbf{l}_{c2} \sin \theta_2 (\dot{\mathbf{q}}_1 + \dot{\mathbf{q}}_2)) \dot{\mathbf{q}}_{r2} + \\ & \mathbf{g}(\mathbf{l}_{c2} \cos(\theta_1 + \theta_2) + \mathbf{l}_1 \cos \theta_1) \end{aligned} \quad (114)$$

Y:

$$\begin{aligned} \bar{\mathbf{y}}_2 = & (\mathbf{l}_1\mathbf{l}_{c2} \cos \theta_2 + \mathbf{l}_{c2}^2) \ddot{\mathbf{q}}_{r1} + \mathbf{l}_{c2}^2 \ddot{\mathbf{q}}_{r2} + (\mathbf{l}_1\mathbf{l}_{c2} \sin \theta_2 \dot{\mathbf{q}}_1) \dot{\mathbf{q}}_{r1} \\ & + \mathbf{g}\mathbf{l}_{c2} \cos(\theta_1 + \theta_2) \end{aligned} \quad (115)$$

Adicionalmente:

$$\mathbf{H}_o(\mathbf{q}) = \begin{bmatrix} \mathbf{m}_1 l_{c1}^2 + I_1 + I_2 & I_2 \\ I_2 & I_2 \end{bmatrix} \quad (116)$$

$$\mathbf{C}_o(\mathbf{q}, \mathbf{w}) = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (117)$$

$$\mathbf{g}_o(\mathbf{q}) = \begin{bmatrix} \mathbf{m}_1 l_{c1} g \sin \theta_1 \\ \mathbf{0} \end{bmatrix} \quad (118)$$

Por otro lado, la ley de adaptación se puede considerar como:

$$\hat{\mathbf{a}} = \Gamma \bar{\mathbf{Y}}^T(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r) \boldsymbol{\sigma} \quad (119)$$

Finalmente, la ley de control queda de la siguiente manera:

$$\boldsymbol{\tau} = \begin{bmatrix} \bar{\mathbf{y}}_1 \\ \bar{\mathbf{y}}_2 \end{bmatrix} [\mathbf{m}_2] + \begin{bmatrix} \mathbf{m}_1 l_{c1}^2 + I_1 + I_2 & I_2 \\ I_2 & I_2 \end{bmatrix} \ddot{\mathbf{q}}_r + \begin{bmatrix} \mathbf{m}_1 l_{c1} g \sin \theta_1 \\ \mathbf{0} \end{bmatrix} + \mathbf{K}_D \boldsymbol{\sigma} \quad (120)$$

Con la siguiente ecuación en lazo cerrado:

$$\ddot{\mathbf{q}} = \mathbf{H}^{-1}(\mathbf{q}) [\boldsymbol{\tau} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})] \quad (121)$$

Y la siguiente ley de adaptación:

$$\hat{\mathbf{a}} = \mathbf{0.005} \bar{\mathbf{Y}}^T(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r) \boldsymbol{\sigma} \quad (122)$$

2 CONSIDERACIONES PREVIAS PARA LA SIMULACIÓN

A fin de analizar el comportamiento de las estrategias de control previamente diseñadas, se realizaron simulaciones numéricas del sistema en lazo cerrado siguiendo una trayectoria deseada utilizando el software MATLAB. Los valores nominales de los parámetros dinámicos del robot se obtuvieron de Slotine (1991). Se consideraron las siguientes posiciones iniciales de las juntas:

$$\mathbf{q} = \left[0, \frac{\pi}{2}\right]^T \quad (123)$$

Por otro lado, se consideraron las siguientes trayectorias deseadas:

Trayectoria curva suavizada:

$$\mathbf{q}_d(t) = [-R\cos(\omega_i t) - R, -R\cos(\omega_i t) + 5R] \quad (124)$$

Donde $R=0.5$ m es el radio de la trayectoria.

Este tipo de trayectoria se realizará en un tiempo de 10s y el radio será de 0.5 m, se eligió este tipo de trayectoria porque presenta un tiempo considerable y una trayectoria no complicada para un primer análisis de resultados de los controladores. Con esta trayectoria observaremos cuál de ellos presenta un comportamiento casi exacto frente al trabajo deseado porque no presenta una gran complejidad.

Trayectoria circular con radio de 0.5 m:

$$\mathbf{P}(t) = [0.5 + 0.5 * \cos(t), 0.5 + 0.5 * \sen(t)] \quad (125)$$

Se presenta una trayectoria circular con un radio de 0.5m y un tiempo de 10s, esta presenta una dificultad mayor para el seguimiento de trayectoria en el controlador por el constante cambio de direcciones a tener en cuenta, más el tiempo considerable es de gran ayuda para los distintos controladores.

Trayectoria lineal con cambio de dirección de 87 °:

$$\mathbf{P1}(t) = [t, 0.7 * t + 1] \quad (126)$$

$$\mathbf{P2}(t) = [t, 0.4 * t + 4.1] \quad (127)$$

Finalmente se presenta una trayectoria con un cambio de dirección rápido, la cuál se realizará en un tiempo de 1.5s, esta representa un desafío para los controladores trabajados porque si bien es una trayectoria lineal al presentar un cambio brusco y al tener que efectuarse en un tiempo muy corto, evidenciará la capacidad de los controladores para realizar el seguimiento adecuado de trayectoria.

Además, se consideraron las siguientes ganancias para los controladores PD, PD aumentado y de torque computado: $K_d = \text{diag}\{2\zeta_1\omega_{n1}^2, \dots, 2\zeta_1\omega_{nn}^2\}$ y $K_p = \{\omega_{n1}^2, \dots, \omega_{nn}^2\}$ para que los polos dominantes del sistema tengan un coeficiente de amortiguamiento de $\zeta = 0.5$, y una frecuencia natural de $\omega_n = 5 \text{ rad s}^{-1}$ para $n = 1, 2$.

Para el caso del controlador adaptativo se consideraron las siguientes ganancias por medio del método empírico de ensayo y error, de esta manera se escogieron los valores que lograron un óptimo resultado, siendo lo esperando, si la constante K_d y Λ es pequeña el controlador no llegara al resultado esperado, más al considerar valores de sigma Γ se probó valores pequeños al obtener mejor resultado con ellos aunque se esperaría que al estar directamente relacionado con la ley de adaptación a mayor valor el controlador haría la adaptación de manera más rápida.

Por esta razón se consideró los siguientes valores a usar para el controlador adaptativo:

$$K_D = 100I, \Lambda = 10I \text{ y } \Gamma = 0.005$$

Los parámetros nominales del robot son los siguientes:

$$m_1 = 1$$

$$m_2 = 2$$

$$l_1 = 1$$

$$l_2 = 1$$

$$I_{c1} = 0.5$$

$$I_{c2} = 0.6$$

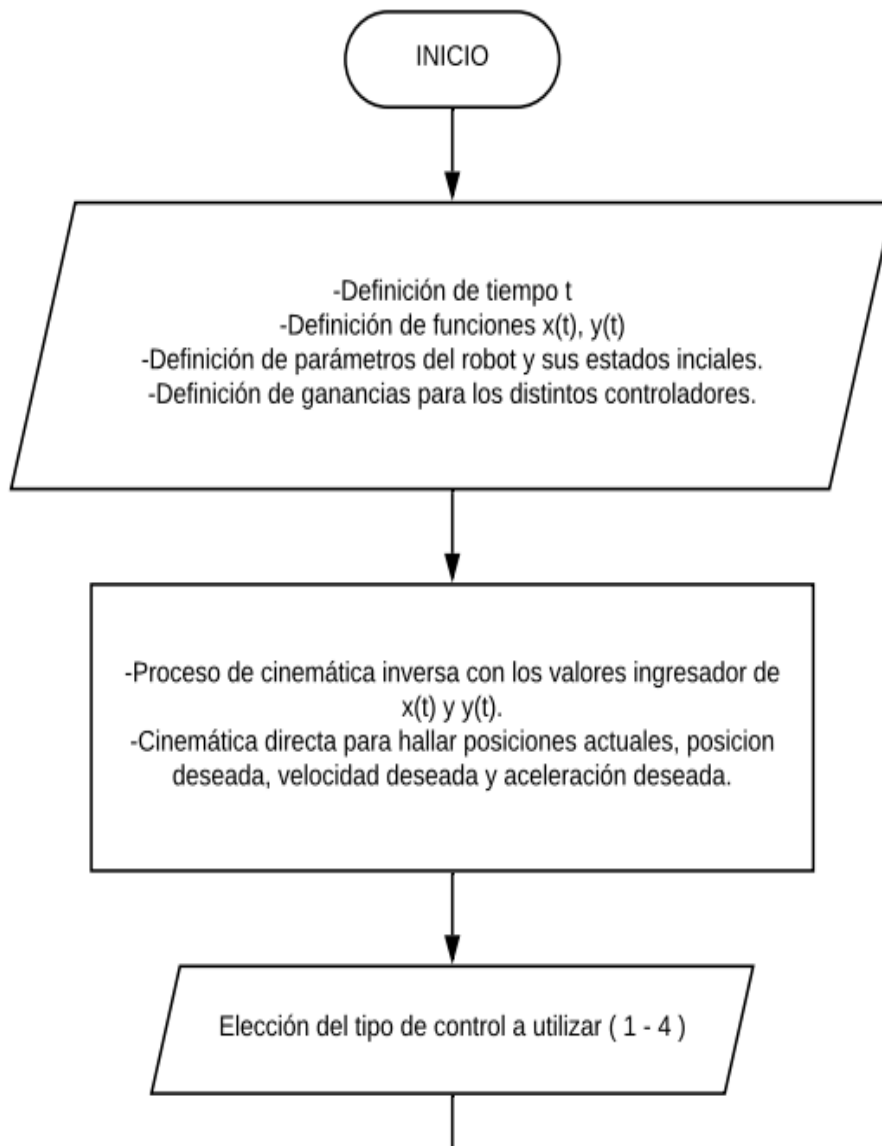
$$I_1 = 0.12$$

$$I_2 = 0.25$$



3 DIAGRAMA DE FLUJO DEL CÓDIGO MATLAB

Para la representación del código desarrollado se presenta la Figura 15, en la cuál se detalla el algoritmo que sigue el código para la obtención de las gráficas, pasando desde la etapa de definiciones de parámetros, procesos y la presentación en pantalla.



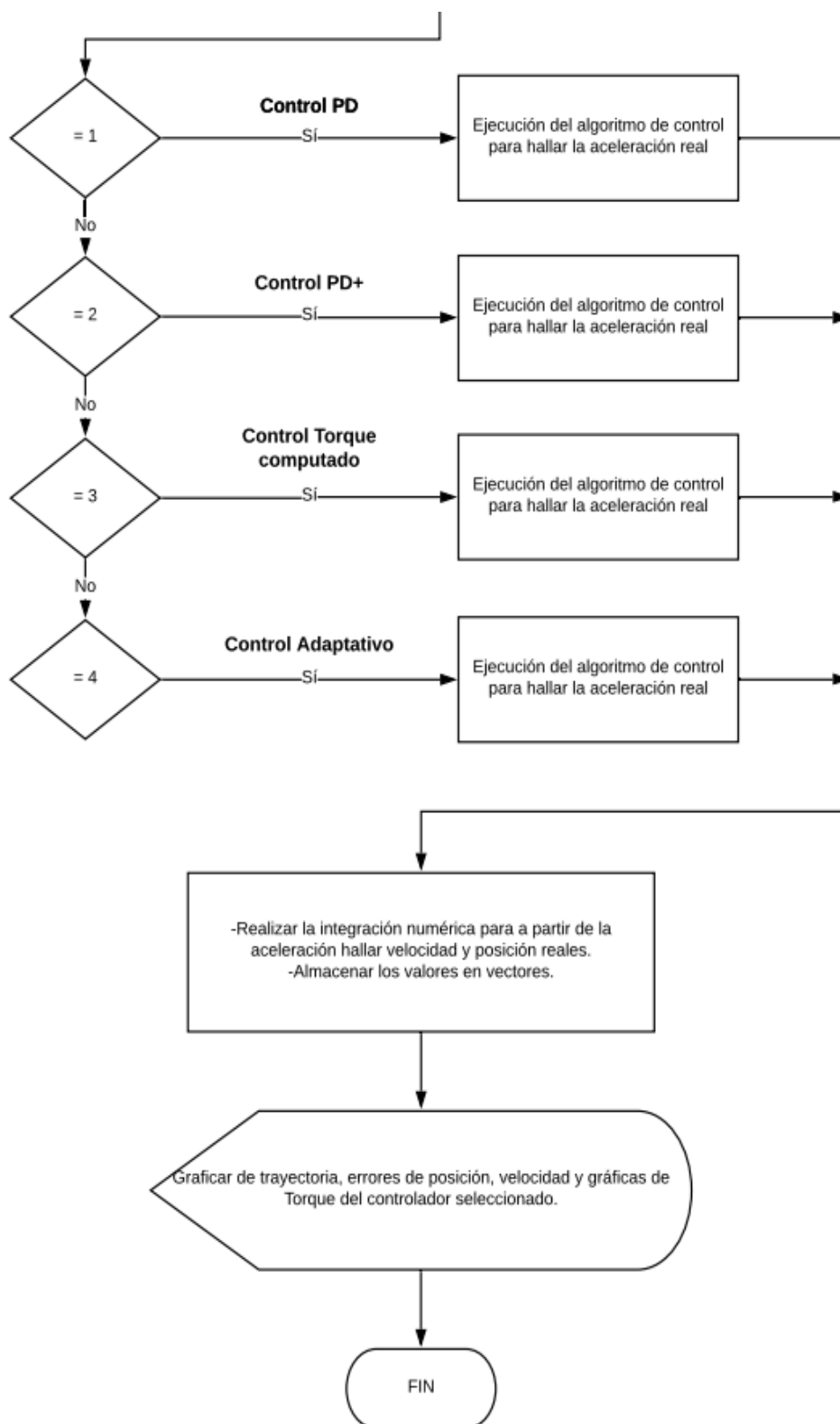


Figura 15: Diagrama de Flujo del algoritmo usado

Fuente: Propia



CAPITULO IV
RESULTADOS Y DISCUSION

CAPITULO IV

RESULTADOS Y DISCUSION

1. TRAYECTORIA CURVA SUAVIZADA

1.1 TRAYECTORIA

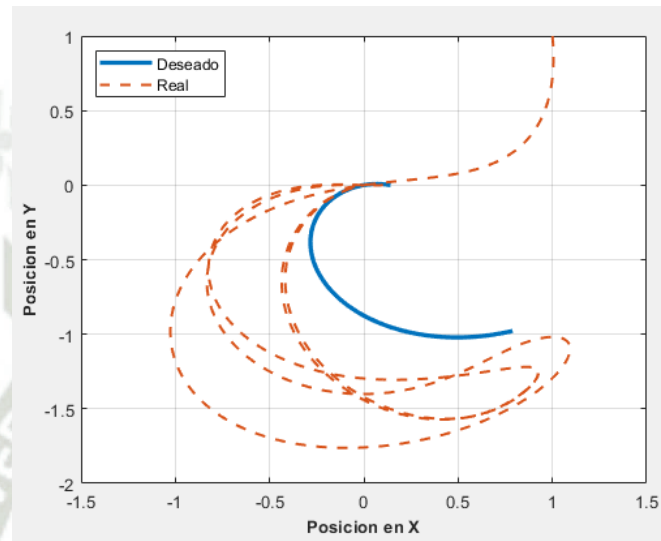


Figura 16. Trayectoria suavizada con controlador PD

Fuente: Propia

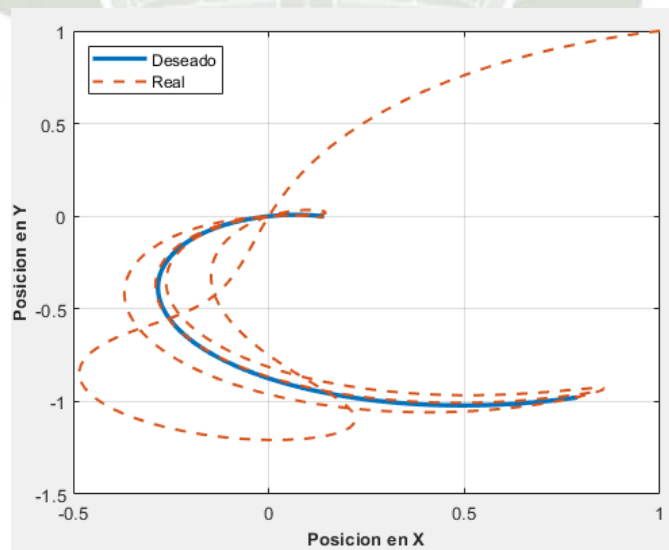


Figura 17. Trayectoria suavizada con controlador PD aumentado

Fuente: Propia

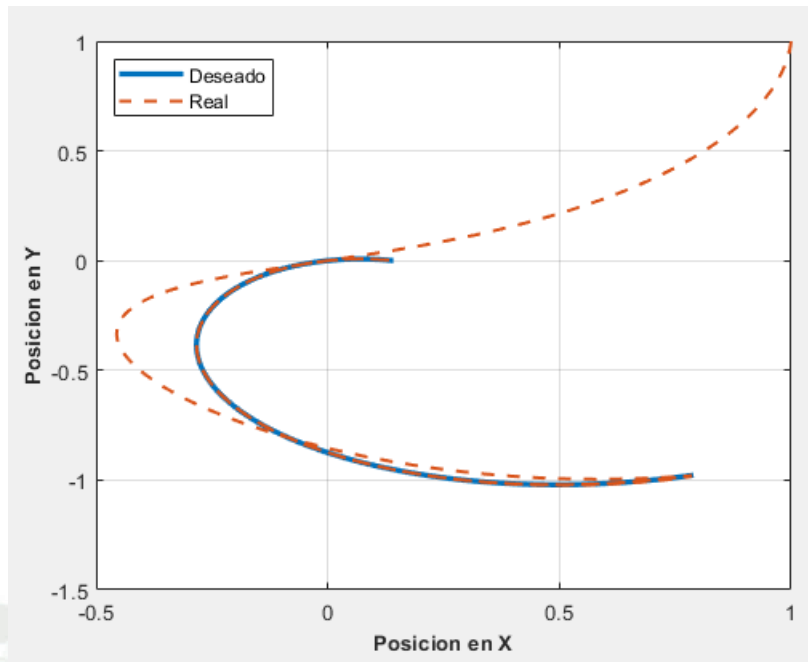


Figura 18. Trayectoria suavizada con controlador Torque Computado

Fuente: Propia

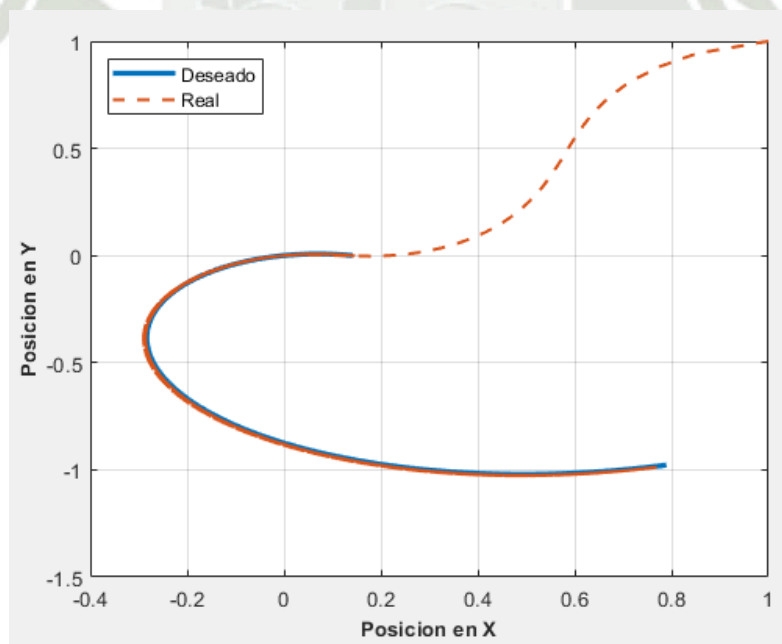


Figura 19. Trayectoria suavizada con controlador adaptativo

Fuente: Propia

Se observa de la Figura 16 a la Figura 19, las trayectorias alcanzadas por cada uno de los controladores, de la simulación obtenemos una diferencia clara con los últimos dos controladores, consiguiendo el seguimiento óptimo de la trayectoria en el control adaptativo, esta se desarrollará mejor con las figuras siguientes, mientras que observamos que los controladores que no toman en cuenta la dinámica del robot y los parámetros no lineales presentan problemas para el seguimiento de trayectoria al punto de no conseguir igualarla en ningún punto como es el caso del controlador PD.

1.2 ERRORES DE POSICIÓN

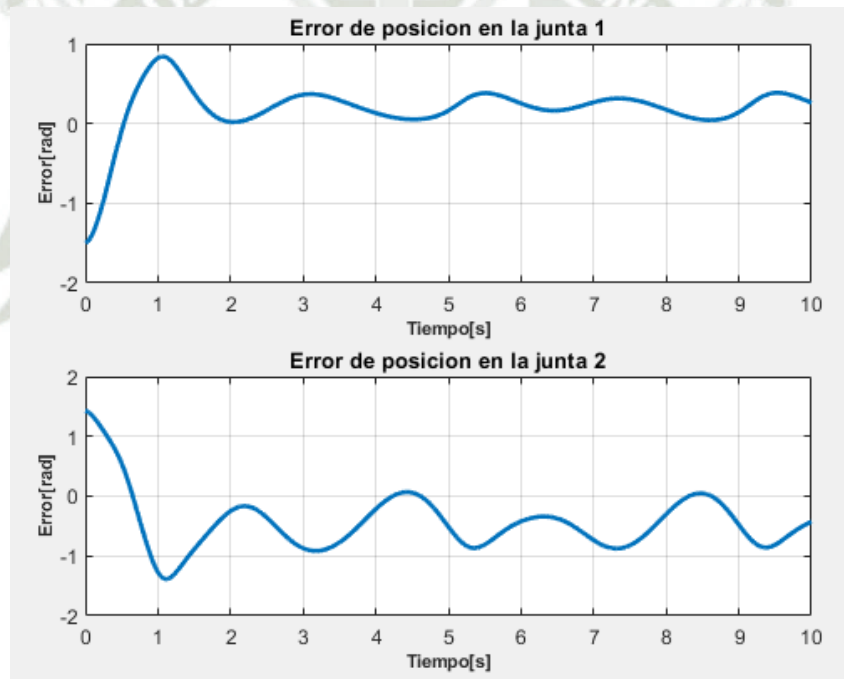


Figura 20. Error de posición con trayectoria suavizada con controlador PD

Fuente: Propia

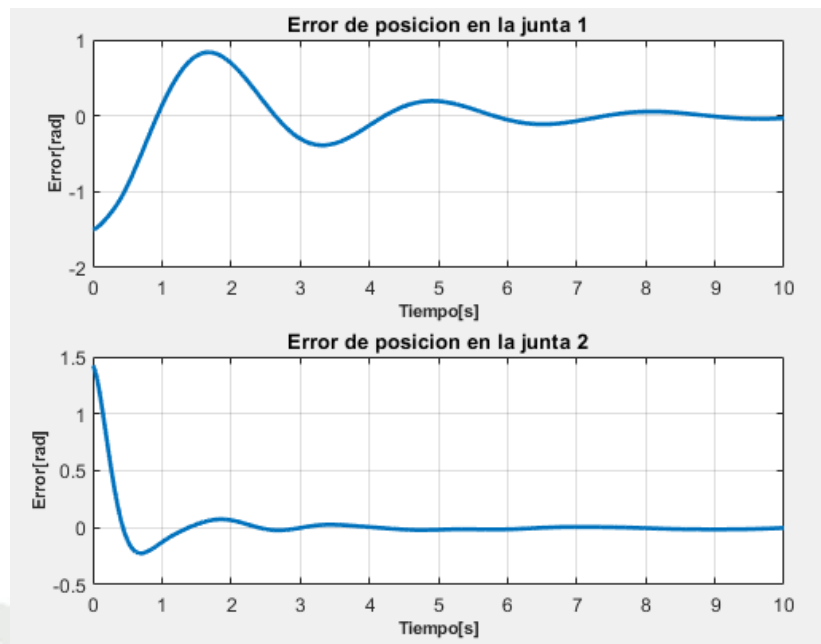


Figura 21. Error de posición con trayectoria suavizada con controlador PD aumentado

Fuente: Propia

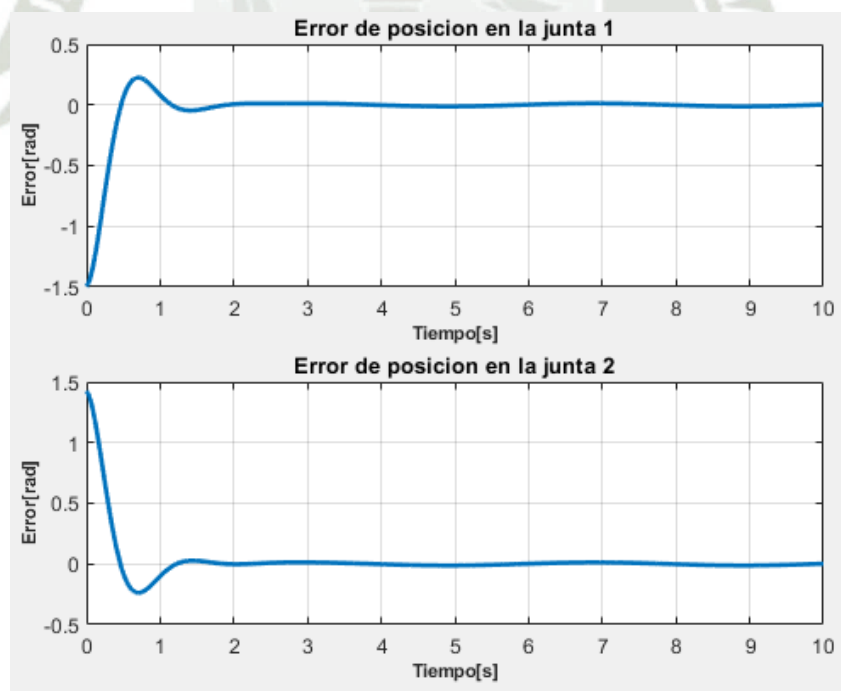


Figura 22. Error de posición con trayectoria suavizada con controlador Torque Computado

Fuente: Propia

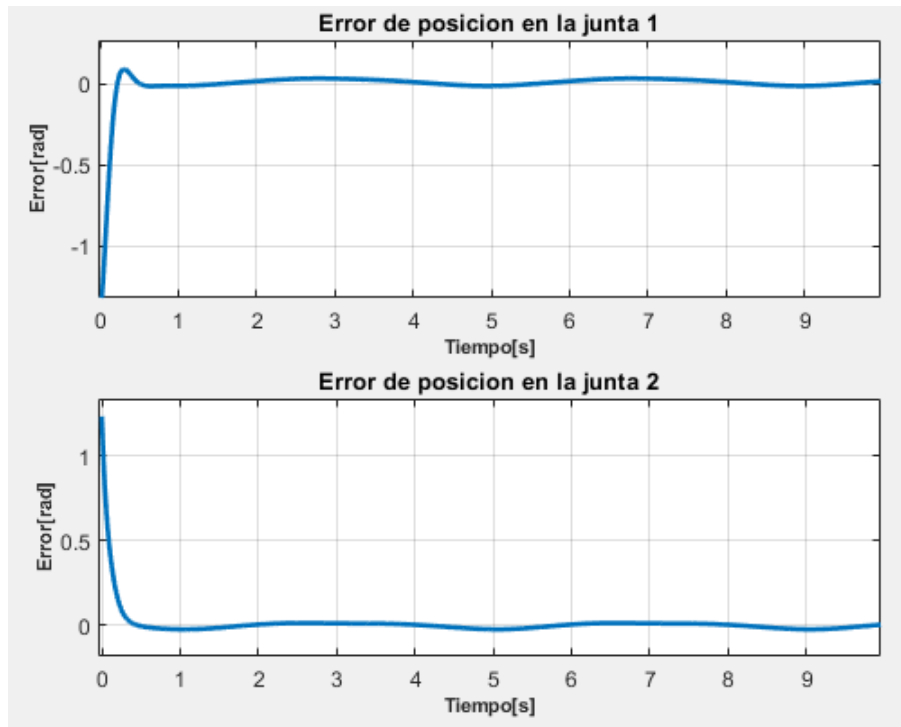


Figura 23. Error de posición con trayectoria suavizada con controlador adaptativo

Fuente: Propia

Se observa de la Figura 20 a la Figura 23, los errores de posición de los controladores, podemos obtener de la simulación una diferencia clara con los últimos dos controladores, de la misma manera que el apartado anterior, consiguiendo el seguimiento óptimo de la trayectoria deseada en un tiempo bajo para el control adaptativo, esta se desarrollará mejor con las figuras siguientes, mientras que observamos que los controladores que no toman en cuenta la dinámica del robot y los parámetros no lineales presentan problemas para el seguimiento de trayectoria al punto de no conseguir igualarla en ningún punto como es el caso del controlador PD.

1.3 ERRORES DE VELOCIDAD

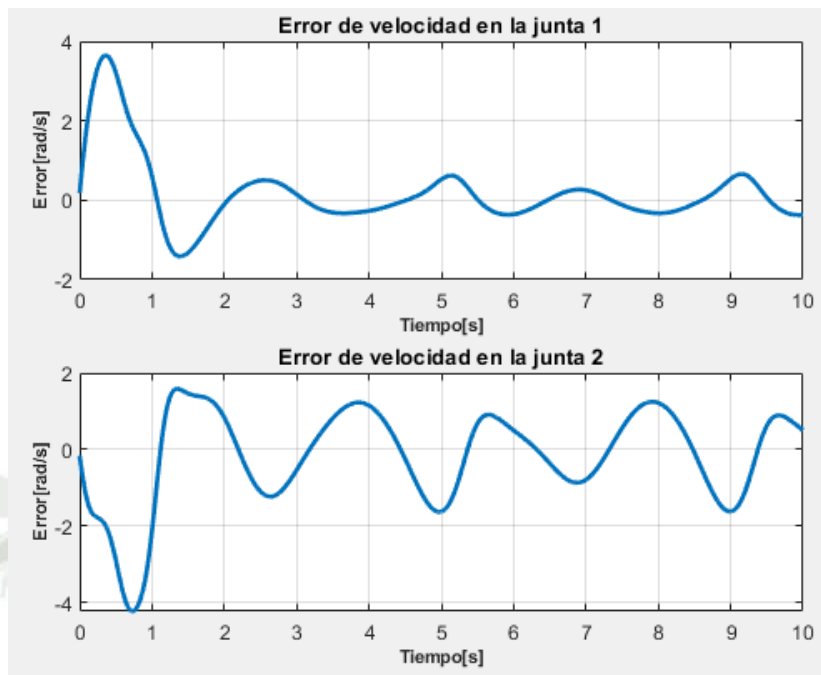


Figura 24. Error de velocidad con trayectoria suavizada con controlador PD

Fuente: Propia

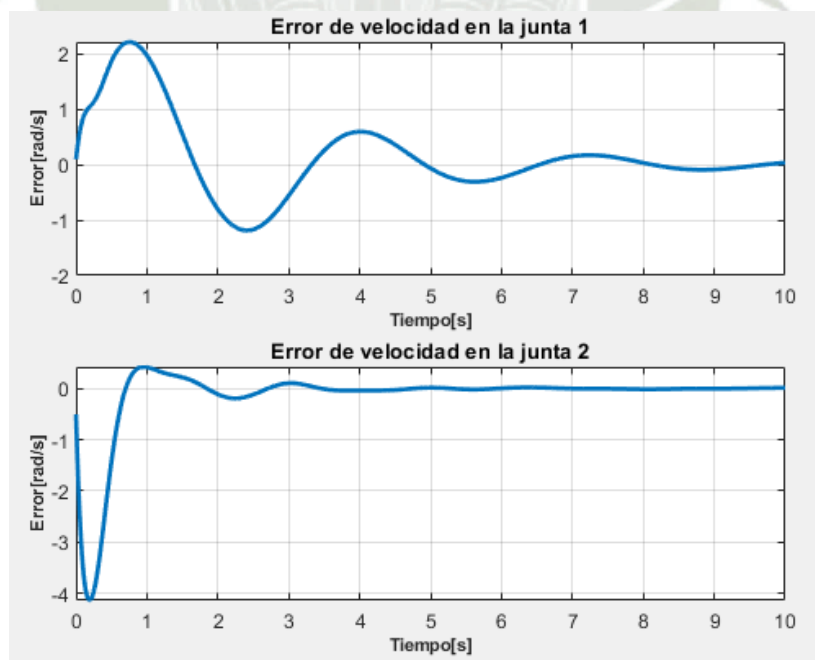


Figura 25. Error de velocidad con trayectoria suavizada con controlador PD aumentado

Fuente: Propia

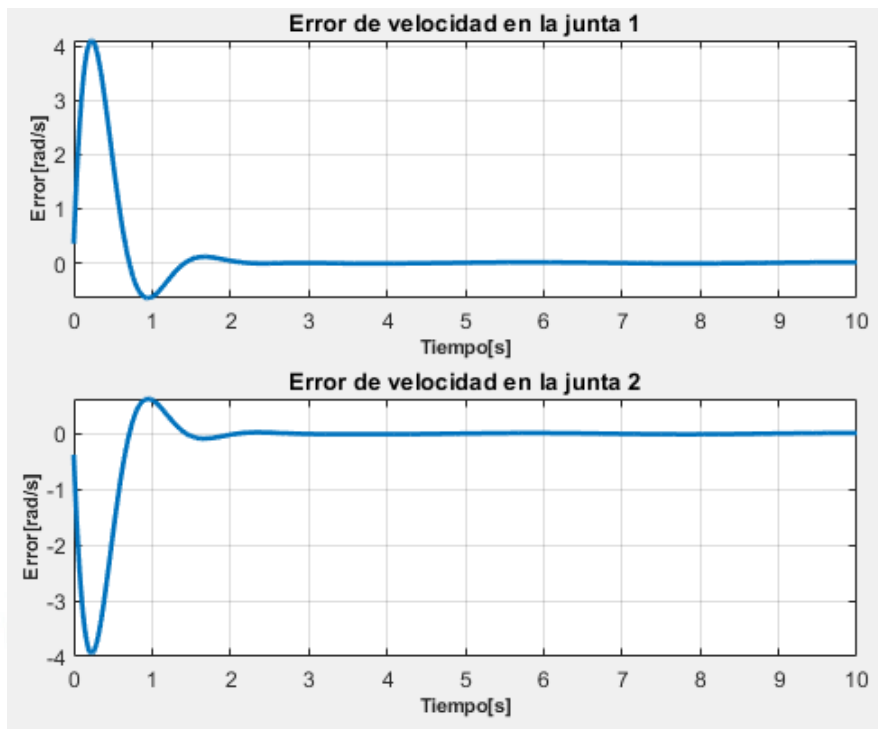


Figura 26. Error de velocidad con trayectoria suavizada con controlador Torque computado

Fuente: Propia

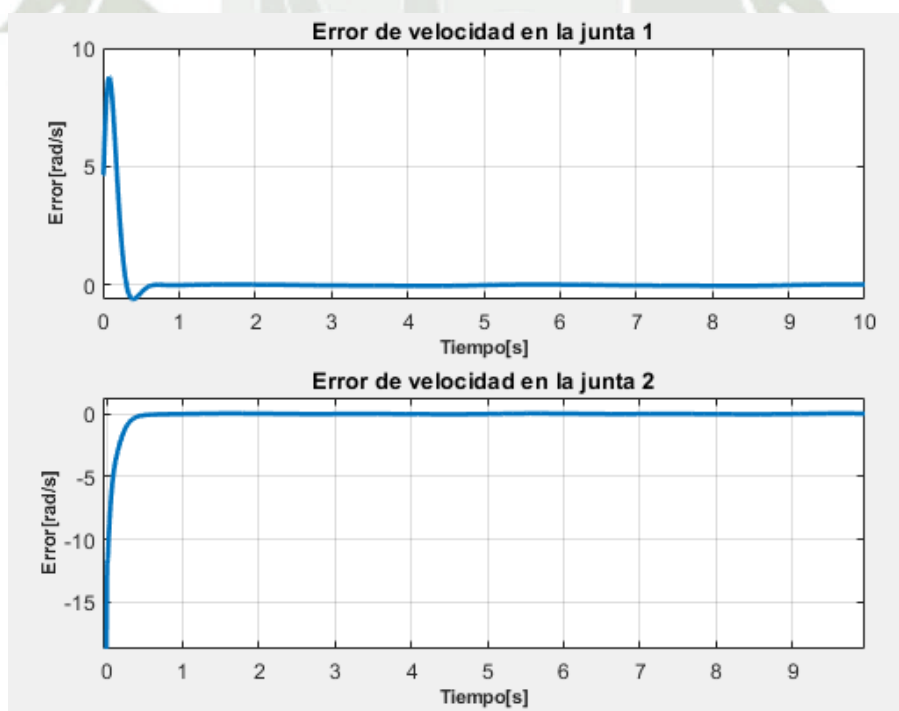


Figura 27. Error de velocidad con trayectoria suavizada con controlador adaptativo

Fuente: Propia

Desde la Figura 24 a la Figura 27 se observa el error de velocidad en juntas, pudiéndose ver que el mayor error se encuentra en el controlador PD, oscilando entre valores que son diferentes de 0 y una constante variación a lo largo del tiempo, por lo tanto, el seguimiento de trayectoria no es el adecuado. Estas variaciones van disminuyendo conforme cambiamos la estrategia de control, el control PD+ presenta oscilaciones en la junta 1 pero en la junta 2 se estabiliza y en los siguientes se presenta un adecuado seguimiento con errores bajos, comparándolos en la tabla 2 presentada más adelante.

TRAYECTORIA SUAVIZADA																
	CONTROL PD				CONTROL PD+				C. TORQUE COMPUTADO				CONTROL ADAPTATIVO			
	EP1	EP2	EV1	EV2	EP1	EP2	EV1	EV2	EP1	EP2	EV1	EV2	EP1	EP2	EV1	EV2
IAE	282.66	561.82	526.81	1002.4	251.89	57.165	475.31	217.57	55.204	52.841	210.47	202.19	31.438	24.035	186.13	158.23
ITAE	1136.9	2526	1488.8	4036.1	558.35	35.006	1122	161.42	53.763	52.274	118.69	115.27	75.169	59.639	112.82	92.165
ISE	138.86	432.73	799.47	1617.2	170.09	70.989	548.47	530.6	42.356	38.319	558.76	515.01	15.846	8.4728	1070	1149.9
MP	0.84	1.38	1.41	1.58	0.83	0.221	1.16	0.42	0.224	0.236	0.65	0.62	0.0895	0.008	0.59	0
Ts	0.48	0.83	1	1.1	0.85	0.4	1.59	0.64	0.44	0.43	0.67	0.68	0.21	0.26	0.3	0.33
EP1: Error de posición en la junta 1								EV2: Error de velocidad en la junta 2								
EP2: Error de posición en la junta 2								EV1: Error de velocidad en la junta 2								

Tabla 2: Indicadores de rendimiento para la trayectoria suavizada.

Fuente: Propia

Como se observa en la tabla 2, tanto para la medición de indicadores IAE, ITAE, ISE el controlador adaptativo presenta un mejor desempeño frente a los demás controladores ya que posee el valor mínimo. Esto nos indica que presenta una menor cantidad de error a lo largo del tiempo para el caso del seguimiento de posición, demostrándose también con el Máximo Sobre impulso (Mp) y el tiempo de establecimiento (Ts) contando con valores para la junta 1 con valores de 8% y 0.21s, así como la junta número 2 de 0.8% y 0.26s respectivamente, esto nos indica que presenta una gran velocidad de respuesta y un sobre impulso aceptable estando dentro de los parámetros.

Se observa que el ITAE para el controlador adaptativo presenta valores superiores para la posición en juntas 1 y 2, lo que representa una mayor variación a lo largo del tiempo, que por mínima que sea este índice penaliza con mayor fuerza mientras más largo sea el tiempo.

1.4 TORQUES APLICADOS

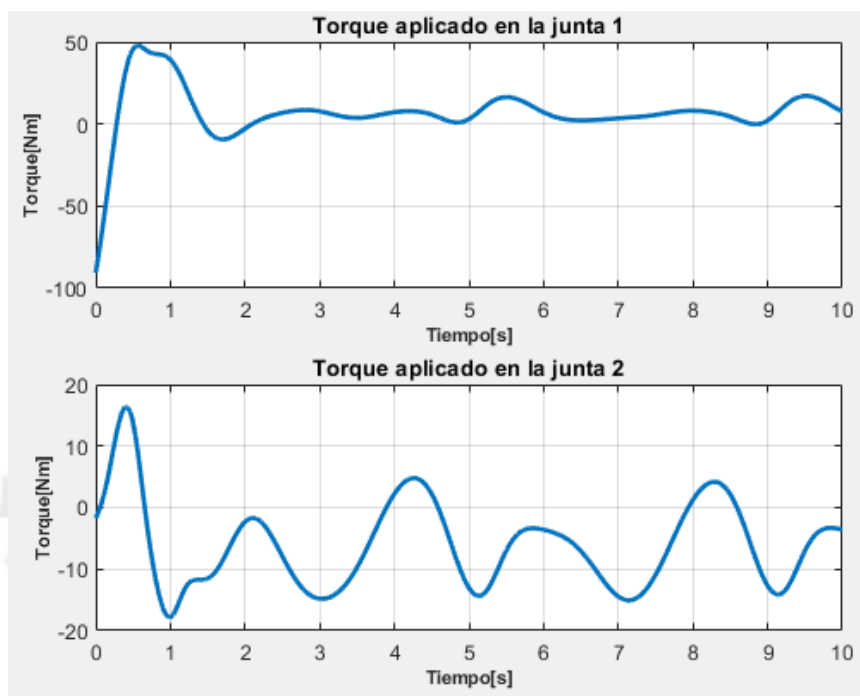


Figura 28. Torque aplicado con trayectoria suavizada con controlador PD

Fuente: Propia

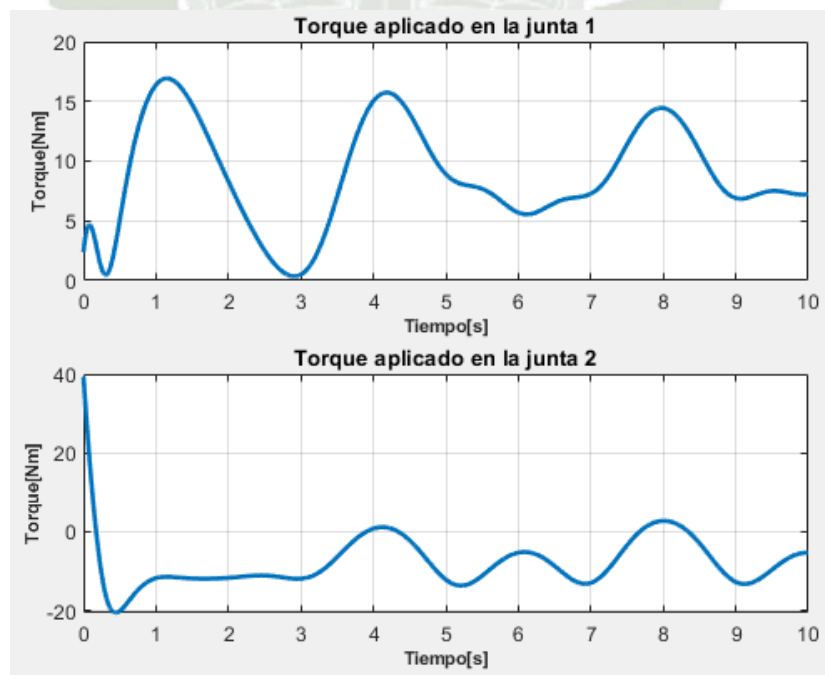


Figura 29. Torque aplicado con trayectoria suavizada con controlador PD aumentado

Fuente: Propia

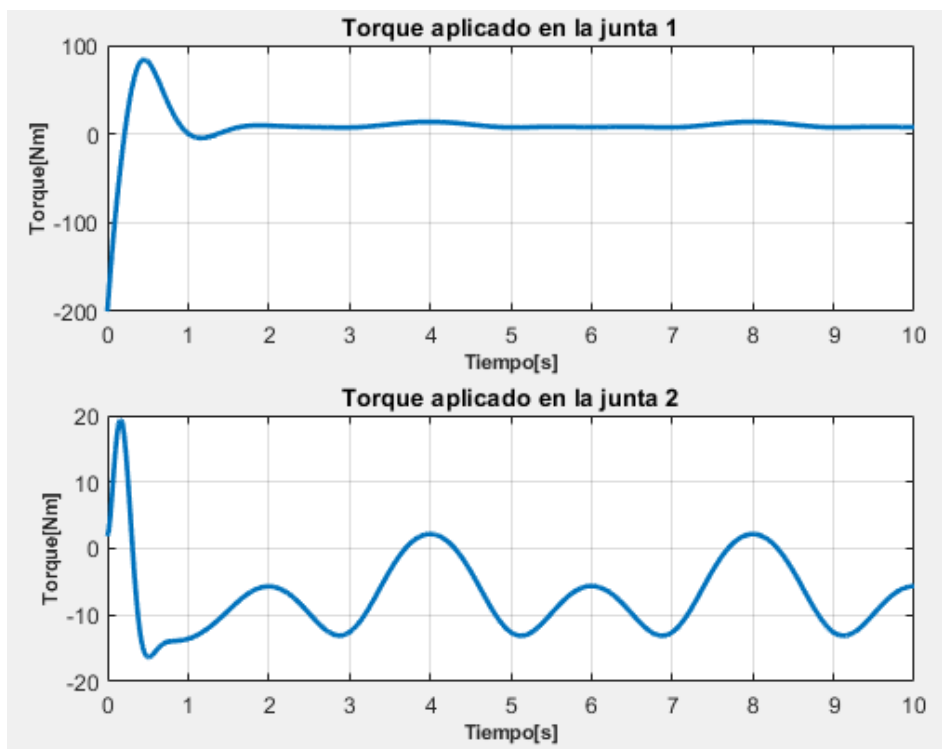


Figura 30. Torque aplicado con trayectoria con controlador Torque computado

Fuente: Propia

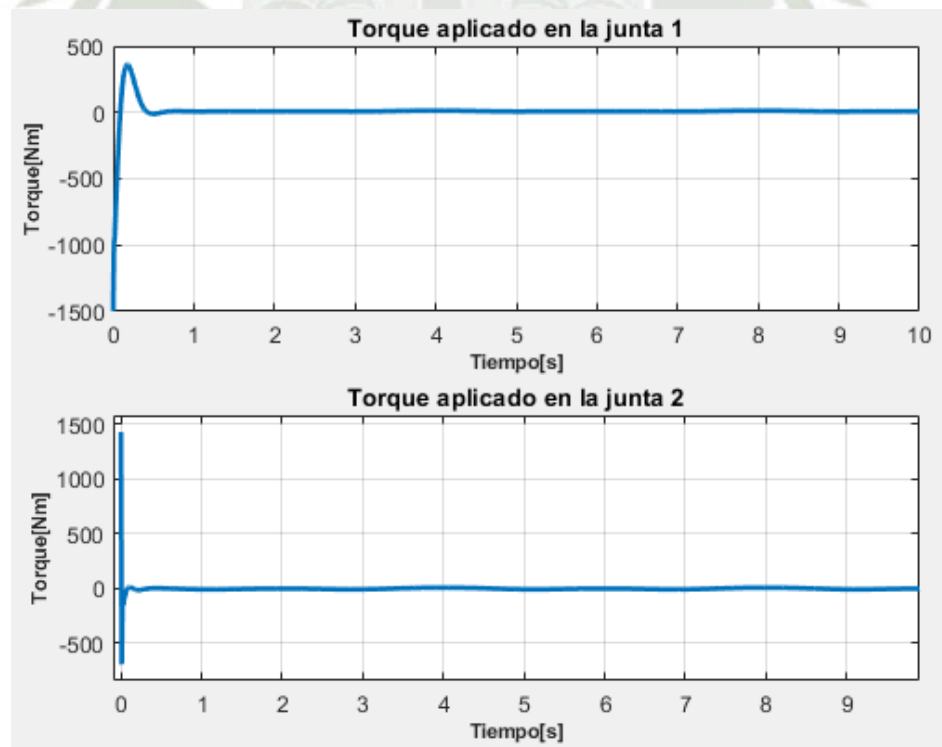


Figura 31. Torque aplicado con trayectoria suavizada con controlador adaptativo

Fuente: Propia

De la Figura 28 a la Figura 31 se presentan los torques aplicados a las juntas, se observa que el controlador adaptativo es el único que alcanza la estabilidad, por ende, alcanza la trayectoria, en un tiempo de 0.4s mientras que los demás controladores presentan una oscilación constante a lo largo del tiempo siendo el controlador de torque computado el más cercano a lograr la trayectoria para la junta 1 pero en la junta 2 presenta constantes cambios en el torque no llegando a una estabilidad. También se observa que el controlador adaptativo sugiere una cantidad muy elevada de torque de hasta 1500 Nm para poder alcanzar la trayectoria en el menor tiempo posible, lo que también se traduce en errores en la velocidad.

2. TRAYECTORIA CIRCULAR CON RADIO REDUCIDO

2.1 TRAYECTORIA

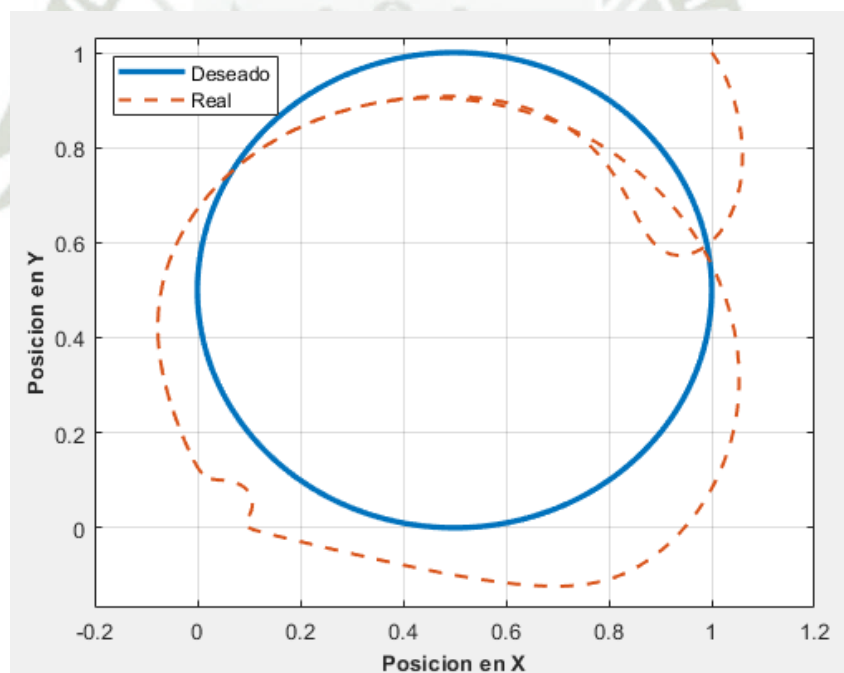


Figura 32 Trayectoria circular con controlador PD

Fuente: Propia

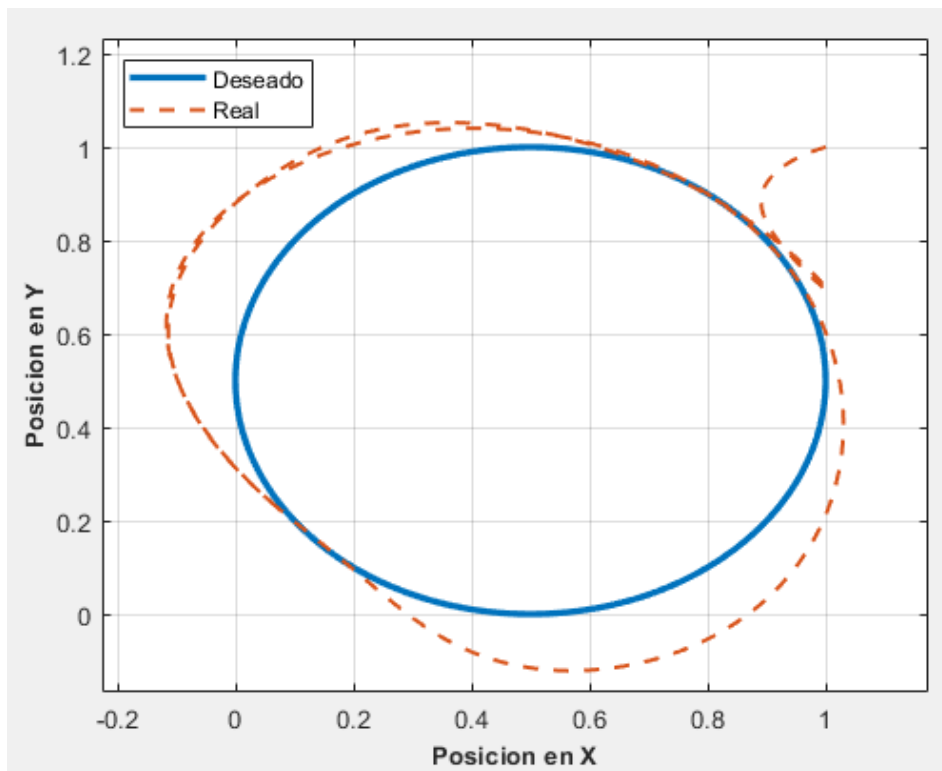


Figura 33. Trayectoria circular con controlador PD aumentado

Fuente: Propia

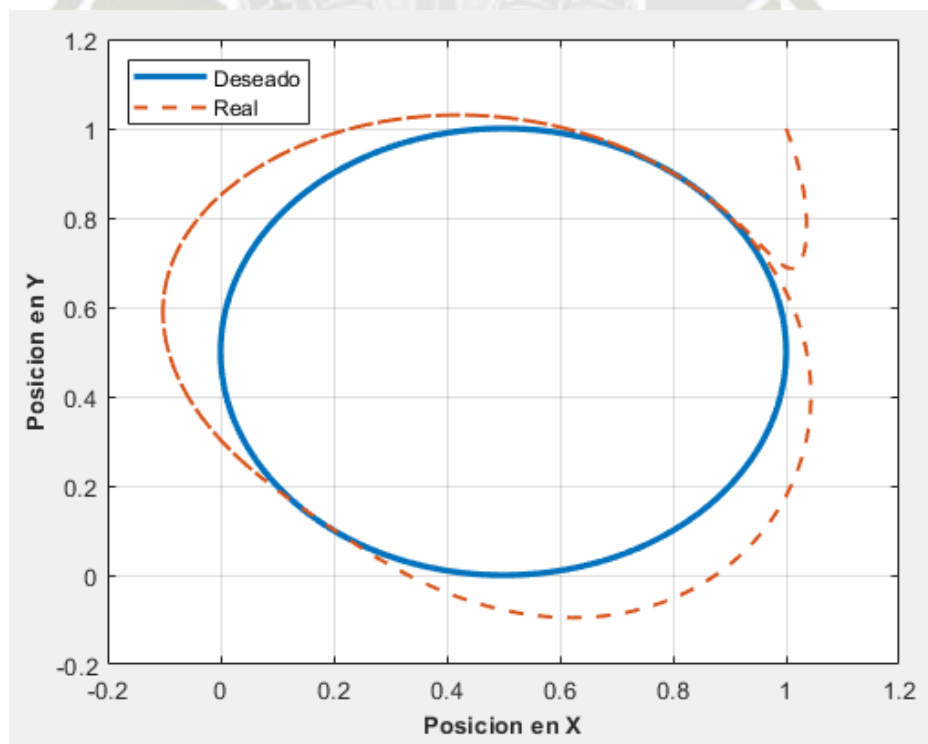


Figura 34 Trayectoria circular con controlador Torque computado

Fuente: Propia

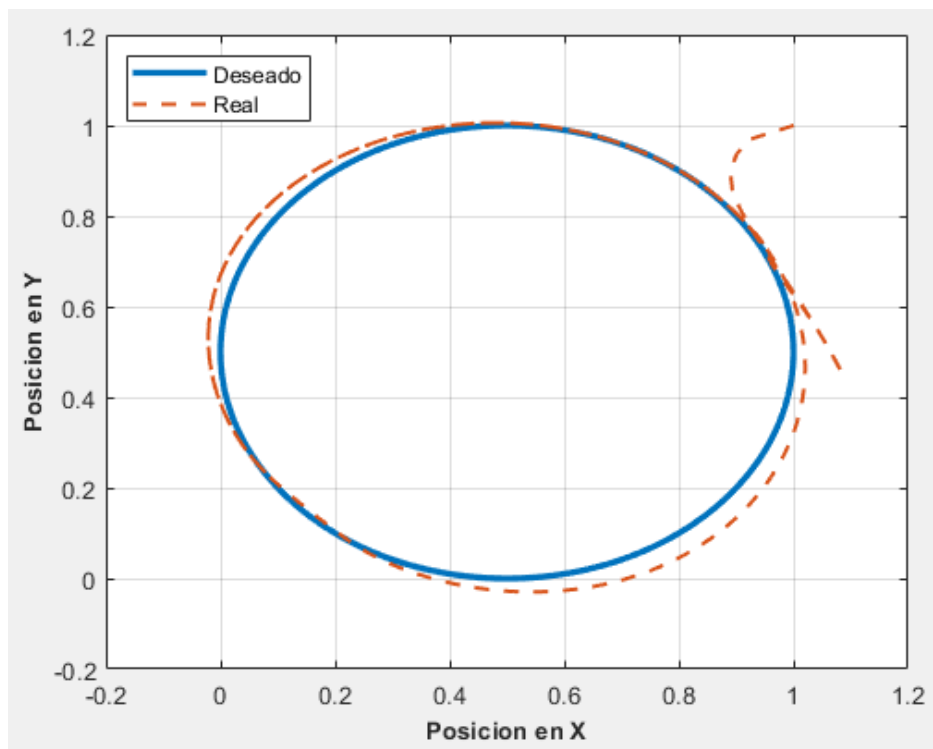


Figura 35. Trayectoria circular con controlador adaptativo

Fuente: Propia

De la Figura 32 a la Figura 35 se presenta el seguimiento de trayectoria de los controladores frente a una trayectoria circular con un radio de 0.5. De la misma manera que en la trayectoria anterior se observa un mejor comportamiento en la gráfica del control adaptativo consiguiendo seguir la trayectoria casi de manera ideal pero aún con pequeñas variaciones y oscilaciones mayores al inicio del seguimiento, seguido del control por Torque Computado el cuál presenta una menor oscilación al inicio de la trayectoria, pero este no consigue igualarla, presentando mayores variaciones, lo mismo sucede con el control PD+. Mientras que el control PD no consigue igualar la trayectoria en ningún momento. Estos comportamientos se observarán mejor con las siguientes figuras e índices

2.2 ERRORES DE POSICIÓN

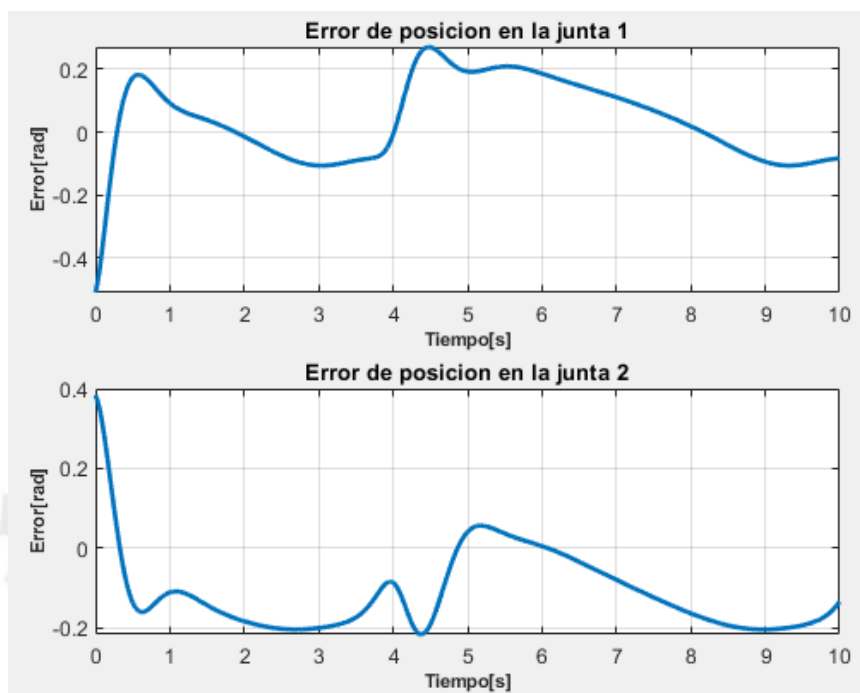


Figura 36. Error de posición con trayectoria circular con controlador PD

Fuente: Propia

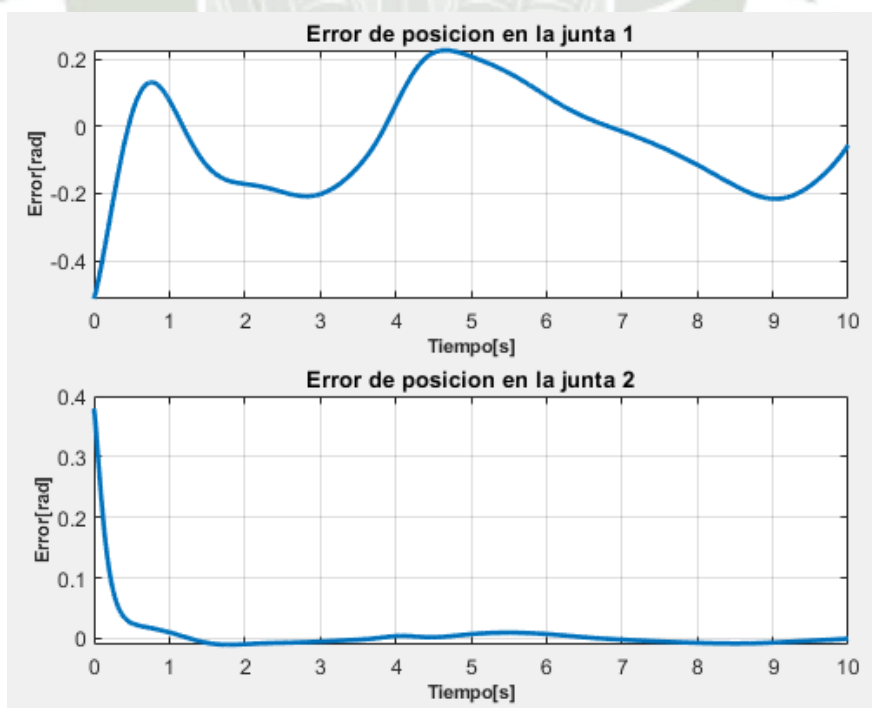


Figura 37. Error de posición con trayectoria circular con controlador PD aumentado

Fuente: Propia

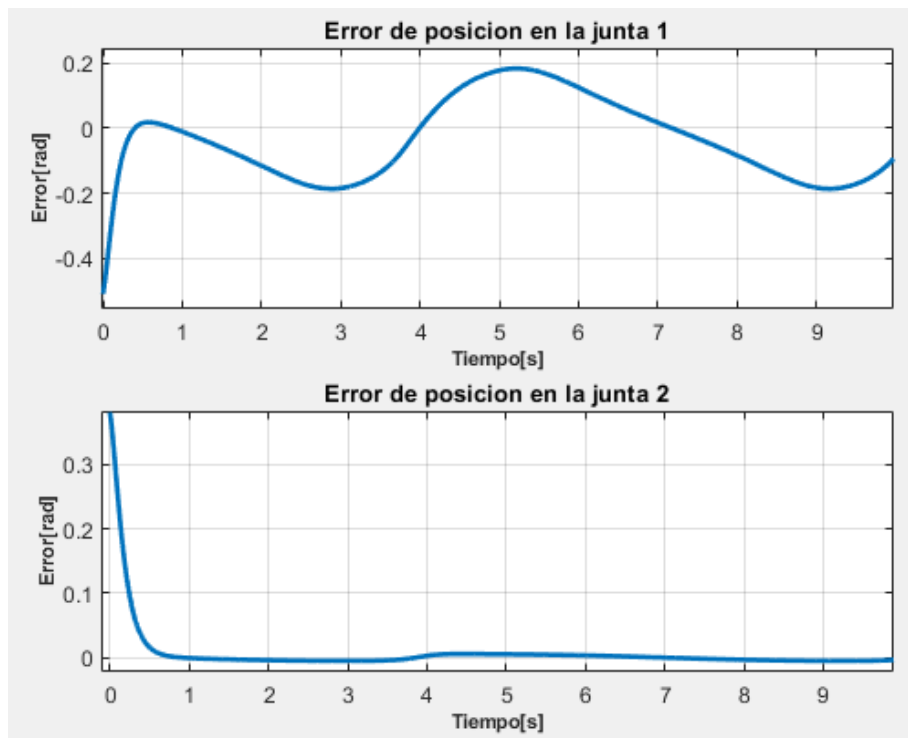


Figura 38. Error de posición con trayectoria circular con controlador Torque computado

Fuente: Propia

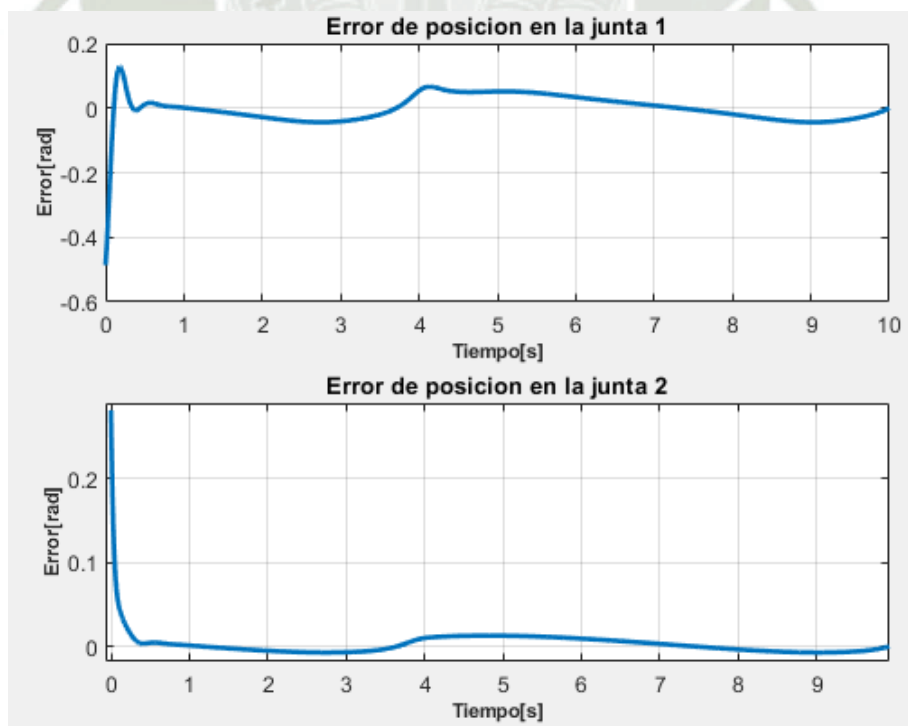


Figura 39. Error de posición con trayectoria circular con controlador adaptativo

Fuente: Propia

De la Figura 36 a la Figura 39, se presentan los errores de posición en las juntas para la trayectoria circular, Se observa que el control PD no logra una tendencia a llevar el error a 0 en ninguna de las juntas, mientras que el control PD+ y Torque computado presentan problemas a la hora de seguir la trayectoria de la junta 1 que sigue presentando oscilaciones durante el tiempo de simulación esto se debe a la necesidad de cubrir un mayor espacio para la junta 1 y se observa que para la junta 2 si obtiene la tendencia a 0 deseada, es decir que la trayectoria 2 es seguida satisfactoriamente.

De la misma manera se observa que el controlador adaptativo logra la tendencia a 0 deseada en ambas juntas, siguiendo satisfactoriamente la trayectoria y haciéndolo en un corto tiempo. Presentando el mejor desempeño para el seguimiento de trayectoria circular, el cual se cuantificará más adelante.

2.3 ERRORES DE VELOCIDAD

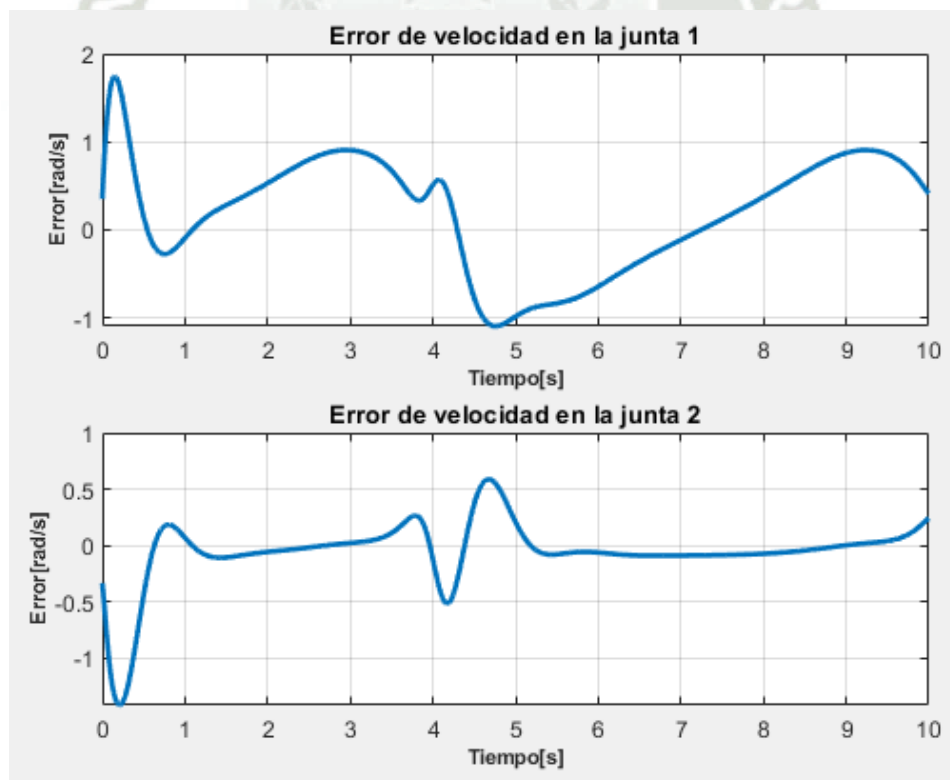


Figura 40. Error de velocidad con trayectoria circular con controlador PD

Fuente: Propia

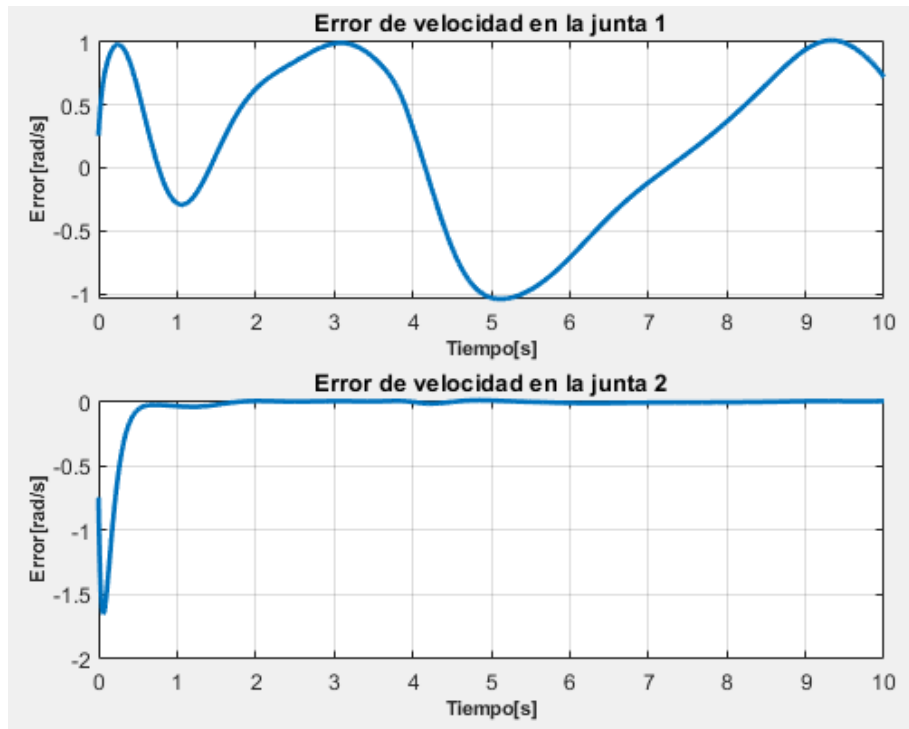


Figura 41. Error de velocidad con trayectoria circular con controlador PD aumentado

Fuente: Propia

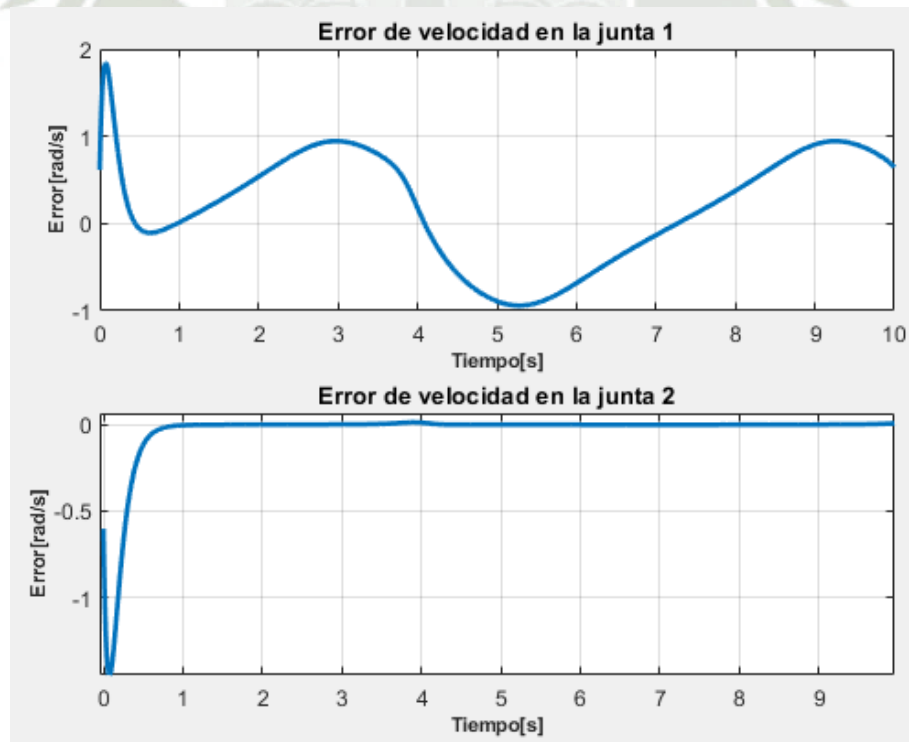


Figura 42. Error de velocidad con trayectoria circular con controlador Torque computado

Fuente: Propia

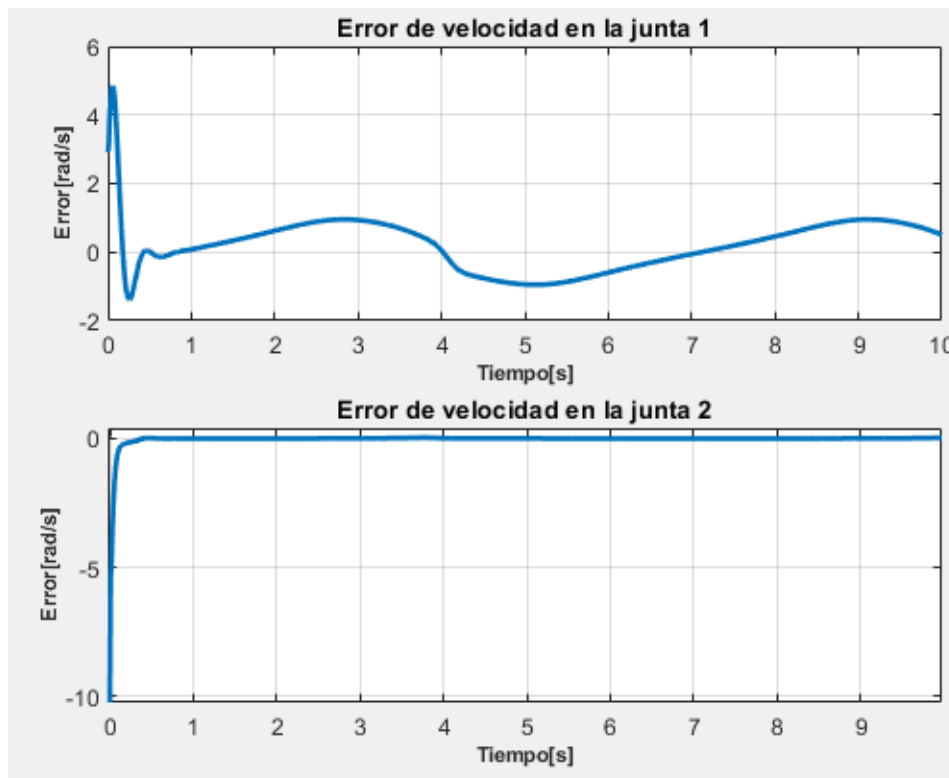


Figura 43. Error de velocidad con trayectoria circular con controlador adaptativo

Fuente: Propia

De la Figura 40 a la Figura 43 tenemos los errores de velocidad en las juntas para la trayectoria circular, al igual que las figuras de error de posición el control PD presenta el mismo comportamiento, presentando oscilaciones a lo largo del tiempo sin llegar al 0, así mismo los controladores siguientes consiguen un error de 0 para la junta número 2 pero oscilaciones considerables para la junta 1, no logrando igualar a la velocidad deseada.

De la misma manera el controlador adaptativo, presentando una gran velocidad al inicio de la trayectoria, pero en un corto tiempo logra una tendencia al 0 con un error menor y consiguiendo un error de 0 para la junta 2 con un gran cambio de velocidad al inicio.

TRAYECTORIA CIRCULAR																
	CONTROL PD				CONTROL PD+				C. TORQUE COMPUTADO				CONTROL ADAPTATIVO			
	EP1	EP2	EV1	EV2	EP1	EP2	EV1	EV2	EP1	EP2	EV1	EV2	EP1	EP2	EV1	EV2
IAE	108	96.902	574.27	115.59	135.89	12.22	610.94	43.904	110.68	10.502	568.81	40.193	32.086	7.6906	622.08	43.174
ITAE	534.6	470.64	2916	350.97	649.98	27.283	3152	33.753	576.78	19.383	2981.6	15.354	147.63	30.889	3025.7	26.998
ISE	16.309	12.41	440.82	64.325	24.541	1.4762	478.54	40.938	17.085	1.6273	438.92	37.6	2.2173	0.2866	643.72	195.78
MP	0.213	0.153	0.94	0.4	0.225	0.009	1.03	0	0.186	0	0.945	0	0.126	0.01	1.38	0
Ts	0.25	0.29	0.45	0.53	0.43	0.75	0.73	0.45	0.33	0.46	0.38	0.58	0.1	0.23	0.17	0.11
EP1: Error de posición en la junta 1								EV2: Error de velocidad en la junta 2								
EP2: Error de posición en la junta 2								EV1: Error de velocidad en la junta 2								

Tabla 3: Indicadores de rendimiento para la trayectoria circular.

Fuente: Propia

De la tabla 3 obtenemos los diferentes valores de los indicadores de desempeño para la trayectoria circular, presentando el mejor desempeño en el controlador adaptativo para el seguimiento de trayectoria referido a la posición, presentando los valores más bajos en IAE, ISE e ITAE con valores de 32.086, 2.21 y 147.63 para la junta 1 y con un tiempo de subida de 0.1s con un Mp igual a 13% siendo el que mejor desempeño presenta, mientras que en la junta 2 se observa que el valor ITAE es ligeramente mayor al controlador por Torque Computado y PD+, lo que supone que al avanzar el tiempo de simulación presenta ligeras variaciones lo que hacen que este error aumente y castigando más el efecto del tiempo, mientras que los controladores anterior mencionados presentan una menor variación con el tiempo avanzado.

Sin embargo, la velocidad se ve afectada al lograr el seguimiento de trayectoria, siendo así que la junta número 1 del control adaptativo presenta los mayores errores cuantificables, siendo los que presentan mejor desempeño el control por Torque Computado y el PD+, de la misma manera el control de Torque computado presenta el mejor seguimiento de la velocidad, con valores de IAE, ITAE e ISE de 40.193, 15.35 y 37.6 respectivamente así como un tiempo de subida de 0.58s y un Mp casi igual a 0.

De estos datos obtenemos que el control adaptativo requiere de una mayor velocidad para lograr un seguimiento de adecuado de la trayectoria lo que significa en mayores variaciones en la velocidad de las juntas, mientras que el control por Torque Computado suaviza estas variaciones de velocidad de una mejor manera, sin embargo, estas suponen que no se logre un seguimiento de trayectoria tan acertado, observándose esto en el tiempo de subida que es casi 5 veces más que el del control adaptativo, lo que significa que requiere de un mayor tiempo lo que le ayuda a la velocidad mas no a la posición.

2.4 TORQUES APLICADOS

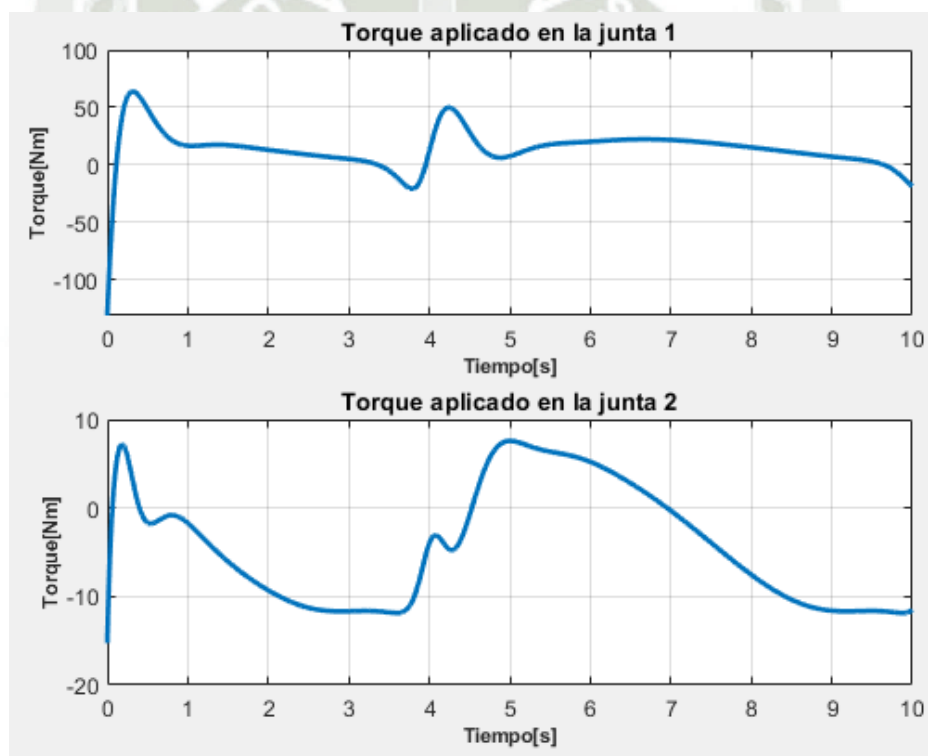


Figura 44. Torque con trayectoria circular con controlador PD

Fuente: Propia

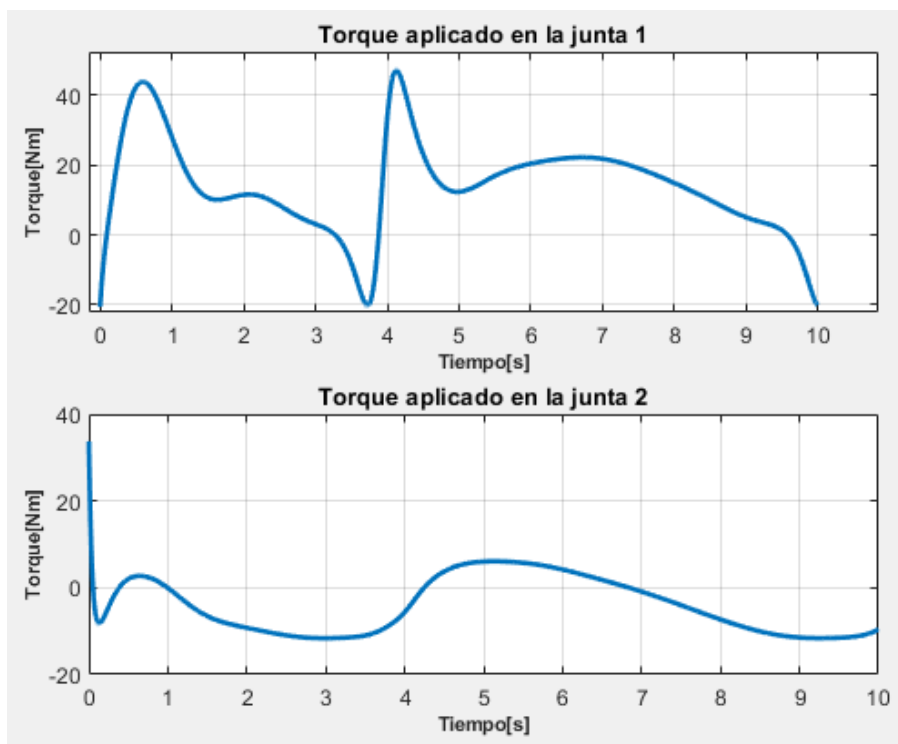


Figura 45. Torque con trayectoria circular con controlador PD +

Fuente: Propia

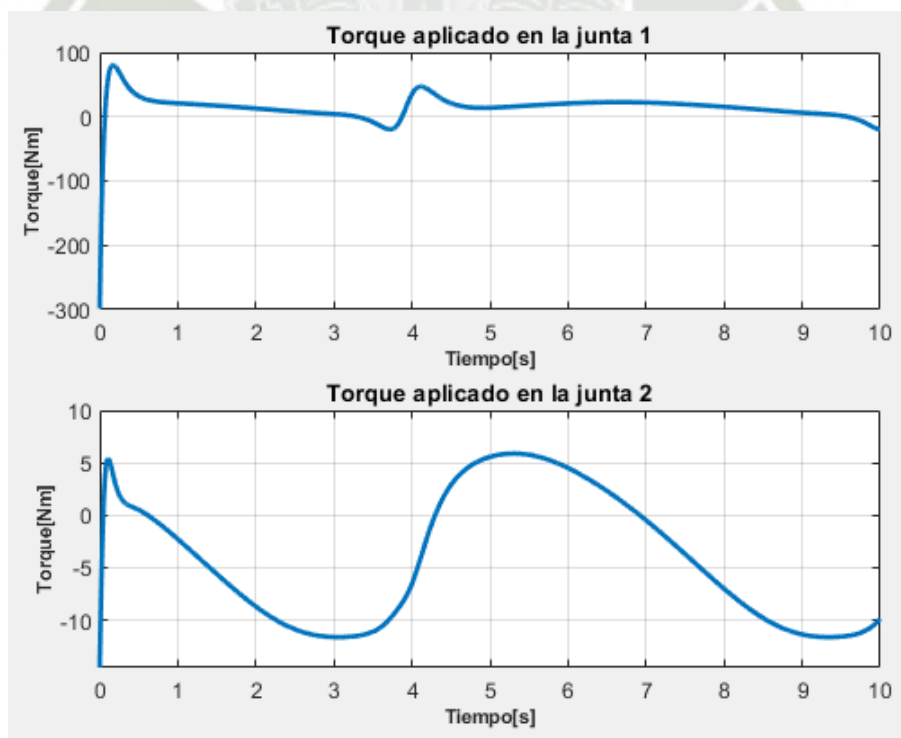


Figura 46. Torque con trayectoria circular con controlador Torque computado

Fuente: Propia

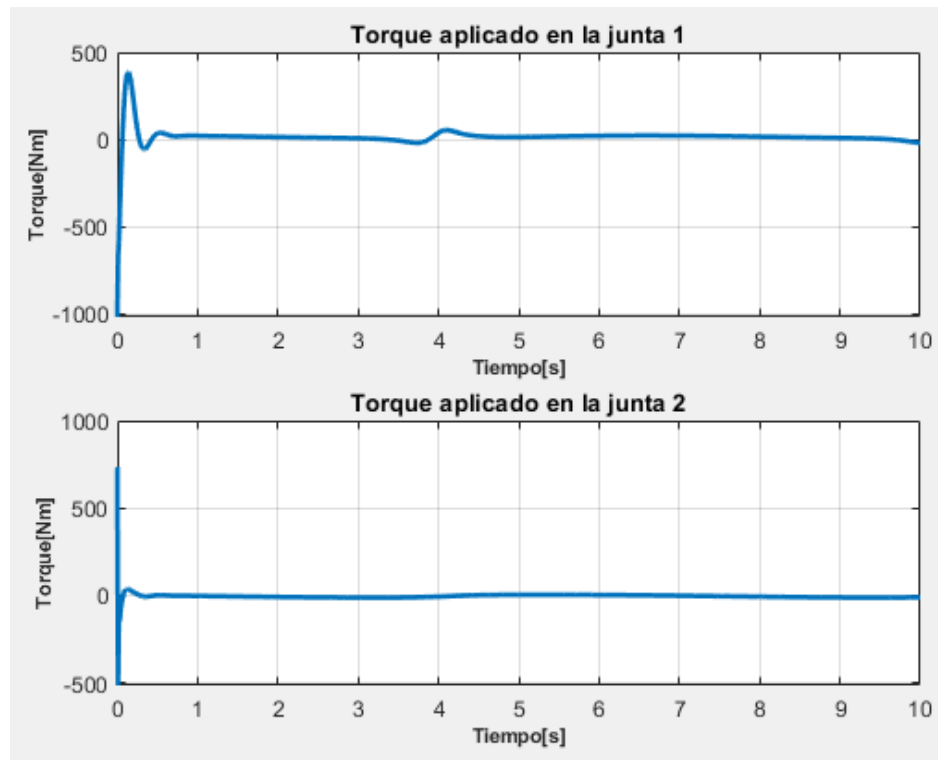


Figura 47. Torque con trayectoria circular con controlador adaptativo

Fuente: Propia

De la Figura 44 a la Figura 47 tenemos los torques aplicados a las juntas para la trayectoria circular, de este podemos observar que el controlador adaptativo es el único que logra aproximar el torque de ambas juntas a 0, sin presentar oscilaciones al comienzo de la trayectoria, sin embargo, este requiere de una gran cantidad de torque al inicio, aproximadamente 1000 Nm de torque para la junta , esto se observa al inicio de las gráficas del error como errores muy grandes al inicio de la trayectoria; lo que nos indica que alcanza un control óptimo de trayectoria en un tiempo pequeño y debido a la dificultad de la figura requiere de muy altos cambios de torque y por ende, de velocidad.

3. TRAYECTORIA LINEAL CON CAMBIO DE DIRECCIÓN

3.1 TRAYECTORIA

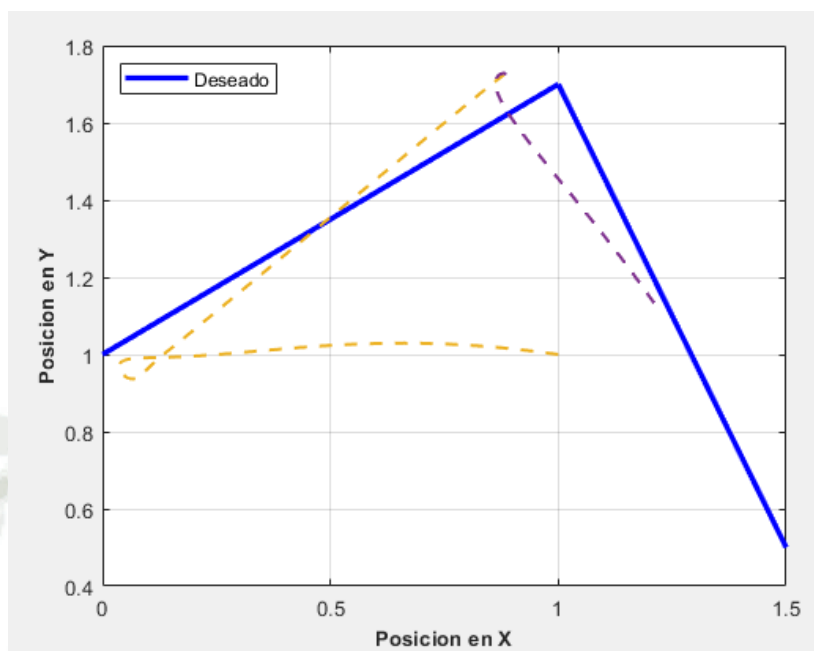


Figura 48. Trayectoria lineal con cambio de dirección con controlador PD

Fuente: Propia

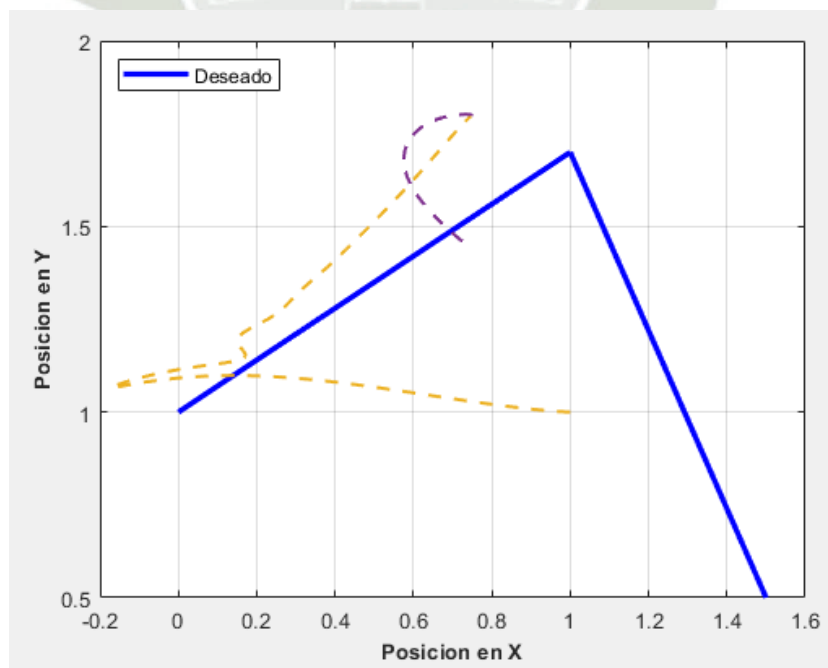


Figura 49. Trayectoria lineal con cambio de dirección con controlador PD aumentado

Fuente: Propia

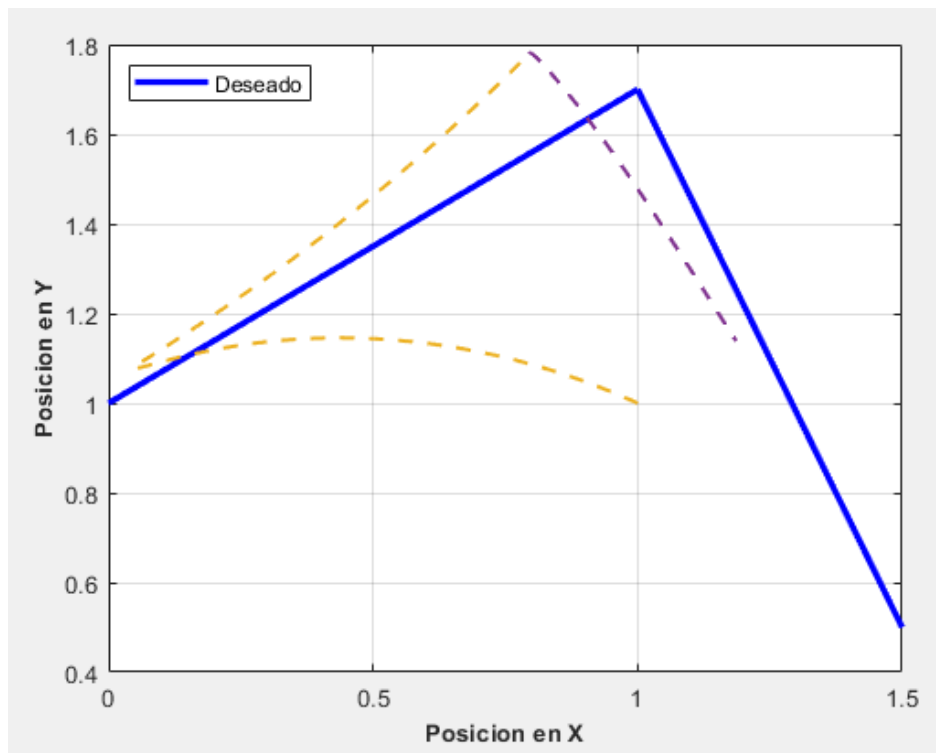


Figura 50. Trayectoria lineal con cambio de dirección con controlador Torque Computado

Fuente: Propia

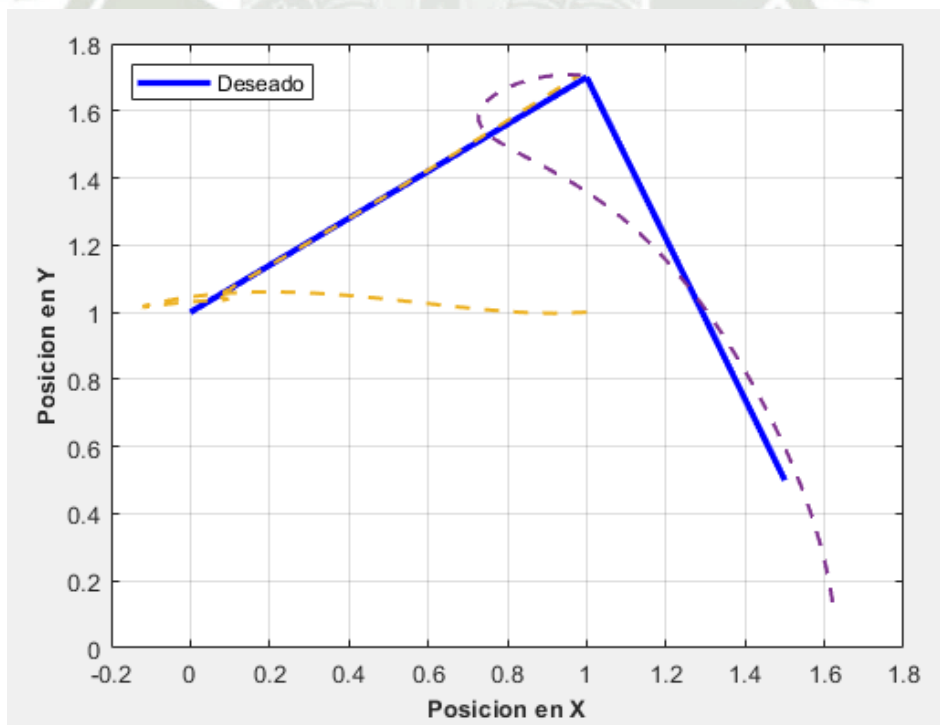


Figura 51. Trayectoria lineal con cambio de dirección con controlador adaptativo

Fuente: Propia

De la Figura 48 a la Figura 51 se presenta el seguimiento de trayectoria con los distintos controladores para una trayectoria lineal con cambio de dirección, presentando esta trayectoria un mayor desafío para los distintos dispositivos, ya que presenta un cambio de dirección y un corto tiempo. Sin embargo, el resultado es un poco diferente a los anteriores, los controladores PD+ y Torque Computado presentan un seguimiento menos acertado que el controlador PD, esto se debe a que al considerar los valores no lineales y el poco tiempo de trabajo significa que no alcanzan la trayectoria.

A diferencia de las anteriores trayectorias, esta tiene un muy corto tiempo para realizarse lo que supone un problema mayor para estos dos controladores ya que si comparamos las anteriores lograban alcanzar un resultado “óptimo” en el seguimiento, pero se lograba por que el tiempo de simulación era mayor y la velocidad no necesitaba realizar un cambio tan brusco. Mientras que el control PD al tener un valor compensador que multiplica el error hace que el cambio sea más rápido consiguiendo un seguimiento mejor que los otros dos controladores, más no un seguimiento óptimo.

Mientras que el controlador adaptativo consigue la seguir la trayectoria en la mayoría de puntos, presentando oscilaciones al inicio y al momento de realizar el cambio de dirección, esto se debe a la velocidad que requiere alcanzar para lograr el seguimiento en el tiempo requerido, lo cual se podrá observar mejor en las figuras siguientes.

3.2 ERRORES DE POSICIÓN

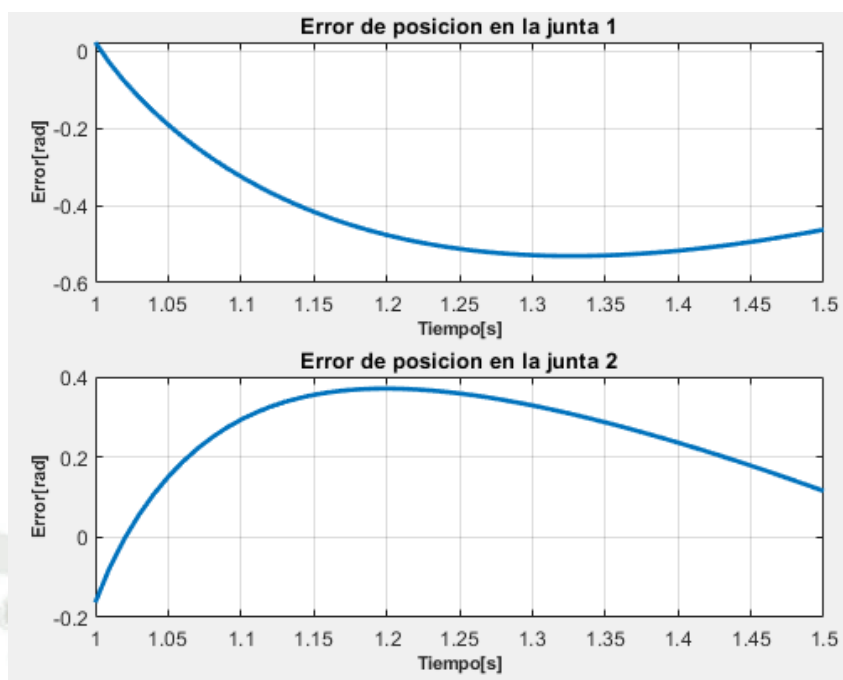


Figura 52. Error de posición en Trayectoria lineal con cambio de dirección con controlador PD

Fuente: Propia

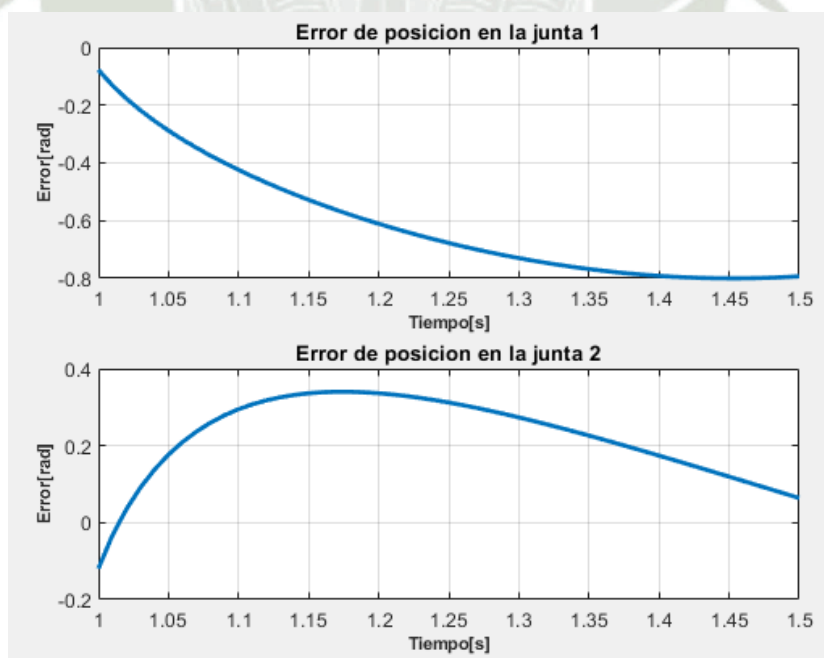


Figura 53. Error de posición en Trayectoria lineal con cambio de dirección con controlador PD aumentado

Fuente: Propia

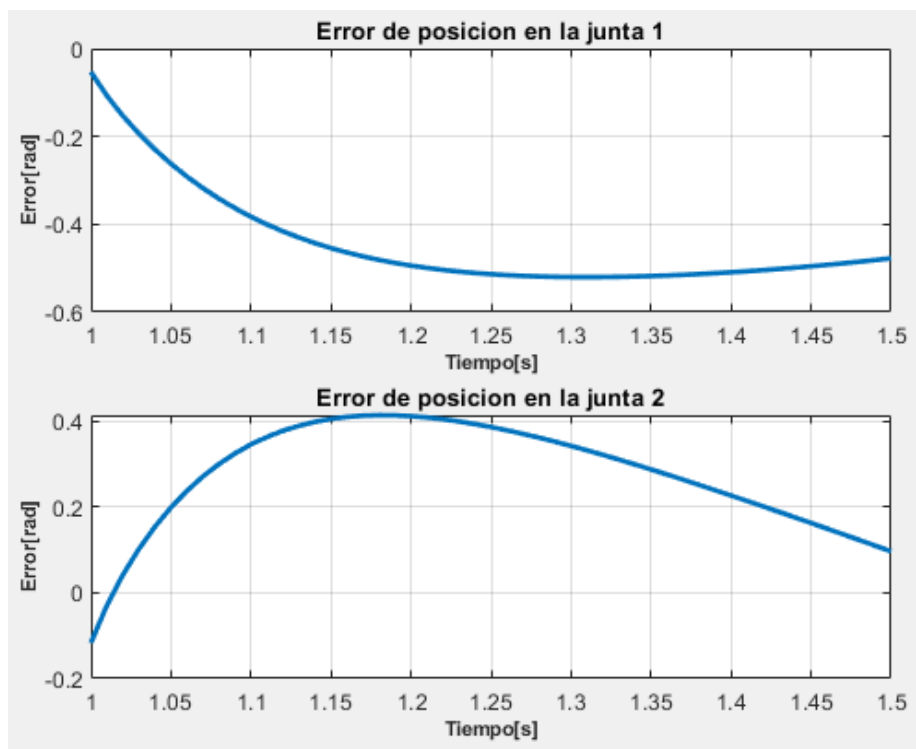


Figura 54. Error de posición en Trayectoria lineal con cambio de dirección con controlador Torque computado

Fuente: Propia

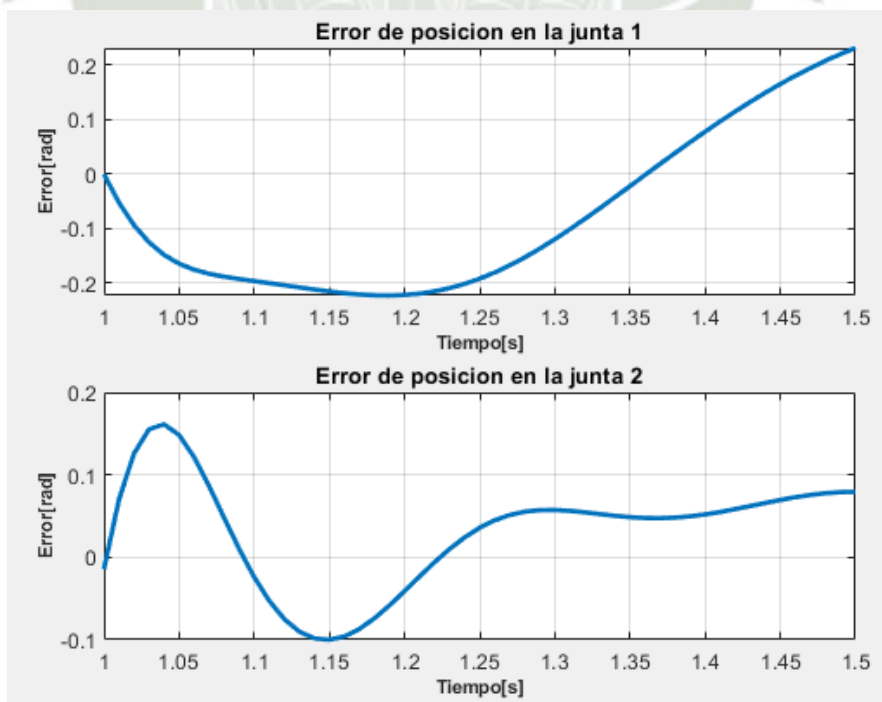


Figura 55. Error de posición en Trayectoria lineal con cambio de dirección con adaptativo

Fuente: Propia

De la Figura 52 a la Figura 55 se presentan los errores de posición para el seguimiento de la trayectoria, debido a la complejidad de la trayectoria a realizar se puede apreciar en las gráficas de posición la dificultad que presentan los controladores para hacer este seguimiento, presentando en todos un error considerable y casi ninguno presenta una tendencia a llegar a 0, sin embargo, el control adaptativo es el que menor valores de error presenta y para el caso de seguimiento en la junta 2 ya se va notando una pequeña tendencia a lograr el 0 pero no se consigue en el tiempo de simulación.

De las gráficas obtenemos un resultado visual que representa así un mejor comportamiento del controlador PD frente al PD+ y al Torque computado, las cuales serán evaluados más adelante.

3.3 ERRORES DE VELOCIDAD

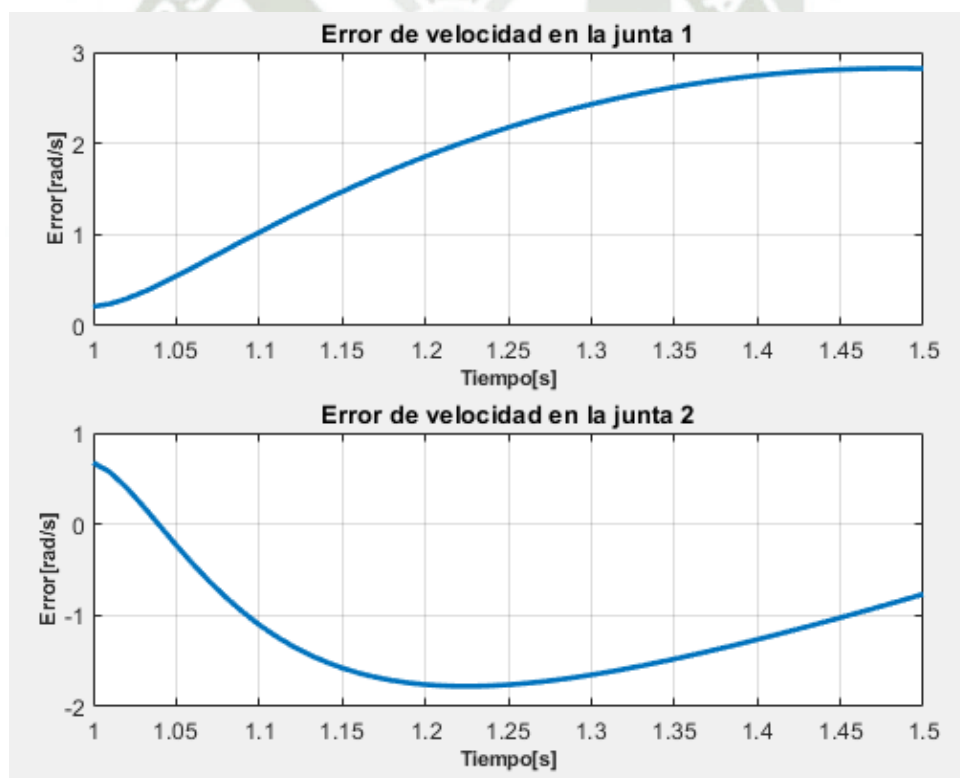


Figura 56. Error de velocidad en Trayectoria lineal con cambio de dirección con controlador PD

Fuente: Propia

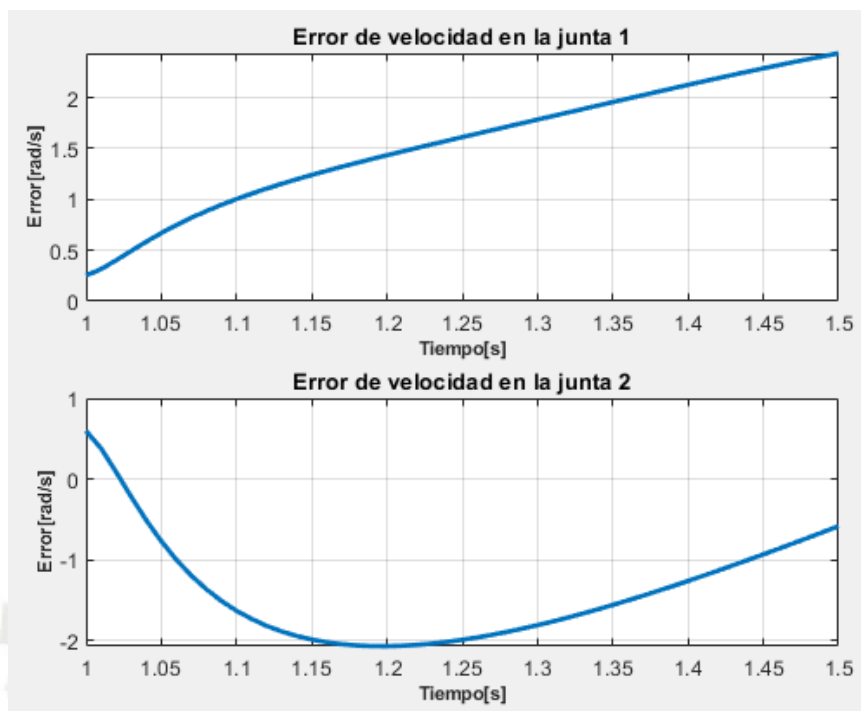


Figura 57. Error de velocidad en Trayectoria lineal con cambio de dirección con controlador PD aumentado

Fuente: Propia

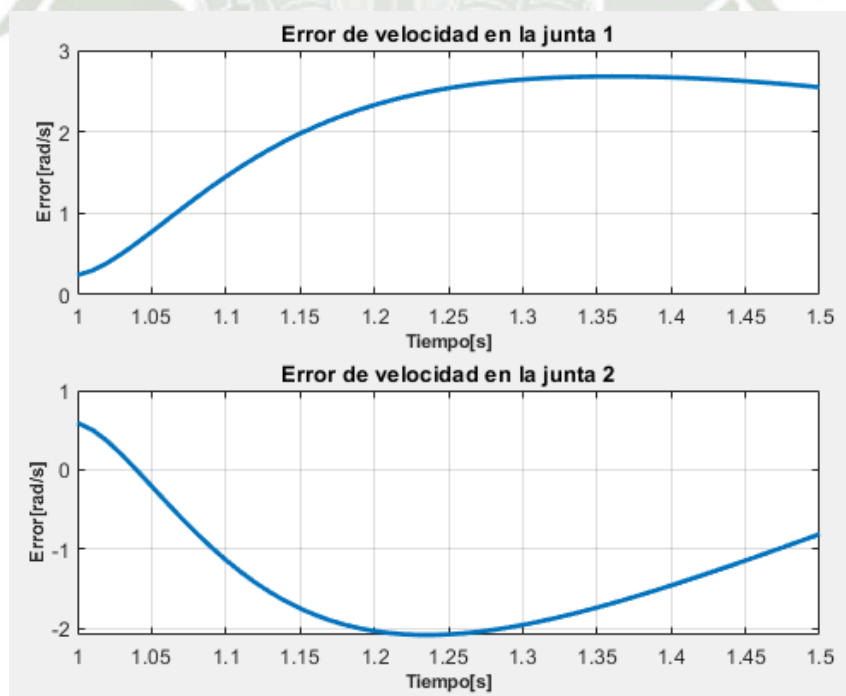


Figura 58. Error de velocidad en Trayectoria lineal con cambio de dirección con controlador Torque computado

Fuente: Propia

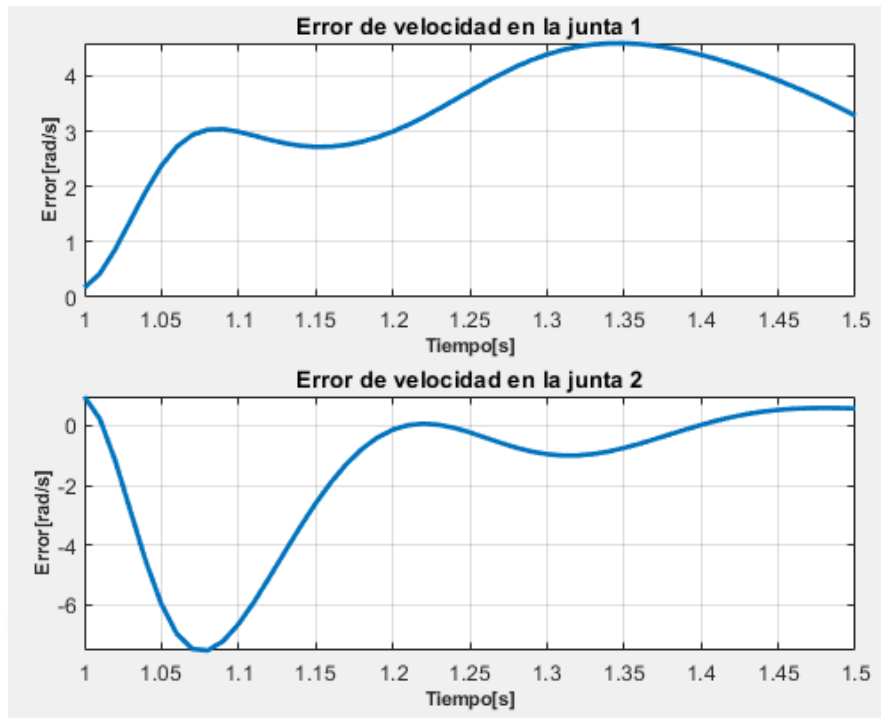


Figura 59. Error de velocidad en Trayectoria lineal con cambio de dirección con controlador adaptativo

Fuente: Propia

De la Figura 56 a la Figura 59 se presentan los errores de velocidad que presentan las juntas, parecidas a las gráficas en la posición e observa un error aún más grande, en los controladores PD, PD+ y Torque computado se observa un parecido, en la junta 1 y 2 una curva suavizada alcanzando valores de hasta 3 y -2 respectivamente, sin conseguir una tendencia a 0.

Sin embargo, se observa que el controlador adaptativo en la junta 1 presenta un error mucho más grande y una cantidad mayor de oscilaciones para la junta 1 y 2, con valores de hasta -7. El comportamiento es parecido con las otras trayectorias, siendo la razón de estas grandes variaciones de velocidad se deben a la gran dificultad de la trayectoria y al corto tiempo a realizarse, requiriendo cambios bruscos que se representan en un mayor error en velocidad, pero un seguimiento más acertado.

TRAYECTORIA RECTA CON CAMBIO DE DIRECCIÓN																
	CONTROL PD				CONTROL PD+				C. TORQUE COMPUTADO				CONTROL ADAPTATIVO			
	EP1	EP2	EV1	EV2	EP1	EP2	EV1	EV2	EP1	EP2	EV1	EV2	EP1	EP2	EV1	EV2
IAE	21.165	13.128	104.29	66.521	30.851	11.596	77.665	73.323	22.466	14.256	106.84	71.548	7.6806	3.3028	172.33	91.909
ITAE	27.197	16.375	135.93	83.789	39.997	14.294	101.49	91.548	28.733	17.662	138.41	90.819	9.52	4.0579	221.48	104.09
ISE	9.6419	3.8306	245.91	98.388	20.787	3.0843	136.28	121.69	10.584	4.6216	252.83	118.11	1.3926	0.2732	638.68	437.35
MP	0.87	1.32	2.76	1.82	0.8	0.34	1.5	2	0.52	0.41	2.67	2.07	0.23	0.16	4.6	7.47
Ts	0.041	0.017	0.047	0.084	0.045	0.051	0.077	0.085	0.03	0.046	0.044	0.092	0.012	0.017	0.02	0.018
EP1: Error de posición en la junta 1								EV2: Error de velocidad en la junta 2								
EP2: Error de posición en la junta 2								EV1: Error de velocidad en la junta 2								

Tabla 4: Indicadores de rendimiento para la trayectoria lineal.

Fuente: Propia

De la tabla 4 podemos observar el valor de los indicadores de desempeño para la trayectoria, observamos el mejor desempeño del controlador adaptativo a pesar de la dificultad de la trayectoria presenta bajos índices de error, siendo los más bajos entre todos los controladores con valores de IAE, ITAE e ISE para la junta 1 y 2 valores de 7.68, 9.52, 1.39, 3.30, 4.057 y 0.2732, los cuales nos indican que realiza un seguimiento de trayectoria adecuado pero no perfecto, siempre presentan oscilaciones al inicio de la trayectoria y al momento de cambio de dirección.

Asimismo, observamos que el tiempo de subida para la junta 1 y 2 es de 0.012s y 0.017s, siendo valores hasta 7 veces menor que el máximo tiempo para el control PD+, dándonos a entender que presenta una alta velocidad de respuesta y un adecuado seguimiento de trayectoria, siendo de esta manera la mejor opción y más efectiva en cuanto a realizar una trayectoria deseada en un tiempo deseado.

De la misma manera observamos que en el seguimiento de posición para este tipo de trayectoria el control PD presenta un gran desempeño frente al control PD+ y Torque computado, presentando índices más bajos que los anteriores mencionados con valores de hasta 12 puntos menos para el caso del ITAE en 27.197, sin embargo, el sobre impulso máximo que alcanza es superior al de los otros dos para la junta 1 y 2, con valores de 87% y 1.32% lo que nos indicaría que presenta oscilaciones al inicio de la trayectoria y al cambiar de dirección mayores a las de los anteriores.

Al centrarnos en la velocidad podemos observar que opuesto al resultado de la posición el controlador adaptativo presenta los mayores índices de error en IAE.ITA.ISE con valores para la junta 1 y 2 de 173.33, 221.48, 638.68, 91.9, 104.1 y 437.35. Siendo el valor de ISE 638 hasta 4.5 veces más que el control PD+ así como su máximo sobre impulso llegando hasta un valor de 747%.

Lo cuál nos indica que estos controladores presentan distintos comportamientos para realizar una trayectoria, el control PD, PD+ y Torque computado no consiguen un control de trayectoria óptimo pero lo reemplazan con no teniendo variaciones tan fuertes en la velocidad y demorándose un poco más. Mientras que el controlador adaptativo “sacrifica” la velocidad para poder llegar a la trayectoria que necesita en el tiempo requerido y esto se aprecia también en el tiempo de subida, siendo el más bajo con tiempos de 0.02s y 0.018s para la junta 1 y 2 respectivamente.

3.4 TORQUES APLICADOS

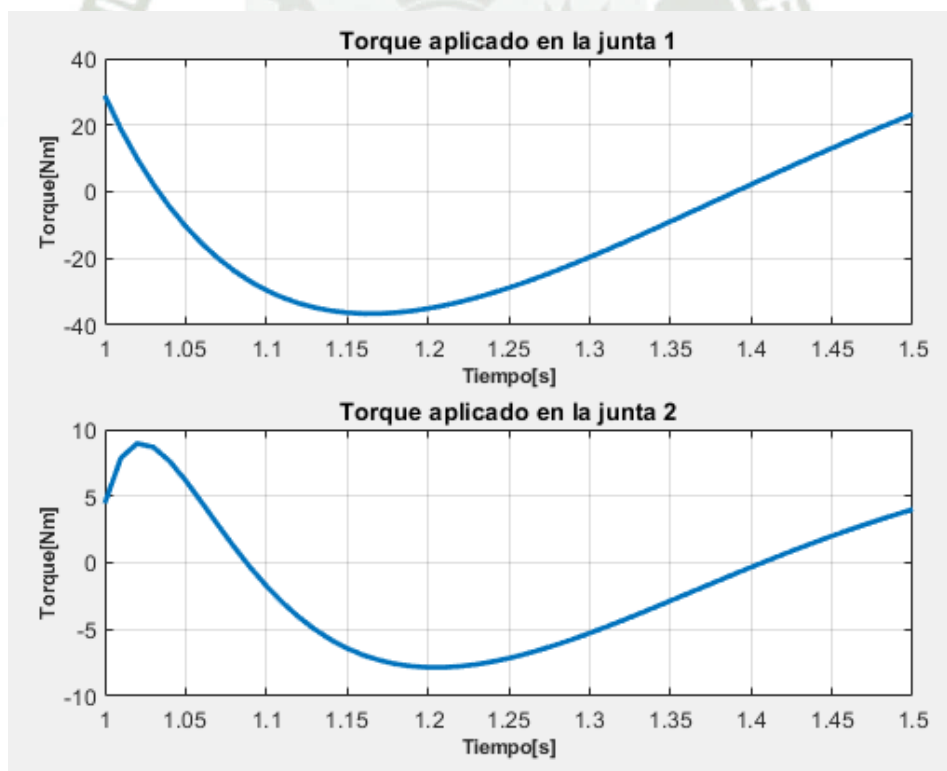


Figura 60. Torque en Trayectoria lineal con cambio de dirección con controlador PD

Fuente: Propia

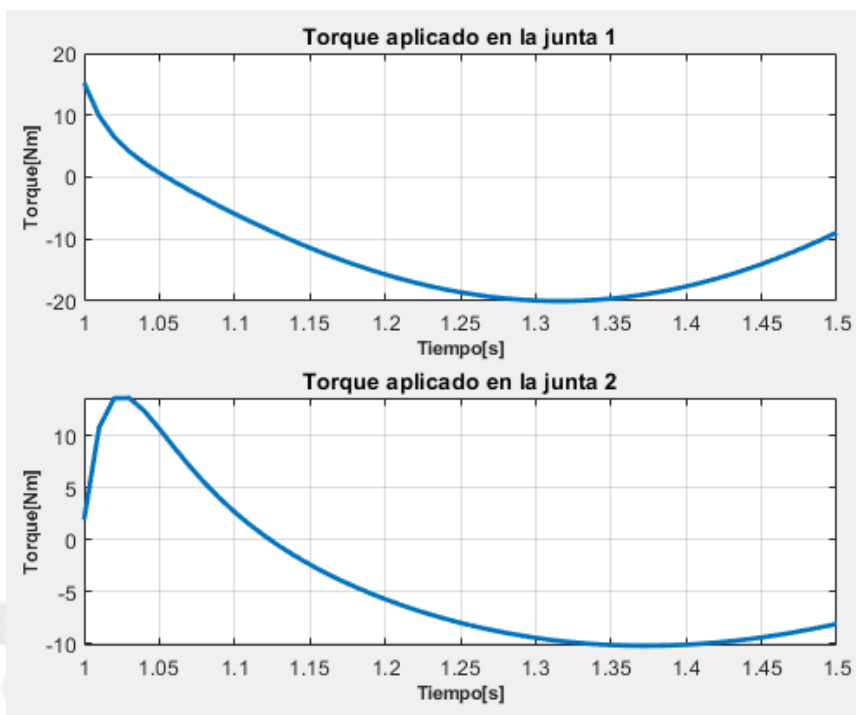


Figura 61. Torque en Trayectoria lineal con cambio de dirección con controlador PD aumentado

Fuente: Propia

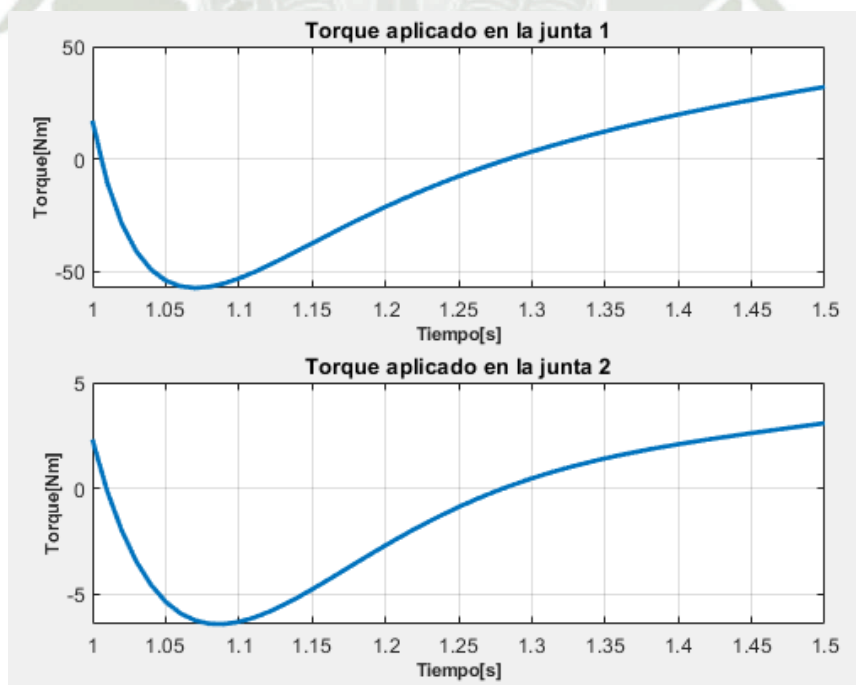


Figura 62. Torque en Trayectoria lineal con cambio de dirección con controlador Torque Computado

Fuente: Propia

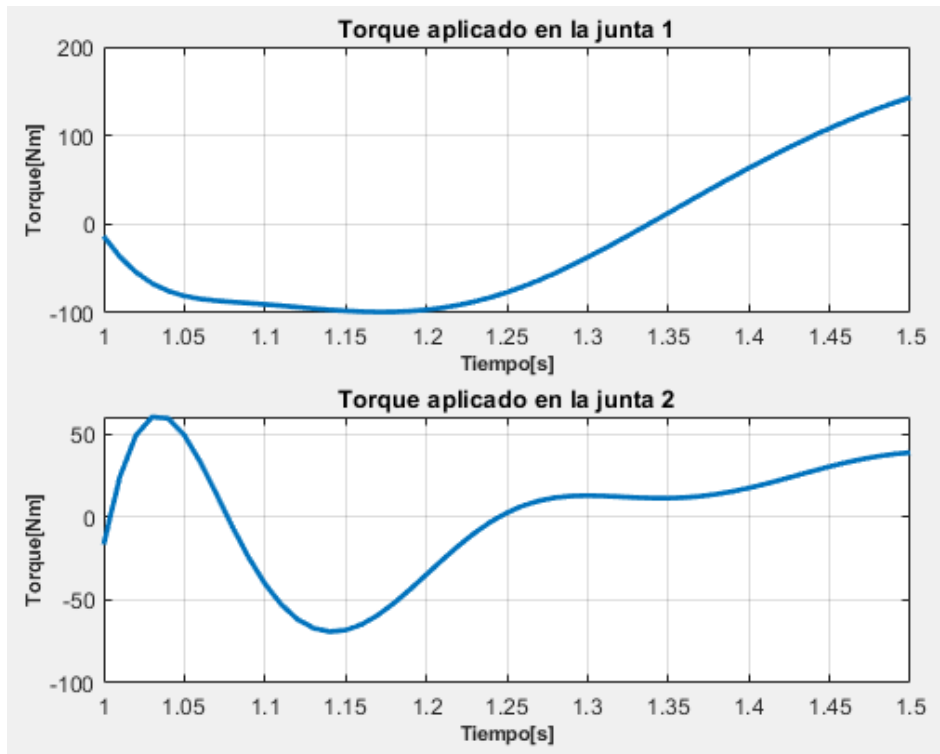


Figura 63. Torque en Trayectoria lineal con cambio de dirección con controlador adaptativo

Fuente: Propia

De la Figura 51 a la Figura 54 se presentan los torques aplicados a las juntas para lograr el seguimiento de trayectoria, para este caso se observa que en ningún momento los controladores consiguen valores estables para el torque en las juntas, observando que el cambio brusco de dirección supone un cambio rápido del torque y su crecimiento a lo largo del tiempo, se observan valores que no alcanzan una tendencia hacia el 0 para los controladores PD, PD+ y Torque computado. Mientras que el controlador adaptativo presenta una pequeña tendencia a buscar el 0 para la junta 2 y de la misma manera que para los casos anteriores observamos valores de torque elevados en comparación a los otros controladores, de hasta 100 Nm, estos valores son debido a la velocidad a la que el controlador tiene que alcanzar la trayectoria y lo logra de una manera óptima mas no perfecta.

DISCUSIÓN

Como se presentó en este trabajo de investigación, se compararon diferentes estrategias de control aplicados a un manipulador robotico de dos grados de libertad para tareas de seguimiento de tres tipos de trayectoria, una trayectoria con una curva suavizada, una trayectoria circular de radio reducido y una trayectoria lineal con un cambio de dirección. Para ello se modeló dinamicamente el manipulador robotico utilizando la formulación de Euler-Lagrange. Utilizando este modelo dinamico se desarrollaron y evaluaron cuatro tipos de estrategias de control: un controlador PD, un controlador PD aumentado, un controlador por torque-computado y finalmente un controlador adaptativo. Todo el sistema robotico y los controladores diseñados fueron simulados utilizando el programa MATLAB. De igual manera los resultados de las simulaciones fueron presentadas con su respectivo analisis y discusion mostrando para ello las diferentes graficas que fueron generadas.

Al comparar estas estrategias de control, el control adaptativo presenta el mejor desempeño para el seguimiento de trayectoria para los 3 casos planteados, siendo este la estrategia de control que presenta los menores índices de error para sus diferentes casos, en algunos casos obteniendo hasta el 4% respecto de los errores de la otras estrategias, esto nos indica que se realizó adecuadamente el diseño del controlador y el resultado frente al seguimiento de trayectoria es el adecuado, contando con bajos índices de error, bajos sobreimpulso y un corto tiempo de subida. Sin embargo, este seguimiento satisfactorio de trayectoria se ve afectado por una gran cantidad de variacion y oscilaciones al inicio de la trayectoria, lo cuál afecta considerablemente en el error de velocidad, mientras más rápido se requiere la acción, más será el error en velocidad y mayor será el requerimiento de torques para lograrlo.

De la misma manera, sin contar al control adaptativo, observamos que el control PD presenta mejor desempeño cuando se requiere bajos tiempos de proceso, claramente observando en las primeras dos trayectorias en las cuales se dio un tiempo de 10s el control PD+ y Torque computado presentan mejores resultados, mientras que en la trayectoria lineal con cambio de dirección el control PD presenta un mejor seguimiento de trayectoria, debido a su mayor velocidad de respuesta y sus características lineales, superando en este caso a los siguientes dos controladores.

Centrándonos en la velocidad podemos observar mientras más complicada sea la trayectoria a realizar y menor sea el tiempo necesario para hacer el trabajo, mayor será la dificultad para controlar la velocidad, viendo que el controlador adaptativo logra conseguir la trayectoria pero

conforme se complica la trayectoria el error de velocidad va aumentando considerablemente al punto de en la trayectoria lineal con cambio de dirección presenta el error de velocidad más grande entre todas las estrategias de control, significando esto que para conseguir el seguimiento adecuado de la posición es necesario unas altas variaciones de velocidad mayormente al iniciar la trayectoria, de lo contrario no sería capaz de obtener los resultados de posición. Esto le supone un mayor error en la velocidad. Mientras que en las diferentes estrategias de control no se consigue un control de trayectoria tan adecuado pero consigue disminuir el error de velocidad, lo que significa que intentan buscar un equilibrio al momento de posición, velocidad y torque, pero esto para un caso en el cuál se requiera de una alta precisión no es lo más adecuado.

Por último al evaluar el torque necesario en cada estrategia de control observamos que para todos los casos en controlador adaptativo necesita de grandes cantidades iniciales de Torque, de hasta 7 veces mayor que el requerido por el control PD, esto nos indica que se requiere altos cambios de dirección y aceleración, efectuar movimientos rápidos es necesario para alcanzar la trayectoria, por ello observamos estas cantidades grandes de torque al iniciar el proceso. Como se mencionó anteriormente el control adaptativo para este caso prioriza el control y el seguimiento de trayectoria ya que utiliza los recursos necesarios para poder alcanzarlo y esto supone entonces de oscilaciones iniciales, debido a los altos niveles de torque y al error de velocidad encontrado. Mientras que las otras estrategias buscan un intermedio entre posición, velocidad y torque.

CONCLUSIONES

1. Esta investigación tuvo como objetivo desarrollar un controlador no lineal adaptativo y compararlo con 3 estrategias de control lineales para el seguimiento de trayectoria de un robot de 2DOF, concluyendo de esta manera que la estrategia de control con mejor desempeño fue el control no lineal adaptativo, presentando en su caso más bajo el 4% del error máximo encontrado en la estrategia PD en la tabla N° 2 con valores de ITAE en la junta 2 para el control PD y adaptativo respectivamente: 2526 y 59 y al compararlo con una estrategia más moderna como el torque computado presenta valores de: 53 y 75 para la junta1 y para la junta 2: 52 y 59; de lo cuál podemos obtener que en un tiempo de simulación más avanzado el control adaptativo presenta pequeñas variaciones en el error que hace que este error aumente. Mientras que al analizar los otros índices de desempeño en el IAE el control adaptativo presenta una notable mejoría con 31 y 24 para ambas juntas siendo seguida solamente por el control de torque computado con valores de 55 y 52.
2. De la misma manera se logró satisfactoriamente el diseño de un robot de 2DOF mediante Euler-lagrange para su modelamiento dinámico y la obtención de sus ecuaciones. Así mismo, se desarrollaron 3 estrategias de control lineal, por medio de sus ecuaciones de control, consideración de parámetros y el uso del modelamiento del robot.
3. Se desarrolló un controlador no lineal adaptativo con las consideraciones de una incertidumbre del 50% en el link 2 del robot, obteniendo un resultado positivo en el seguimiento de trayectoria frente a los demás controladores presentando los menores índices de error con valores de:

Los cuales en comparación con las otras estrategias de control presentan una mejoría notable en el seguimiento de trayectorias disminuyendo considerablemente los índices de desempeño. }

	TRAY. 1		TRAY. 2		TRAY. 3	
	EP1	EP2	EP1	EP2	EP1	EP2
IAE	31.438	24.035	32.086	7.6906	7.6806	3.3028
ITAE	75.169	59.639	147.63	30.889	9.52	4.0579
ISE	15.846	8.4728	2.2173	0.2866	1.3926	0.2732

Tabla 5: Índices de error par el control adaptativo.

Fuente: Propia

4. Se concluye que los controladores típicos que consideran factores no lineales como el PD+ y el Torque Computado obtienen mejores resultados al momento del seguimiento de trayectoria que no requieren de una gran complicación ni un bajo tiempo de trabajo como la presentada en 10s, logrando un resultado adecuado sin muchas oscilaciones iniciales más esto no sucede cuando se requiere una alta velocidad de respuesta, si este es el caso como la presentada con un tiempo de 1.5s, estos controladores disminuyen la variación a costa de no alcanzar el objetivo adecuadamente, mientras que el controlador PD al no contar con factores no lineales y una constante que multiplica el error proporcionalmente y la derivada del error, con figuras lineales es capaz de realizar un seguimiento adecuado y obtener un buen tiempo de respuesta ($t_s = 0.04$ y 0.017) para la junta 1 y 2 respectivamente, que al igual que el control adaptativo requiere de oscilaciones al inicio de la trayectoria y de ser el caso al momento de hacer un cambio de dirección.
5. Finalmente se concluye que el diseño del controlador adaptativo obtuvo los mejores resultados en las 3 trayectorias evaluadas, presentando los menores índices de error y los menores tiempos de respuesta, sin embargo, consiguiendo estos resultados acarreado un error considerable de velocidad según la complejidad del trabajo a realizar y la necesidad de altos niveles de torque, en algunos casos con varoles de hasta 1500 N.m para lo cuál se tendría que dimensionar adecuadamente los actuadores o aplicar limitadores para evitar daños a la estructura del robot. De la misma manera obtenemos el resultado que las diferentes estrategias de control vistas en el presente trabajo se comportan de manera distinta dependiendo del tipo de trabajo a realizar y se deberá escoger la mejor evaluando lo que se necesite para el trabajo, si necesita de alta precisión en posición o en velocidad, del tiempo de respuesta, del sobreimpulso máximo y del adecuado conocimiento del modelo del robot, ya que ante más incertidumbre presente este modelo los resultados de los controladores serán variables.

RECOMENDACIONES Y TRABAJO FUTURO

Aunque los diferentes objetivos se cumplieron, diferentes mejoras pueden ser realizadas, en primer lugar se desarrollaron las diferentes estrategias de control sin realizar un análisis de estabilidad de los mismos, de igual forma, solo se consideró una incerteza paramétrica en uno de los parámetros, un futuro trabajo podría considerar la implementación de estas mejoras.

Se diseñó el controlador en base a un solo parámetro desconocido, se recomienda para trabajos posteriores considerar una cantidad mayor de estos parámetros para poder analizar el desempeño del controlador adaptativo cuando se consideran mayores incertidumbres paramétricas.

Por otro lado, los valores seleccionados de las ganancias fueron obtenidos de la literatura existente, es recomendable realizar una sintonización de las ganancias para observar su influencia en el sistema desarrollado.

Finalmente, se recomienda la implementación real del sistema y analizar las estrategias de control presentadas en este trabajo de investigación. Para realizar la implementación se tiene que considerar necesariamente el hardware y software del sistema, se recomienda un tiempo de respuesta del controlador por encima de los 700 MHz, dedicada al sistema para poder obtener buenos tiempos de respuesta, así mismo, para la selección de actuadores y sensores es necesario tener en cuenta la velocidad de transmisión tanto de entrada como de salida de estos, ya que para obtener los resultados más cercanos a las simulaciones hay que tener tiempos de lectura y respuesta muy rápidos y que los actuadores y sensores puedan responder a esta velocidad.

De la misma manera implementar un limitador de torque para el uso del controlador adaptativo ya que este requiere de valores muy altos para el seguimiento de trayectoria.

REFERENCIAS

- A. Barrientos, L. P. (1997). *Fundamentos de Robótica*. Madrid: McGraw-Hill.
- A. Ishiguro, T. Furuhashi, S. Okuma, Y. Uchikawa. (1992). A neural network compensator for uncertainties of robotics manipulators. *IEEE Transactions on Industrial Electronics*, 565-570.
- Adhikary, N., & Mahanta, C. (2015). Adaptive Backstepping Sliding Mode Controller with PID Sliding Surface for a Co-Ordinated Links (COOL) Robotic Arm. *Proceedings of the 2015 Conference on Advances In Robotics*. Goa, India.
- B. Rahmani, M. Belkheiri. (2016). Robust adaptive control of robotic manipulators using neural networks. *2016 8th International Conference on Modelling, Identification and Control (ICMIC)*, (págs. 839-844). Algiers.
- Bin Ren, Yao Wang, Lilan Liu, X. Tu, Rogelio Lozano. (2018). An Adaptive Robust Control for Trajectory Tracking Of a Robotic Manipulator System. *International Journal of New Technology and Research*.
- Craig, J. J. (2009). *Introduction to robotics: mechanics and control*. Pearson Education India.
- Daniel, A. A. (2017). Evaluación del desempeño de los controladores PID aplicados al control de velocidad y tensión del grupo turbina-generador de la Central Hidroeléctrica Alazán de 6.23MW y elaboración de una propuesta de mejoramiento basada en el diseño de un controlador . Cuenca, Ecuador.
- Deotalu, R., & Chiddarwar, S. (2020). Trajectory Tracking of the Manipulator using Adaptive Computed Torque Control. *2020 IEEE International Conference for Innovation in Technology (INOCON)*, (págs. 1-6).
- Duhart, A. M. (12 de Diciembre de 2014). *DISEÑO, SIMULACIÓN Y CONTROL DE LA DINÁMICA DE UN ROBOT PLANAR DE DOS GRADOS DE LIBERTAD*. Obtenido de Slideshare: <https://es.slideshare.net/bronson.duhart/proyectovirtualsimulacion>
- Fan, L., & Joo, E. M. (2009). Linear and nonlinear PD-type control of robotic manipulators for trajectory tracking. *4th IEEE Conference on Industrial Electronics and Applications*, (págs. 3442-3447).
- Fareh, R., Khadraoui, S., Baziyad, M., & Bettayeb, M. (2020). Synergetic Workspace Tracking Control for 4-DOF Robot Manipulator. *Proceedings of the 2020 12th International Conference on Computer and Automation Engineering*, (págs. 189–193). New York.
- Gonzalez, C. P. (2012). *PROTOTIPO DE UN BRAZO MODELO SCARA CON UN PID*. MEXICO.
- González.V, L. .. (2006). *Control y Robótica*. Obtenido de https://openvrg.moodlehub.com/file.php/30/cyr_0708/archivos/_15/Tema_5.4.htm
- Liu Fan and E. M. Joo. (2009). Linear and nonlinear PD-type control of robotic manipulators for trajectory tracking. *IEEE Conference on Industrial Electronics and Applications*, (págs. 3442-3447). Xi'an.

- Ma, X., Zhao, Y., & Di, Y. (2020). Trajectory Tracking Control of Robot Manipulators Based on U-Model. *Mathematical Problems in Engineering*.
- Nguyen, Kim D., Dankowicz, Harry. (2014). *Robust Adaptive Control of Manipulators*. Illinois.
- Nicosia, S., & Tomei, P. (1991). A PD Control Law for Trajectory Tracking of Flexible Joint Robots. *3rd IFAC Symposium on Robot Control 1991 (SYROCO'91)*, (págs. 357-362). Vienna.
- O. Arrieta, V. Alfaro. (2003). Sintonización de controladores PI y PID utilizando los criterios integrales IAE e ITAE. *Ingeniería*.
- Ogata, K. (1997). *Modern control engineering*. Upper Saddle River. N.J.: Prentice Hall.
- Perez Hidalgo, A. (2018). *RESUMEN DE LAS PRINCIPALES CARACTERISTICAS DE CONTROLADORES PID*.
- Poved, M. (2012). *MODELADO, SIMULACION Y CONTROL DE UN MANIPULADOR ROBOTICO DE 2 GRADOS DE LIBERTAD EMPLEANDO ADAMS/MATLAB*. BUCARAMANGA.
- Rodriguez, F, L. (1996). *Control Adaptativo y Robusto*. Sevilla.
- S. Castaño. (Abril de 2021). *Control automatico educación*. Obtenido de <https://controlautomaticoeducacion.com/control-realimentado/indices-de-desempeno/>
- Saltarén, R, A. (s.f.). *Prácticas de Robótica*.
- Shang, W., Cong, S., Li, Z., & Jiang, S. (2009). Augmented Nonlinear PD Controller for a Redundantly Actuated Parallel Manipulator. *Advanced Robotics*, 1725-1742.
- Siciliano, B. (2009). *Robotics: Modelling, planning and control*. London: Springer.
- Slotine, J. L. (1991). *Applied nonlinear control*. Englewood Cliffs. N.J.: Prentice Hall.
- Song, Z., Yi, J., Zhao, D., & Li, X. (2005). A computed torque controller for uncertain robotic manipulator systems: Fuzzy approach. *Fuzzy Sets and Systems*, 208-226.
- Spong, M. W. (2005). *Robot modeling and control*. John Wiley & Sons.
- W, M. C. (2013). *Diseño e implementación de un controlador adaptativo autosintonizado para el control de un péndulo triple*. Puno.
- Wang, X., & Hou, B. (2018). Trajectory tracking control of a 2-DOF manipulator using computed torque control combined with an implicit lyapunov function method. *Journal of Mechanical Science and Technology*.

ANEXOS**ANEXO 1: Código trayectoria suavizada**

```
close all
clc
clear

% ----- Inicialización de parámetros ----- %

% Constantes generales
g = 9.8;
w = pi/2;
R = 0.75;
t = 0:0.01:10;

tray = zeros(2, length(t));
pos = zeros(2, length(t));
vel = zeros(2, length(t));
tray_d = zeros(2, length(t));
pos_d = zeros(2, length(t));
vel_d = zeros(2, length(t));
entrada_control = zeros(2, length(t));

% Parámetros del modelo dinámico
brazo_robot.m1 = 1;
brazo_robot.m2 = 2;
brazo_robot.l1 = 1;
brazo_robot.l2 = 1;
brazo_robot.I1 = 0.12;
brazo_robot.lc1 = 0.5;
brazo_robot.I2 = 0.25;
brazo_robot.lc2 = 0.6;

% Parámetros reales del robot
m1 = 1;
m2 = 3;
l1 = 1;
l2 = 1;
I1 = 0.12;
lc1 = 0.5;
```

```

I2 = 0.25;
lc2 = 0.6;
m2_a = 2.2;

% Estado inicial
q1 = 0; % Posición de la junta 1
q2 = pi/2; % Posición de la junta 2
dq1 = 0; % Velocidad de la junta 1
dq2 = 0; % Velocidad de la junta 2

% Ganancias
zeta = 0.5;
wn = 5;
kp = wn^2*eye(2);
kd = 2*zeta*wn*eye(2);

% Ganancias del control adaptativo
kd_a = 100*eye(2);
lambda = 10*eye(2);
gamma = 0.005;
a_hat = zeros(1,length(t));

% Matriz de inercia constante de control PD
Mo11 = m1*lc1^2 + I1 + m2*(l1^2 + lc2^2) + I2;
Mo12 = m2*lc2^2 + I2;
Mo22 = m2*lc2^2 + I2;
Mo = [Mo11, Mo12; Mo12, Mo22];

% ----- Secuencia principal ----- %

% Selección del tipo de control
disp("Selección de control:")
disp("1- PD control")
disp("2- PD aumentado")
disp("3- Torque Computado")
disp("4- Control Adaptativo")
ctrl = input('Ingresa el tipo de control (1~4):');
if ctrl == 1
    c="PD";
elseif ctrl == 2

```

```
c="PD aumentado";
elseif ctrl == 3
    c="Torque-computado";
elseif ctrl == 4
    c="Adaptativo";
end

% Selección del tipo de trayectoria
disp("trayectoria selection:")
disp("1- Trayectoria lenta")
disp("2- Trayectoria rapida")
tipo = input('Ingresa el tipo de trayectoria (1~2):');
if tipo == 1
    tr="Trayectoria lenta";
elseif tipo == 2
    tr="Trayectoria rapida";
end

for i=1:length(t)
    q = [q1; q2];
    dq = [dq1; dq2];
    % Posición actual de la extremidad del robot (x,y)
    [x, y] = cinematica_directa(q, brazo_robot);
    % Calculo de la posición deseada de las juntas (q_deseado)
    [q1_d, q2_d] = trayectoria(tipo, t(i), 1);
    q_d = [q1_d; q2_d];
    % Calculo de la velocidad deseada de las juntas (q'_deseado)
    [dq1_d, dq2_d] = trayectoria(tipo, t(i), 2);
    dq_d = [dq1_d; dq2_d];
    % Calculo de la aceleración deseada de las juntas (q''_deseado)
    [ddq1_d, ddq2_d] = trayectoria(tipo, t(i), 3);
    ddq_d = [ddq1_d; ddq2_d];
    % Calculo de la posición deseada de la extremidad del robot
    (x_deseado,y_deseado)
    [xd, yd] = cinematica_directa(q_d, brazo_robot);
    % Calculo del error de posición y velocidad de las juntas
    e = q_d - q;
    de = dq_d - dq;

    if ctrl == 1 % <-----PD
```

```

% Ley de control:
% tau = kd*e'+kp*e
kp = Mo*(wn^2*eye(2));
kd = Mo*(2*zeta*wn*eye(2));
tau = kd*de+kp*e;

% Calculo de las matrices del modelo dinamico del robot
M = matriz_inercia(brazo_robot, q);
C = matriz_fuerzas(brazo_robot, q, dq);
g = matriz_gravitacional(brazo_robot, q);

% Ecuación en lazo cerrado:
% q'' = M(q)^-1[tau-C(q,q')*q'-g(q)]
ddq = M^-1*(tau-C*dq-g);

elseif ctrl == 2 % <-----PD aumentado
% Calculo de las matrices del modelo dinamico del robot
M = matriz_inercia(brazo_robot, q);
C = matriz_fuerzas(brazo_robot, q, dq);
g = matriz_gravitacional(brazo_robot, q);

% Ley de control:
% tau = M(q)*qd''+C(q,q')*qd'+g(q)+kd*e'+kp*e;
tau = M*ddq_d+C*dq_d+g+kd*de+kp*e;

% Ecuación en lazo cerrado:
% q'' = M(q)^-1[M(q)*qd''+[C(q,q')+kd]*e'+kp*e]
ddq = M^-1*(M*ddq_d+(C+kd)*de+kp*e);

elseif ctrl == 3 % <-----Torque computado
% Calculo de las matrices del modelo dinamico del robot
M = matriz_inercia(brazo_robot, q);
C = matriz_fuerzas(brazo_robot, q, dq);
g = matriz_gravitacional(brazo_robot, q);

% Ley de control:
% tau = M(q)*[qd''+kd*e'+kp*e]+C(q,q')*q+g(q);
tau = M*(ddq_d+kd*de+kp*e)+C*dq+g;

% Ecuación en lazo cerrado:
% q'' = qd''+kd*e'+kp*e
ddq = ddq_d+kd*de+kp*e;

elseif ctrl == 4 % <-----Adaptive control
% Calculo de las matrices del modelo dinamico del robot
M = matriz_inercia(brazo_robot, q);

```

```

C = matriz_fuerzas(brazo_robot, q, dq);
G = matriz_gravitacional(brazo_robot, q);
%     Calculo de qr y qr'
dqr = dq_d + lambda*(q_d-q);
ddqr = ddq_d + lambda*(dq_d-q);
%     Calculo de sigma
sigma = dqr-dq;
%     Parametrización del modelo dinamico
yy1 =
(brazo_robot.l1^2+brazo_robot.lc2^2+2*brazo_robot.l1*cos(q(2)))*ddqr(1)+(br
azo_robot.l1*brazo_robot.lc2*cos(q(2)))+...

brazo_robot.lc2^2)*ddqr(2)+(brazo_robot.l1*brazo_robot.lc2*sin(q(2)))*dqr(1
)-...

(brazo_robot.l1*brazo_robot.lc2*sin(q(2)))*(dq(1)+dq(2))*dqr(2)+...
    g*(brazo_robot.lc2*cos(q(1)+q(2))+brazo_robot.l1*cos(q(1)));
yy2 =
(brazo_robot.l1*brazo_robot.lc2*cos(q(2))+brazo_robot.lc2^2)*ddqr(1)+brazo_
robot.lc2^2*ddqr(2)+...

(brazo_robot.l1*brazo_robot.lc2*sin(q(2))*dq(1))*dqr(1)+brazo_robot.lc2*g*c
os(q(1)+q(2));
yy = [yy1; yy2];
%     Calculo de la ley de adaptación
da_hat = gamma*yy'*sigma;
a_hat(i) = a_hat(i)+0.01*da_hat;
%     Calculo de matrices que no dependen del parámetro incierto
Mo11 =
brazo_robot.m1*brazo_robot.lc1^2+brazo_robot.I1+brazo_robot.I2;
Mo12 = brazo_robot.I2;
Mo22 = brazo_robot.I2;
Mo = [Mo11 Mo12; Mo12 Mo22];

go1 = brazo_robot.m1*brazo_robot.lc1*g*cos(q(1));
go = [go1;0];
%     Ley de control:
%     tau = Y'*a'+Mo(q)qr'+Co(q,q')qr'+go(q)+Kd*sigma
tau = yy*a_hat + Mo*ddqr + go + kd_a*sigma;
%     Ecuación en lazo cerrado:
%     q'' = Mq^-1[tau-C(q,q')q'-g(q)]

```

```
    ddq = M^-1*(tau-C*dq-G);
end

% Realizar integración
dq1 = dq1 + 0.01*ddq(1);
dq2 = dq2 + 0.01*ddq(2);
q1 = q1 + 0.01*dq1;
q2 = q2 + 0.01*dq2;
% Almacenar valores
tray(1,i) = x;
tray(2,i) = y;
tray_d(1,i) = xd;
tray_d(2,i) = yd;
pos(1,i) = q1;
pos(2,i) = q2;
pos_d(1,i) = q1_d;
pos_d(2,i) = q2_d;
vel(1,i) = dq1;
vel(2,i) = dq2;
vel_d(1,i) = dq1_d;
vel_d(2,i) = dq2_d;
entrada_control(1,i) = tau(1);
entrada_control(2,i) = tau(2);
end

fig1=figure;
plot(tray_d(1,:),tray_d(2:,:), 'LineWidth',2.5)
grid on
hold on
plot(tray(1,:),tray(2:,:), '--', 'LineWidth',1.5)
legend('Deseado', 'Real', 'Location', 'NorthWest')
xlabel('Posicion en X', 'FontSize',10, 'FontWeight', 'bold')
ylabel('Posicion en Y', 'FontSize',10, 'FontWeight', 'bold')

fig2=figure;
subplot(2,1,1)
plot(t,pos_d(1,:),t,pos(1:,:), '--', 'LineWidth',2)
title('Posicion en la junta 1')
xlabel('Tiempo[s]', 'FontSize',8, 'FontWeight', 'bold')
ylabel('[Rad]', 'FontSize',8, 'FontWeight', 'bold')
```

```
legend('Desired','Real','Location','southeast')
grid on
subplot(2,1,2)
plot(t,pos_d(2,:),t,pos(2:),'--','LineWidth',2)
title('Posicion en la junta 2')
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('[Rad]','FontSize',8,'FontWeight','bold')
legend('Deseado','Real','Location','southeast')
grid on
%
fig3=figure;
subplot(2,1,1)
plot(t,pos_d(1:)-pos(1:),'LineWidth',2)
title('Error de posicion en la junta 1')
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('Error[rad]','FontSize',8,'FontWeight','bold')
grid on
subplot(2,1,2)
plot(t,pos_d(2:)-pos(2:),'LineWidth',2)
title('Error de posicion en la junta 2')
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('Error[rad]','FontSize',8,'FontWeight','bold')
grid on

fig4=figure;
subplot(2,1,1)
plot(t,vel_d(1:),t,vel(1:),'--','LineWidth',2)
title('Velocidad en la junta 1')
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('[Rad/s]','FontSize',8,'FontWeight','bold')
legend('Deseado','Real','Location','southeast')
grid on
subplot(2,1,2)
plot(t,vel_d(2:),t,vel(2:),'--','LineWidth',2)
title('Velocidad en la junta 2')
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('[Rad/s]','FontSize',8,'FontWeight','bold')
legend('Deseado','Real','Location','southeast')
grid on
%
fig5=figure;
```

```
subplot(2,1,1)
plot(t,vel_d(1,:)-vel(1,:), 'LineWidth',2)
xlabel('Tiempo[s]', 'FontSize',8, 'FontWeight', 'bold')
ylabel('Error[rad/s]', 'FontSize',8, 'FontWeight', 'bold')
title('Error de velocidad en la junta 1')
grid on
subplot(2,1,2)
plot(t,vel_d(2,:)-vel(2,:), 'LineWidth',2)
xlabel('Tiempo[s]', 'FontSize',8, 'FontWeight', 'bold')
ylabel('Error[rad/s]', 'FontSize',8, 'FontWeight', 'bold')
title('Error de velocidad en la junta 2')
grid on

fig6=figure;
subplot(2,1,1)
plot(t,entrada_control(1,:), 'LineWidth',2)
xlabel('Tiempo[s]', 'FontSize',8, 'FontWeight', 'bold')
ylabel('Torque[Nm]', 'FontSize',8, 'FontWeight', 'bold')
title('Torque aplicado en la junta 1')
grid on
subplot(2,1,2)
plot(t,entrada_control(2,:), 'LineWidth',2)
xlabel('Tiempo[s]', 'FontSize',8, 'FontWeight', 'bold')
ylabel('Torque[Nm]', 'FontSize',8, 'FontWeight', 'bold')
title('Torque aplicado en la junta 2')
grid on
```

ANEXO 2: Código trayectoria circular

```
close all
clc
clear
t = 0:0.01:10;
w = 1;
xf=0.5+0.5*cos(t*w);
yf=0.5+0.5*sin(t*w);

% ----- Inicialización de parámetros ----- %

% Constantes generales
g = 9.8;
tray = zeros(2, length(t));
pos = zeros(2, length(t));
vel = zeros(2, length(t));
tray_d = zeros(2, length(t));
pos_d = zeros(2, length(t));
vel_d = zeros(2, length(t));
entrada_control = zeros(2, length(t));

% Parámetros del modelo dinámico
brazo_robot.m1 = 1;
brazo_robot.m2 = 2;
brazo_robot.l1 = 1;
brazo_robot.l2 = 1;
brazo_robot.I1 = 0.12;
brazo_robot.lc1 = 0.5;
brazo_robot.I2 = 0.25;
brazo_robot.lc2 = 0.6;

% Parámetros reales del robot
m1 = 1;
m2 = 3;
l1 = 1;
l2 = 1;
I1 = 0.12;
lc1 = 0.5;
I2 = 0.25;
```

```

lc2 = 0.6;
m2_a = 2.2;

% Estado inicial
q1 = 0; % Posición de la junta 1
q2 = pi/2; % Posición de la junta 2
dq1 = 0; % Velocidad de la junta 1
dq2 = 0; % Velocidad de la junta 2

% Ganancias
zeta = 1;
wn = 10;
kp = wn^2*eye(2);
kd = 2*zeta*wn*eye(2);

% Ganancias del control adaptativo
kd_a = 100*eye(2);
lambda = 20*eye(2);
gamma = 0.0005;
a_hat = zeros(1,length(t));

% Matriz de inercia constante de control PD
Mo11 = m1*lc1^2 + I1 + m2*(l1^2 + lc2^2) + I2;
Mo12 = m2*lc2^2 + I2;
Mo22 = m2*lc2^2 + I2;
Mo = [Mo11, Mo12; Mo12, Mo22];

% ----- Secuencia principal ----- %

% Selección del tipo de control
disp("Selección de control:")
disp("1- PD control")
disp("2- PD aumentado")
disp("3- Torque Computado")
disp("4- Control Adaptativo")
ctrl = input('Ingresa el tipo de control (1~4):');
if ctrl == 1
    c="PD";
elseif ctrl == 2
    c="PD aumentado";

```

```

elseif ctrl == 3
    c="Torque-computado";
elseif ctrl == 4
    c="Adaptativo";
end

%CINEMÁTICA INVERSA
for i=1:length(t)

D(i)=((xf(i))^2+(yf(i))^2-l2^2-l1^2)/(2*l2*l1);
q2n(i)= atan2(+sqrt(1-D(i)^2),D(i));
q1n(i)= atan2(yf(i),xf(i))-atan2(l2*sin(q2n(i)),l1+l2*cos(q2n(i)));
q1g=q1n*180/pi;
q2g=q2n*180/pi;
q2d(i)=-((cos(t(i))*(sin(t(i))/2 + 1/2))/2 - (sin(t(i))*(cos(t(i))/2 +
1/2))/2)/(1 - ((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 -
1)^2)^(1/2);
q1d(i)=((sin(t(i))*(cos(t(i))/2 + 1/2))/2 - (cos(t(i))*(sin(t(i))/2 +
1/2))/2 + (((cos(t(i))*(sin(t(i))/2 + 1/2))/2 - (sin(t(i))*(cos(t(i))/2 +
1/2))/2)*((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 - 1))/(1 -
((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 -
1)^2)^(1/2))/(((sin(t(i))/2 + 1/2)^2/2 + (1 - ((sin(t(i))/2 + 1/2)^2/2 +
(cos(t(i))/2 + 1/2)^2/2 - 1)^2)^(1/2) + (cos(t(i))/2 + 1/2)^2/2 - 1)^2 + 1)
+ (cos(t(i))/(2*(cos(t(i))/2 + 1/2)) + (sin(t(i))*(sin(t(i))/2 +
1/2))/(2*(cos(t(i))/2 + 1/2)^2))/((sin(t(i))/2 + 1/2)^2/(cos(t(i))/2 +
1/2)^2 + 1);
q2dd(i)=- (cos(t(i))^2/4 + sin(t(i))^2/4 - (cos(t(i))*(cos(t(i))/2 +
1/2))/2 - (sin(t(i))*(sin(t(i))/2 + 1/2))/2)/(1 - ((sin(t(i))/2 + 1/2)^2/2
+ (cos(t(i))/2 + 1/2)^2/2 - 1)^2)^(1/2) - (((cos(t(i))*(sin(t(i))/2 +
1/2))/2 - (sin(t(i))*(cos(t(i))/2 + 1/2))/2)^2*((sin(t(i))/2 + 1/2)^2/2 +
(cos(t(i))/2 + 1/2)^2/2 - 1))/(1 - ((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2
+ 1/2)^2/2 - 1)^2)^(3/2);
q1dd(i)=((cos(t(i))*sin(t(i)))/(2*(cos(t(i))/2 + 1/2)^2) -
sin(t(i))/(2*(cos(t(i))/2 + 1/2)) + (sin(t(i))^2*(sin(t(i))/2 +
1/2))/(2*(cos(t(i))/2 + 1/2)^3) + (cos(t(i))*(sin(t(i))/2 +
1/2))/(2*(cos(t(i))/2 + 1/2)^2))/((sin(t(i))/2 + 1/2)^2/(cos(t(i))/2 +
1/2)^2 + 1) + (((cos(t(i))*(sin(t(i))/2 + 1/2))/2 - (sin(t(i))*(cos(t(i))/2
+ 1/2))/2)^2/(1 - ((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 -
1)^2)^(1/2) - sin(t(i))^2/4 - cos(t(i))^2/4 + (cos(t(i))*(cos(t(i))/2 +
1/2))/2 + (sin(t(i))*(sin(t(i))/2 + 1/2))/2 + (((sin(t(i))/2 + 1/2)^2/2 +

```

```

(cos(t(i))/2 + 1/2)^2/2 - 1)*(cos(t(i))^2/4 + sin(t(i))^2/4 -
(cos(t(i))*(cos(t(i))/2 + 1/2))/2 - (sin(t(i))*(sin(t(i))/2 + 1/2))/2))/(1
- ((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 - 1)^2)^(1/2) +
(((cos(t(i))*(sin(t(i))/2 + 1/2))/2 - (sin(t(i))*(cos(t(i))/2 +
1/2))/2)^2*((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 - 1)^2)/(1 -
((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 -
1)^2)^(3/2))/(((sin(t(i))/2 + 1/2)^2/2 + (1 - ((sin(t(i))/2 + 1/2)^2/2 +
(cos(t(i))/2 + 1/2)^2/2 - 1)^2)^(1/2) + (cos(t(i))/2 + 1/2)^2/2 - 1)^2 + 1)
- ((cos(t(i))/(2*(cos(t(i))/2 + 1/2)) + (sin(t(i))*(sin(t(i))/2 +
1/2))/(2*(cos(t(i))/2 + 1/2)^2))*((sin(t(i))*(sin(t(i))/2 +
1/2)^2)/(cos(t(i))/2 + 1/2)^3 + (cos(t(i))*(sin(t(i))/2 +
1/2))/(cos(t(i))/2 + 1/2)^2))/((sin(t(i))/2 + 1/2)^2/(cos(t(i))/2 + 1/2)^2
+ 1)^2 + (2*((sin(t(i))*(cos(t(i))/2 + 1/2))/2 - (cos(t(i))*(sin(t(i))/2 +
1/2))/2 + (((cos(t(i))*(sin(t(i))/2 + 1/2))/2 - (sin(t(i))*(cos(t(i))/2 +
1/2))/2)*((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 - 1)))/(1 -
((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 -
1)^2)^(1/2))^2*((sin(t(i))/2 + 1/2)^2/2 + (1 - ((sin(t(i))/2 + 1/2)^2/2 +
(cos(t(i))/2 + 1/2)^2/2 - 1)^2)^(1/2) + (cos(t(i))/2 + 1/2)^2/2 -
1))/(((sin(t(i))/2 + 1/2)^2/2 + (1 - ((sin(t(i))/2 + 1/2)^2/2 +
(cos(t(i))/2 + 1/2)^2/2 - 1)^2)^(1/2) + (cos(t(i))/2 + 1/2)^2/2 - 1)^2 +
1)^2;
xa(i) = l1*cos(q1n(i))+l2*cos(q1n(i)+q2n(i));
ya(i) = l1*sin(q1n(i))+l2*sin(q1n(i)+q2n(i));
end

```

```

for i=1:length(t)
    q = [q1; q2];
    dq = [dq1; dq2];

    % Posición actual de la extremidad del robot (x,y)
    [x, y] = cinematica_directa(q, brazo_robot);
    % Calculo de la posición deseada de las juntas (q_deseado)
    % [q1_d, q2_d] = trayectorial(tipo, t(i), 1);
    q_d = [q1n(i); q2n(i)];
    % Calculo de la velocidad deseada de las juntas (q'_deseado)
    % [dq1_d, dq2_d] = trayectorial(tipo, t(i), 2);
    dq_d = [q1d(i); q2d(i)];
    % Calculo de la aceleración deseada de las juntas (q''_deseado)
    % [ddq1_d, ddq2_d] = trayectorial(tipo, t(i), 3);

```

```

ddq_d = [q1dd(i); q2dd(i)];
%   Calculo de la posición deseada de la extremidad del robot
(x_deseado,y_deseado)
[xd, yd] = cinematica_directa(q_d, brazo_robot);
%   Calculo del error de posición y velocidad de las juntas
e = q_d - q;
de = dq_d - dq;

if ctrl == 1 % <-----PD
%   Ley de control:
%   tau = kd*e'+kp*e
kp = Mo*(wn^2*eye(2));
kd = Mo*(2*zeta*wn*eye(2));
tau = kd*de+kp*e;
%   Calculo de las matrices del modelo dinamico del robot
M = matriz_inercia(brazo_robot, q);
C = matriz_fuerzas(brazo_robot, q, dq);
g = matriz_gravitacional(brazo_robot, q);
%   Ecuación en lazo cerrado:
%   q'' = M(q)^-1[tau-C(q,q')*q'-g(q)]
ddq = M^-1*(tau-C*dq-g);

elseif ctrl == 2 % <-----PD aumentado
%   Calculo de las matrices del modelo dinamico del robot
M = matriz_inercia(brazo_robot, q);
C = matriz_fuerzas(brazo_robot, q, dq);
g = matriz_gravitacional(brazo_robot, q);
%   Ley de control:
%   tau = M(q)*qd''+C(q,q')*qd'+g(q)+kd*e'+kp*e;
tau = M*ddq_d+C*dq_d+g+kd*de+kp*e;
%   Ecuación en lazo cerrado:
%   q'' = M(q)^-1[M(q)*qd''+[C(q,q')+kd]*e'+kp*e]
ddq = M^-1*(M*ddq_d+(C+kd)*de+kp*e);

elseif ctrl == 3 % <-----Torque computado
%   Calculo de las matrices del modelo dinamico del robot
M = matriz_inercia(brazo_robot, q);
C = matriz_fuerzas(brazo_robot, q, dq);
g = matriz_gravitacional(brazo_robot, q);
%   Ley de control:

```

```

%      tau = M(q)*[qd''+kd*e'+kp*e]+C(q,q')*q+g(q);
tau = M*(ddq_d+kd*de+kp*e)+C*dq+g;

%      Ecuación en lazo cerrado:
%      q'' = qd''+kd*e'+kp*e
ddq = ddq_d+kd*de+kp*e;

elseif ctrl == 4 % <-----Control adaptativo
%      Calculo de las matrices del modelo dinamico del robot
M = matriz_inercia(brazo_robot, q);
C = matriz_fuerzas(brazo_robot, q, dq);
G = matriz_gravitacional(brazo_robot, q);

%      Calculo de qr y qr'
dqr = dq_d + lambda*(q_d-q);
ddqr = ddq_d + lambda*(dq_d-q);

%      Calculo de sigma
sigma = dqr-dq;

%      Parametrización del modelo dinamico
yy1 =
(brazo_robot.l1^2+brazo_robot.lc2^2+2*brazo_robot.l1*cos(q(2)))*ddqr(1)+(br
azo_robot.l1*brazo_robot.lc2*cos(q(2)))+...

brazo_robot.lc2^2)*ddqr(2)+(brazo_robot.l1*brazo_robot.lc2*sin(q(2)))*dqr(1
)-...

(brazo_robot.l1*brazo_robot.lc2*sin(q(2))*(dq(1)+dq(2)))*dqr(2)+...
g*(brazo_robot.lc2*cos(q(1)+q(2))+brazo_robot.l1*cos(q(1)));
yy2 =
(brazo_robot.l1*brazo_robot.lc2*cos(q(2))+brazo_robot.lc2^2)*ddqr(1)+brazo_
robot.lc2^2*ddqr(2)+...

(brazo_robot.l1*brazo_robot.lc2*sin(q(2))*dq(1))*dqr(1)+brazo_robot.lc2*g*c
os(q(1)+q(2));
yy = [yy1; yy2];

%      Calculo de la ley de adaptación
da_hat = gamma*yy'*sigma;
a_hat(i) = a_hat(i)+0.01*da_hat;

%      Calculo de matrices que no dependen del parámetro incierto
Mo11 =
brazo_robot.m1*brazo_robot.lc1^2+brazo_robot.I1+brazo_robot.I2;
Mo12 = brazo_robot.I2;
Mo22 = brazo_robot.I2;

```

```
Mo = [Mo11 Mo12; Mo12 Mo22];

gol = brazo_robot.m1*brazo_robot.lc1*g*cos(q(1));
go = [gol;0];

% Ley de control:
% tau = Y'*a'+Mo(q)qr'+Co(q,q')qr'+go(q)+Kd*sigma
tau = yy*a_hat + Mo*ddq + go + kd_a*sigma;
% Ecuación en lazo cerrado:
% q'' = Mq^-1[tau-C(q,q')q'-g(q)]
ddq = M^-1*(tau-C*dq-G);
end

% Realizar integración
dq1 = dq1 + 0.01*ddq(1);
dq2 = dq2 + 0.01*ddq(2);
q1 = q1 + 0.01*dq1;
q2 = q2 + 0.01*dq2;
% Almacenar valores
tray(1,i) = x;
tray(2,i) = y;
tray_d(1,i) = xa(i);
tray_d(2,i) = ya(i);
pos(1,i) = q1;
pos(2,i) = q2;
pos_d(1,i) = q1n(i);
pos_d(2,i) = q2n(i);
vel(1,i) = dq1;
vel(2,i) = dq2;
vel_d(1,i) = q1d(i);
vel_d(2,i) = q2d(i);
entrada_control(1,i) = tau(1);
entrada_control(2,i) = tau(2);
end

fig1=figure;
plot(tray_d(1,:),tray_d(2,:), 'LineWidth',2.5)
grid on
hold on
plot(tray(1,:),tray(2,:), '--', 'LineWidth',1.5)
legend('Deseado', 'Real', 'Location', 'NorthWest')
```

```
xlabel('Posicion en X','FontSize',10,'FontWeight','bold')
ylabel('Posicion en Y','FontSize',10,'FontWeight','bold')
% saveas(fig1,strcat(int2str(tipo),int2str(ctrl))+".png")

fig2=figure;
subplot(2,1,1)
plot(t,pos_d(1,:),t,pos(1:,:), '--', 'LineWidth',2)
title('Posicion en la junta 1')
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('[Rad]','FontSize',8,'FontWeight','bold')
legend('Desired','Real','Location','southeast')
grid on
subplot(2,1,2)
plot(t,pos_d(2,:),t,pos(2:,:), '--', 'LineWidth',2)
title('Posicion en la junta 2')
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('[Rad]','FontSize',8,'FontWeight','bold')
legend('Deseado','Real','Location','southeast')
grid on
% saveas(fig2,strcat(int2str(tipo),int2str(ctrl))+".png")
%
fig3=figure;
subplot(2,1,1)
plot(t,pos_d(1:)-pos(1:,:), 'LineWidth',2)
title('Error de posicion en la junta 1')
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('Error[rad]','FontSize',8,'FontWeight','bold')
grid on
subplot(2,1,2)
plot(t,pos_d(2:)-pos(2:,:), 'LineWidth',2)
title('Error de posicion en la junta 2')
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('Error[rad]','FontSize',8,'FontWeight','bold')
grid on
% saveas(fig3,strcat(int2str(tipo),int2str(ctrl))+".pos_err.png")
fig4=figure;
subplot(2,1,1)
plot(t,vel_d(1,:),t,vel(1:,:), '--', 'LineWidth',2)
title('Velocidad en la junta 1')
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('[Rad/s]','FontSize',8,'FontWeight','bold')
```

```

legend('Deseado','Real','Location','southeast')
grid on
subplot(2,1,2)
plot(t,vel_d(2,:),t,vel(2:),'--','LineWidth',2)
title('Velocidad en la junta 2')
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('[Rad/s]','FontSize',8,'FontWeight','bold')
legend('Deseado','Real','Location','southeast')
grid on
% saveas(fig4,strcat(int2str(tipo),int2str(ctrl))+"vel.png")
%
fig5=figure;
subplot(2,1,1)
plot(t,vel_d(1:)-vel(1:),'LineWidth',2)
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('Error[rad/s]','FontSize',8,'FontWeight','bold')
title('Error de velocidad en la junta 1')
grid on
subplot(2,1,2)
plot(t,vel_d(2:)-vel(2:),'LineWidth',2)
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('Error[rad/s]','FontSize',8,'FontWeight','bold')
title('Error de velocidad en la junta 2')
grid on
% saveas(fig5,strcat(int2str(tipo),int2str(ctrl))+"vel_err.png")

fig6=figure;
subplot(2,1,1)
plot(t,entrada_control(1:),'LineWidth',2)
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('Torque[Nm]','FontSize',8,'FontWeight','bold')
title('Torque aplicado en la junta 1')
grid on
subplot(2,1,2)
plot(t,entrada_control(2:),'LineWidth',2)
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('Torque[Nm]','FontSize',8,'FontWeight','bold')
title('Torque aplicado en la junta 2')
grid on
% saveas(fig7,strcat(int2str(tipo),int2str(ctrl))+"par.png")

```

ANEXO 3: Código Trayectoria lineal con cambio de dirección

```
close all
clc
clear
t=0:0.001:1;
xf=t;
yf=0.7*t+1;

% ----- Inicialización de parámetros ----- %

% Constantes generales
g = 9.8;
tray = zeros(2, length(t));
pos = zeros(2, length(t));
vel = zeros(2, length(t));
tray_d = zeros(2, length(t));
pos_d = zeros(2, length(t));
vel_d = zeros(2, length(t));
entrada_control = zeros(2, length(t));

% Parámetros del modelo dinámico
brazo_robot.m1 = 1;
brazo_robot.m2 = 2;
brazo_robot.l1 = 1;
brazo_robot.l2 = 1;
brazo_robot.I1 = 0.12;
brazo_robot.lc1 = 0.5;
brazo_robot.I2 = 0.25;
brazo_robot.lc2 = 0.6;

% Parámetros reales del robot
m1 = 1;
m2 = 3;
l1 = 1;
l2 = 1;
I1 = 0.12;
lc1 = 0.5;
I2 = 0.25;
lc2 = 0.6;
m2_a = 2.2;
```

```

% Estado inicial
q1 = 0; % Posición de la junta 1
q2 = pi/2; % Posición de la junta 2
dq1 = 0; % Velocidad de la junta 1
dq2 = 0; % Velocidad de la junta 2

% Ganancias
zeta = 1;
wn = 10;
kp = wn^2*eye(2);
kd = 2*zeta*wn*eye(2);

% Ganancias del control adaptativo
kd_a = 80*eye(2);
lambda = 20*eye(2);
gamma = 0.005;
a_hat = zeros(1,length(t));

% Matriz de inercia constante de control PD
Mo11 = m1*lc1^2 + I1 + m2*(l1^2 + lc2^2) + I2;
Mo12 = m2*lc2^2 + I2;
Mo22 = m2*lc2^2 + I2;
Mo = [Mo11, Mo12; Mo12, Mo22];

% ----- Secuencia principal ----- %

% Selección del tipo de control
disp("Selección de control:")
disp("1- PD control")
disp("2- PD aumentado")
disp("3- Torque Computado")
disp("4- Control Adaptativo")
ctrl = input('Ingresa el tipo de control (1~4):');
if ctrl == 1
    c="PD";
elseif ctrl == 2
    c="PD aumentado";
elseif ctrl == 3
    c="Torque-computado";
elseif ctrl == 4

```

```

c="Adaptativo";
end

%CINEMÁTICA INVERSA
for i=1:length(t)

D(i)=(xf(i))^2+(yf(i))^2-l2^2-l1^2)/(2*l2*l1);
q2n(i)= atan2(+sqrt(1-D(i)^2),D(i));
q1n(i)= atan2(yf(i),xf(i))-atan2(l2*sin(q2n(i)),l1+l2*cos(q2n(i)));
q1g=q1n*180/pi;
q2g=q2n*180/pi;
q2d(i)=-((cos(t(i))*(sin(t(i))/2 + 1/2))/2 - (sin(t(i))*(cos(t(i))/2 +
1/2))/2)/(1 - ((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 -
1)^2)^(1/2);
q1d(i)=((sin(t(i))*(cos(t(i))/2 + 1/2))/2 - (cos(t(i))*(sin(t(i))/2 +
1/2))/2+ (((cos(t(i))*(sin(t(i))/2 + 1/2))/2 - (sin(t(i))*(cos(t(i))/2 +
1/2))/2)*((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 - 1))/(1 -
((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 -
1)^2)^(1/2))/(((sin(t(i))/2 + 1/2)^2/2 + (1 - ((sin(t(i))/2 + 1/2)^2/2 +
(cos(t(i))/2 + 1/2)^2/2 - 1)^2)^(1/2) + (cos(t(i))/2 + 1/2)^2/2 - 1)^2 + 1)
+ (cos(t(i))/(2*(cos(t(i))/2 + 1/2)) + (sin(t(i))*(sin(t(i))/2 +
1/2))/(2*(cos(t(i))/2 + 1/2)^2))/((sin(t(i))/2 + 1/2)^2/(cos(t(i))/2 +
1/2)^2 + 1);
q2dd(i)=- (cos(t(i))^2/4 + sin(t(i))^2/4 - (cos(t(i))*(cos(t(i))/2 +
1/2))/2 - (sin(t(i))*(sin(t(i))/2 + 1/2))/2)/(1 - ((sin(t(i))/2 + 1/2)^2/2
+ (cos(t(i))/2 + 1/2)^2/2 - 1)^2)^(1/2) - (((cos(t(i))*(sin(t(i))/2 +
1/2))/2 - (sin(t(i))*(cos(t(i))/2 + 1/2))/2)^2*((sin(t(i))/2 + 1/2)^2/2 +
(cos(t(i))/2 + 1/2)^2/2 - 1))/(1 - ((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2
+ 1/2)^2/2 - 1)^2)^(3/2);
q1dd(i)=((cos(t(i))*sin(t(i)))/(2*(cos(t(i))/2 + 1/2)^2) -
sin(t(i))/(2*(cos(t(i))/2 + 1/2)) + (sin(t(i))^2*(sin(t(i))/2 +
1/2))/(2*(cos(t(i))/2 + 1/2)^3) + (cos(t(i))*(sin(t(i))/2 +
1/2))/(2*(cos(t(i))/2 + 1/2)^2))/((sin(t(i))/2 + 1/2)^2/(cos(t(i))/2 +
1/2)^2 + 1) + (((cos(t(i))*(sin(t(i))/2 + 1/2))/2 - (sin(t(i))*(cos(t(i))/2
+ 1/2))/2)^2/(1 - ((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 -
1)^2)^(1/2) - sin(t(i))^2/4 - cos(t(i))^2/4 + (cos(t(i))*(cos(t(i))/2 +
1/2))/2 + (sin(t(i))*(sin(t(i))/2 + 1/2))/2 + (((sin(t(i))/2 + 1/2)^2/2 +
(cos(t(i))/2 + 1/2)^2/2 - 1)*(cos(t(i))^2/4 + sin(t(i))^2/4 -
(cos(t(i))*(cos(t(i))/2 + 1/2))/2 - (sin(t(i))*(sin(t(i))/2 + 1/2))/2))/(1
- ((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 - 1)^2)^(1/2) +

```

```

(((cos(t(i))*(sin(t(i))/2 + 1/2))/2 - (sin(t(i))*(cos(t(i))/2 +
1/2))/2)^2*((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 - 1)^2)/(1 -
((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 -
1)^2)^(3/2))/(((sin(t(i))/2 + 1/2)^2/2 + (1 - ((sin(t(i))/2 + 1/2)^2/2 +
(cos(t(i))/2 + 1/2)^2/2 - 1)^2)^(1/2) + (cos(t(i))/2 + 1/2)^2/2 - 1)^2 + 1)
- ((cos(t(i))/(2*(cos(t(i))/2 + 1/2)) + (sin(t(i))*(sin(t(i))/2 +
1/2))/(2*(cos(t(i))/2 + 1/2)^2))*((sin(t(i))*(sin(t(i))/2 +
1/2)^2)/(cos(t(i))/2 + 1/2)^3 + (cos(t(i))*(sin(t(i))/2 +
1/2))/(cos(t(i))/2 + 1/2)^2))/((sin(t(i))/2 + 1/2)^2/(cos(t(i))/2 + 1/2)^2
+ 1)^2 + (2*((sin(t(i))*(cos(t(i))/2 + 1/2))/2 - (cos(t(i))*(sin(t(i))/2 +
1/2))/2 + (((cos(t(i))*(sin(t(i))/2 + 1/2))/2 - (sin(t(i))*(cos(t(i))/2 +
1/2))/2)*((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 - 1))/(1 -
((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 -
1)^2)^(1/2))^2*((sin(t(i))/2 + 1/2)^2/2 + (1 - ((sin(t(i))/2 + 1/2)^2/2 +
(cos(t(i))/2 + 1/2)^2/2 - 1)^2)^(1/2) + (cos(t(i))/2 + 1/2)^2/2 -
1))/(((sin(t(i))/2 + 1/2)^2/2 + (1 - ((sin(t(i))/2 + 1/2)^2/2 +
(cos(t(i))/2 + 1/2)^2/2 - 1)^2)^(1/2) + (cos(t(i))/2 + 1/2)^2/2 - 1)^2 +
1)^2;
xa(i) = l1*cos(q1n(i))+l2*cos(q1n(i)+q2n(i));
ya(i) = l1*sin(q1n(i))+l2*sin(q1n(i)+q2n(i));
end

for i=1:length(t)
    q = [q1; q2];
    dq = [dq1; dq2];

    % Posición actual de la extremidad del robot (x,y)
    [x, y] = cinematica_directa(q, brazo_robot);
    % Calculo de la posición deseada de las juntas (q_deseado)
    [q1_d, q2_d] = trayectorial(tipo, t(i), 1);
    q_d = [q1n(i); q2n(i)];
    % Calculo de la velocidad deseada de las juntas (q'_deseado)
    [dq1_d, dq2_d] = trayectorial(tipo, t(i), 2);
    dq_d = [q1d(i); q2d(i)];
    % Calculo de la aceleración deseada de las juntas (q''_deseado)
    [ddq1_d, ddq2_d] = trayectorial(tipo, t(i), 3);
    ddq_d = [q1dd(i); q2dd(i)];
    % Calculo de la posición deseada de la extremidad del robot
    (x_deseado,y_deseado)

```

```
[xd, yd] = cinematica_directa(q_d, brazo_robot);  
%   Calculo del error de posición y velocidad de las juntas  
e = q_d - q;  
de = dq_d - dq;  
  
if ctrl == 1 % <-----PD  
%   Ley de control:  
%   tau = kd*e'+kp*e  
kp = Mo*(wn^2*eye(2));  
kd = Mo*(2*zeta*wn*eye(2));  
tau = kd*de+kp*e;  
%   Calculo de las matrices del modelo dinamico del robot  
M = matriz_inercia(brazo_robot, q);  
C = matriz_fuerzas(brazo_robot, q, dq);  
g = matriz_gravitacional(brazo_robot, q);  
%   Ecuación en lazo cerrado:  
%   q'' = M(q)^-1[tau-C(q,q')*q'-g(q)]  
ddq = M^-1*(tau-C*dq-g);  
  
elseif ctrl == 2 % <-----PD aumentado  
%   Calculo de las matrices del modelo dinamico del robot  
M = matriz_inercia(brazo_robot, q);  
C = matriz_fuerzas(brazo_robot, q, dq);  
g = matriz_gravitacional(brazo_robot, q);  
%   Ley de control:  
%   tau = M(q)*qd''+C(q,q')*qd'+g(q)+kd*e'+kp*e;  
tau = M*ddq_d+C*dq_d+g+kd*de+kp*e;  
%   Ecuación en lazo cerrado:  
%   q'' = M(q)^-1[M(q)*qd''+[C(q,q')+kd]*e'+kp*e]  
ddq = M^-1*(M*ddq_d+(C+kd)*de+kp*e);  
  
elseif ctrl == 3 % <-----Torque computado  
%   Calculo de las matrices del modelo dinamico del robot  
M = matriz_inercia(brazo_robot, q);  
C = matriz_fuerzas(brazo_robot, q, dq);  
g = matriz_gravitacional(brazo_robot, q);  
%   Ley de control:  
%   tau = M(q)*[qd''+kd*e'+kp*e]+C(q,q')*q+g(q);  
tau = M*(ddq_d+kd*de+kp*e)+C*dq+g;  
%   Ecuación en lazo cerrado:
```

```

%           q'' = qd''+kd*e'+kp*e
           ddq = ddq_d+kd*de+kp*e;

elseif ctrl == 4 % <-----Control adaptativo
%           Calculo de las matrices del modelo dinamico del robot
M = matriz_inercia(brazo_robot, q);
C = matriz_fuerzas(brazo_robot, q, dq);
G = matriz_gravitacional(brazo_robot, q);
%           Calculo de qr y qr'
dqr = dq_d + lambda*(q_d-q);
ddqr = ddq_d + lambda*(dq_d-q);
%           Calculo de sigma
sigma = dqr-dq;
%           Parametrización del modelo dinamico
yy1 =
(brazo_robot.l1^2+brazo_robot.lc2^2+2*brazo_robot.l1*cos(q(2)))*ddqr(1)+(br
azo_robot.l1*brazo_robot.lc2*cos(q(2)))+...

brazo_robot.lc2^2)*ddqr(2)+(brazo_robot.l1*brazo_robot.lc2*sin(q(2)))*dqr(1
)-...

(brazo_robot.l1*brazo_robot.lc2*sin(q(2))*(dq(1)+dq(2)))*dqr(2)+...
           g*(brazo_robot.lc2*cos(q(1)+q(2))+brazo_robot.l1*cos(q(1)));
           yy2 =
(brazo_robot.l1*brazo_robot.lc2*cos(q(2))+brazo_robot.lc2^2)*ddqr(1)+brazo_
robot.lc2^2*ddqr(2)+...

(brazo_robot.l1*brazo_robot.lc2*sin(q(2))*dq(1))*dqr(1)+brazo_robot.lc2*g*c
os(q(1)+q(2));
           yy = [yy1; yy2];
%           Calculo de la ley de adaptación
da_hat = gamma*yy'*sigma;
a_hat(i) = a_hat(i)+0.01*da_hat;
%           Calculo de matrices que no dependen del parámetro incierto
Mo11 =
brazo_robot.m1*brazo_robot.lc1^2+brazo_robot.I1+brazo_robot.I2;
Mo12 = brazo_robot.I2;
Mo22 = brazo_robot.I2;
Mo = [Mo11 Mo12; Mo12 Mo22];

```

```

    go1 = brazo_robot.m1*brazo_robot.lc1*g*cos(q(1));
    go = [go1;0];

% Ley de control:
% tau = Y'*a'+Mo(q)qr'+Co(q,q')qr'+go(q)+Kd*sigma
tau = yy*a_hat + Mo*ddqr + go + kd_a*sigma;

% Ecuación en lazo cerrado:
% q'' = Mq^-1[tau-C(q,q')q'-g(q)]
ddq = M^-1*(tau-C*dq-G);
end

% Realizar integración
dq1 = dq1 + 0.01*ddq(1);
dq2 = dq2 + 0.01*ddq(2);
q1 = q1 + 0.01*dq1;
q2 = q2 + 0.01*dq2;
% Almacenar valores
tray1(1,i) = x;
tray1(2,i) = y;
tray_d1(1,i) = xa(i);
tray_d1(2,i) = ya(i);
pos(1,i) = q1;
pos(2,i) = q2;
pos_d(1,i) = q1n(i);
pos_d(2,i) = q2n(i);
vel(1,i) = dq1;
vel(2,i) = dq2;
vel_d(1,i) = q1d(i);
vel_d(2,i) = q2d(i);
entrada_control(1,i) = tau(1);
entrada_control(2,i) = tau(2);
end

%SEGUNDO TRAMO

t=1:0.01:1.5;
xf=t;
yf=-2.4*t+4.1;

% ----- Inicialización de parámetros ----- %

```

```
% Constantes generales
g = 9.8;

tray = zeros(2, length(t));
pos = zeros(2, length(t));
vel = zeros(2, length(t));
tray_d = zeros(2, length(t));
pos_d = zeros(2, length(t));
vel_d = zeros(2, length(t));
entrada_control = zeros(2, length(t));

% Parámetros del modelo dinámico
brazo_robot.m1 = 1;
brazo_robot.m2 = 2;
brazo_robot.l1 = 1;
brazo_robot.l2 = 1;
brazo_robot.I1 = 0.12;
brazo_robot.lc1 = 0.5;
brazo_robot.I2 = 0.25;
brazo_robot.lc2 = 0.6;

% Parámetros reales del robot
m1 = 1;
m2 = 3;
l1 = 1;
l2 = 1;
I1 = 0.12;
lc1 = 0.5;
I2 = 0.25;
lc2 = 0.6;
m2_a = 2.2;

% Estado inicial
% Ganancias
zeta = 1;
wn = 10;
kp = wn^2*eye(2);
kd = 2*zeta*wn*eye(2);
```

```

% Ganancias del control adaptativo
kd_a = 6*eye(2);
lambda = 70*eye(2);
gamma = 0.0001;
a_hat = zeros(1,length(t));
% kd_a = 5*eye(2); MEJORES
% lambda = 60*eye(2);
% gamma = 0.0005;
% Matriz de inercia constante de control PD
Mo11 = m1*lc1^2 + I1 + m2*(l1^2 + lc2^2) + I2;
Mo12 = m2*lc2^2 + I2;
Mo22 = m2*lc2^2 + I2;
Mo = [Mo11, Mo12; Mo12, Mo22];

% ----- Secuencia principal ----- %

% Selección del tipo de control
disp("Selección de control:")
disp("1- PD control")
disp("2- PD aumentado")
disp("3- Torque Computado")
disp("4- Control Adaptativo")
ctrl = input('Ingresa el tipo de control (1~4):');
if ctrl == 1
    c="PD";
elseif ctrl == 2
    c="PD aumentado";
elseif ctrl == 3
    c="Torque-computado";
elseif ctrl == 4
    c="Adaptativo";
end

for i=1:length(t)

D(i)=((xf(i))^2+(yf(i))^2-l2^2-l1^2)/(2*l2*l1);
q2n(i)= atan2(+sqrt(1-D(i)^2),D(i));
q1n(i)= atan2(yf(i),xf(i))-atan2(l2*sin(q2n(i)),l1+l2*cos(q2n(i)));
q1g=q1n*180/pi;

```

$$q2g=q2n*180/pi;$$

$$q2d(i)=-((\cos(t(i))*(\sin(t(i))/2 + 1/2))/2 - (\sin(t(i))*(\cos(t(i))/2 + 1/2))/2)/(1 - ((\sin(t(i))/2 + 1/2)^2/2 + (\cos(t(i))/2 + 1/2)^2/2 - 1)^2)^{(1/2)};$$

$$q1d(i)=((\sin(t(i))*(\cos(t(i))/2 + 1/2))/2 - (\cos(t(i))*(\sin(t(i))/2 + 1/2))/2 + (((\cos(t(i))*(\sin(t(i))/2 + 1/2))/2 - (\sin(t(i))*(\cos(t(i))/2 + 1/2))/2)*((\sin(t(i))/2 + 1/2)^2/2 + (\cos(t(i))/2 + 1/2)^2/2 - 1))/(1 - ((\sin(t(i))/2 + 1/2)^2/2 + (\cos(t(i))/2 + 1/2)^2/2 - 1)^2)^{(1/2)})/(((\sin(t(i))/2 + 1/2)^2/2 + (1 - ((\sin(t(i))/2 + 1/2)^2/2 + (\cos(t(i))/2 + 1/2)^2/2 - 1)^2)^{(1/2)} + (\cos(t(i))/2 + 1/2)^2/2 - 1)^2 + 1) + (\cos(t(i))/(2*(\cos(t(i))/2 + 1/2)) + (\sin(t(i))*(\sin(t(i))/2 + 1/2))/(2*(\cos(t(i))/2 + 1/2)^2))/((\sin(t(i))/2 + 1/2)^2/(\cos(t(i))/2 + 1/2)^2 + 1);$$

$$q2dd(i)=- (\cos(t(i))^2/4 + \sin(t(i))^2/4 - (\cos(t(i))*(\cos(t(i))/2 + 1/2))/2 - (\sin(t(i))*(\sin(t(i))/2 + 1/2))/2)/(1 - ((\sin(t(i))/2 + 1/2)^2/2 + (\cos(t(i))/2 + 1/2)^2/2 - 1)^2)^{(1/2)} - (((\cos(t(i))*(\sin(t(i))/2 + 1/2))/2 - (\sin(t(i))*(\cos(t(i))/2 + 1/2))/2)^2*((\sin(t(i))/2 + 1/2)^2/2 + (\cos(t(i))/2 + 1/2)^2/2 - 1))/(1 - ((\sin(t(i))/2 + 1/2)^2/2 + (\cos(t(i))/2 + 1/2)^2/2 - 1)^2)^{(3/2)};$$

$$q1dd(i)=((\cos(t(i))*\sin(t(i)))/(2*(\cos(t(i))/2 + 1/2)^2) - \sin(t(i))/(2*(\cos(t(i))/2 + 1/2)) + (\sin(t(i))^2*(\sin(t(i))/2 + 1/2))/(2*(\cos(t(i))/2 + 1/2)^3) + (\cos(t(i))*(\sin(t(i))/2 + 1/2))/(2*(\cos(t(i))/2 + 1/2)^2))/((\sin(t(i))/2 + 1/2)^2/(\cos(t(i))/2 + 1/2)^2 + 1) + (((\cos(t(i))*(\sin(t(i))/2 + 1/2))/2 - (\sin(t(i))*(\cos(t(i))/2 + 1/2))/2)^2/(1 - ((\sin(t(i))/2 + 1/2)^2/2 + (\cos(t(i))/2 + 1/2)^2/2 - 1)^2)^{(1/2)} - \sin(t(i))^2/4 - \cos(t(i))^2/4 + (\cos(t(i))*(\cos(t(i))/2 + 1/2))/2 + (\sin(t(i))*(\sin(t(i))/2 + 1/2))/2 + (((\sin(t(i))/2 + 1/2)^2/2 + (\cos(t(i))/2 + 1/2)^2/2 - 1)*(\cos(t(i))^2/4 + \sin(t(i))^2/4 - (\cos(t(i))*(\cos(t(i))/2 + 1/2))/2 - (\sin(t(i))*(\sin(t(i))/2 + 1/2))/2))/(1 - ((\sin(t(i))/2 + 1/2)^2/2 + (\cos(t(i))/2 + 1/2)^2/2 - 1)^2)^{(1/2)} + (((\cos(t(i))*(\sin(t(i))/2 + 1/2))/2 - (\sin(t(i))*(\cos(t(i))/2 + 1/2))/2)^2*((\sin(t(i))/2 + 1/2)^2/2 + (\cos(t(i))/2 + 1/2)^2/2 - 1)^2)/(1 - ((\sin(t(i))/2 + 1/2)^2/2 + (\cos(t(i))/2 + 1/2)^2/2 - 1)^2)^{(3/2)})/(((\sin(t(i))/2 + 1/2)^2/2 + (1 - ((\sin(t(i))/2 + 1/2)^2/2 + (\cos(t(i))/2 + 1/2)^2/2 - 1)^2)^{(1/2)} + (\cos(t(i))/2 + 1/2)^2/2 - 1)^2 + 1) - ((\cos(t(i))/(2*(\cos(t(i))/2 + 1/2)) + (\sin(t(i))*(\sin(t(i))/2 + 1/2))/(2*(\cos(t(i))/2 + 1/2)^2))*((\sin(t(i))*(\sin(t(i))/2 + 1/2)^2)/(\cos(t(i))/2 + 1/2)^3 + (\cos(t(i))*(\sin(t(i))/2 + 1/2))/(\cos(t(i))/2 + 1/2)^2))/((\sin(t(i))/2 + 1/2)^2/(\cos(t(i))/2 + 1/2)^2 + 1)^2 + (2*((\sin(t(i))*(\cos(t(i))/2 + 1/2))/2 - (\cos(t(i))*(\sin(t(i))/2 + 1/2))/2 + (((\cos(t(i))*(\sin(t(i))/2 + 1/2))/2 - (\sin(t(i))*(\cos(t(i))/2 + 1/2))/2$$

```
1/2))/2)*((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 - 1))/(1 -
((sin(t(i))/2 + 1/2)^2/2 + (cos(t(i))/2 + 1/2)^2/2 -
1)^2)^(1/2))^2*((sin(t(i))/2 + 1/2)^2/2 + (1 - ((sin(t(i))/2 + 1/2)^2/2 +
(cos(t(i))/2 + 1/2)^2/2 - 1)^2)^(1/2) + (cos(t(i))/2 + 1/2)^2/2 -
1))/(((sin(t(i))/2 + 1/2)^2/2 + (1 - ((sin(t(i))/2 + 1/2)^2/2 +
(cos(t(i))/2 + 1/2)^2/2 - 1)^2)^(1/2) + (cos(t(i))/2 + 1/2)^2/2 - 1)^2 +
1)^2;
```

```
xa(i) = l1*cos(q1n(i))+l2*cos(q1n(i)+q2n(i));
```

```
ya(i) = l1*sin(q1n(i))+l2*sin(q1n(i)+q2n(i));
```

```
end
```

```
for i=1:length(t)
```

```
    q = [q1; q2];
```

```
    dq = [dq1; dq2];
```

```
% Posición actual de la extremidad del robot (x,y)
```

```
[x, y] = cinematica_directa(q, brazo_robot);
```

```
% Calculo de la posición deseada de las juntas (q_deseado)
```

```
% [q1_d, q2_d] = trayectorial(tipo, t(i), 1);
```

```
q_d = [q1n(i); q2n(i)];
```

```
% Calculo de la velocidad deseada de las juntas (q'_deseado)
```

```
% [dq1_d, dq2_d] = trayectorial(tipo, t(i), 2);
```

```
dq_d = [q1d(i); q2d(i)];
```

```
% Calculo de la aceleración deseada de las juntas (q''_deseado)
```

```
% [ddq1_d, ddq2_d] = trayectorial(tipo, t(i), 3);
```

```
ddq_d = [q1dd(i); q2dd(i)];
```

```
% Calculo de la posición deseada de la extremidad del robot
```

```
(x_deseado,y_deseado)
```

```
[xd, yd] = cinematica_directa(q_d, brazo_robot);
```

```
% Calculo del error de posición y velocidad de las juntas
```

```
e = q_d - q;
```

```
de = dq_d - dq;
```

```
if ctrl == 1 % <-----PD
```

```
% Ley de control:
```

```
% tau = kd*e'+kp*e
```

```
kp = Mo*(wn^2*eye(2));
```

```
kd = Mo*(2*zeta*wn*eye(2));
```

```
tau = kd*de+kp*e;
```

```

%      Calculo de las matrices del modelo dinamico del robot
M = matriz_inercia(brazo_robot, q);
C = matriz_fuerzas(brazo_robot, q, dq);
g = matriz_gravitacional(brazo_robot, q);
%      Ecuación en lazo cerrado:
%       $q'' = M(q)^{-1}[\tau - C(q, q') * q' - g(q)]$ 
ddq = M^{-1}*(tau - C*dq - g);

elseif ctrl == 2 % <-----PD aumentado
%      Calculo de las matrices del modelo dinamico del robot
M = matriz_inercia(brazo_robot, q);
C = matriz_fuerzas(brazo_robot, q, dq);
g = matriz_gravitacional(brazo_robot, q);
%      Ley de control:
%       $\tau = M(q) * qd'' + C(q, q') * qd' + g(q) + kd * e' + kp * e;$ 
tau = M*ddq_d + C*dq_d + g + kd*de + kp*e;
%      Ecuación en lazo cerrado:
%       $q'' = M(q)^{-1}[M(q) * qd'' + [C(q, q') + kd] * e' + kp * e]$ 
ddq = M^{-1}*(M*ddq_d + (C+kd)*de + kp*e);

elseif ctrl == 3 % <-----Torque computado
%      Calculo de las matrices del modelo dinamico del robot
M = matriz_inercia(brazo_robot, q);
C = matriz_fuerzas(brazo_robot, q, dq);
g = matriz_gravitacional(brazo_robot, q);
%      Ley de control:
%       $\tau = M(q) * [qd'' + kd * e' + kp * e] + C(q, q') * q + g(q);$ 
tau = M*(ddq_d + kd*de + kp*e) + C*dq + g;
%      Ecuación en lazo cerrado:
%       $q'' = qd'' + kd * e' + kp * e$ 
ddq = ddq_d + kd*de + kp*e;
elseif ctrl == 4 % <-----Control adaptativo
%      Calculo de las matrices del modelo dinamico del robot
M = matriz_inercia(brazo_robot, q);
C = matriz_fuerzas(brazo_robot, q, dq);
G = matriz_gravitacional(brazo_robot, q);
%      Calculo de qr y qr'
dqr = dq_d + lambda*(q_d - q);
ddqr = ddq_d + lambda*(dq_d - q);
%      Calculo de sigma

```

```

sigma = dqr-dq;
%   Parametrización del modelo dinamico
yy1 =
(brazo_robot.l1^2+brazo_robot.lc2^2+2*brazo_robot.l1*cos(q(2)))*ddqr(1)+(br
azo_robot.l1*brazo_robot.lc2*cos(q(2)))+...

brazo_robot.lc2^2)*ddqr(2)+(brazo_robot.l1*brazo_robot.lc2*sin(q(2)))*dqr(1
)-...

(brazo_robot.l1*brazo_robot.lc2*sin(q(2))*(dq(1)+dq(2)))*dqr(2)+...
g*(brazo_robot.lc2*cos(q(1)+q(2))+brazo_robot.l1*cos(q(1)));
yy2 =
(brazo_robot.l1*brazo_robot.lc2*cos(q(2))+brazo_robot.lc2^2)*ddqr(1)+brazo_
robot.lc2^2*ddqr(2)+...

(brazo_robot.l1*brazo_robot.lc2*sin(q(2))*dq(1))*dqr(1)+brazo_robot.lc2*g*c
os(q(1)+q(2));
yy = [yy1; yy2];
%   Calculo de la ley de adaptación
da_hat = gamma*yy'*sigma;
a_hat(i) = a_hat(i)+0.01*da_hat;
%   Calculo de matrices que no dependen del parámetro incierto
Mo11 =
brazo_robot.m1*brazo_robot.lc1^2+brazo_robot.I1+brazo_robot.I2;
Mo12 = brazo_robot.I2;
Mo22 = brazo_robot.I2;
Mo = [Mo11 Mo12; Mo12 Mo22];

go1 = brazo_robot.m1*brazo_robot.lc1*g*cos(q(1));
go = [go1;0];
%   Ley de control:
%   tau = Y'*a'+Mo(q)qr'+Co(q,q')qr'+go(q)+Kd*sigma
tau = yy*a_hat + Mo*ddqr + go + kd_a*sigma;
%   Ecuación en lazo cerrado:
%   q'' = Mq^-1[tau-C(q,q')q'-g(q)]
ddq = M^-1*(tau-C*dq-G);
end

%   Realizar integración
dq1 = dq1 + 0.01*ddq(1);

```

```

dq2 = dq2 + 0.01*ddq(2);
q1 = q1 + 0.01*dq1;
q2 = q2 + 0.01*dq2;
%   Almacenar valores
tray2(1,i) = x;
tray2(2,i) = y;
tray_d2(1,i) = xa(i);
tray_d2(2,i) = ya(i);
pos(1,i) = q1;
pos(2,i) = q2;
pos_d(1,i) = q1n(i);
pos_d(2,i) = q2n(i);
vel(1,i) = dq1;
vel(2,i) = dq2;
vel_d(1,i) = q1d(i);
vel_d(2,i) = q2d(i);
entrada_control(1,i) = tau(1);
entrada_control(2,i) = tau(2);
end
fig1=figure;
plot(tray_d1(1,:),tray_d1(2,:), 'b', 'LineWidth',2.5)

grid on
hold on
plot(tray_d2(1,:),tray_d2(2,:), 'b', 'LineWidth',2.5)

grid on
hold on
plot(tray1(1,:),tray1(2,:), '--', 'LineWidth',1.5)
grid on
hold on
plot(tray2(1,:),tray2(2,:), '--', 'LineWidth',1.5)
legend('Deseado', 'Location', 'NorthWest')
xlabel('Posicion en X', 'FontSize',10, 'FontWeight', 'bold')
ylabel('Posicion en Y', 'FontSize',10, 'FontWeight', 'bold')
% saveas(fig1,strcat(int2str(tipo),int2str(ctrl))+".tra.png")

fig2=figure;
subplot(2,1,1)
plot(t,pos_d(1,:),t,pos(1,:), '--', 'LineWidth',2)

```

```

title('Posicion en la junta 1')
xlabel('Tiempo[s]', 'FontSize',8, 'FontWeight', 'bold')
ylabel('[Rad]', 'FontSize',8, 'FontWeight', 'bold')
legend('Desired', 'Real', 'Location', 'southeast')
grid on
subplot(2,1,2)
plot(t,pos_d(2,:),t,pos(2,:), '--', 'LineWidth',2)
title('Posicion en la junta 2')
xlabel('Tiempo[s]', 'FontSize',8, 'FontWeight', 'bold')
ylabel('[Rad]', 'FontSize',8, 'FontWeight', 'bold')
legend('Deseado', 'Real', 'Location', 'southeast')
grid on
% saveas(fig2, strcat(int2str(tipo),int2str(ctrl))+"pos.png")
%
fig3=figure;
subplot(2,1,1)
plot(t,pos_d(1,:)-pos(1,:), 'LineWidth',2)
title('Error de posicion en la junta 1')
xlabel('Tiempo[s]', 'FontSize',8, 'FontWeight', 'bold')
ylabel('Error[rad]', 'FontSize',8, 'FontWeight', 'bold')
grid on
subplot(2,1,2)
plot(t,pos_d(2,:)-pos(2,:), 'LineWidth',2)
title('Error de posicion en la junta 2')
xlabel('Tiempo[s]', 'FontSize',8, 'FontWeight', 'bold')
ylabel('Error[rad]', 'FontSize',8, 'FontWeight', 'bold')
grid on
% saveas(fig3, strcat(int2str(tipo),int2str(ctrl))+"pos_err.png")

fig4=figure;
subplot(2,1,1)
plot(t,vel_d(1,:),t,vel(1,:), '--', 'LineWidth',2)
title('Velocidad en la junta 1')
xlabel('Tiempo[s]', 'FontSize',8, 'FontWeight', 'bold')
ylabel('[Rad/s]', 'FontSize',8, 'FontWeight', 'bold')
legend('Deseado', 'Real', 'Location', 'southeast')
grid on
subplot(2,1,2)
plot(t,vel_d(2,:),t,vel(2,:), '--', 'LineWidth',2)
title('Velocidad en la junta 2')
xlabel('Tiempo[s]', 'FontSize',8, 'FontWeight', 'bold')

```

```
ylabel(' [Rad/s]','FontSize',8,'FontWeight','bold')
legend('Deseado','Real','Location','southeast')
grid on
% saveas(fig4,strcat(int2str(tipo),int2str(ctrl))+"vel.png")
%
fig5=figure;
subplot(2,1,1)
plot(t,vel_d(1,:)-vel(1:,:), 'LineWidth',2)
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('Error[rad/s]','FontSize',8,'FontWeight','bold')
title('Error de velocidad en la junta 1')
grid on
subplot(2,1,2)
plot(t,vel_d(2,:)-vel(2:,:), 'LineWidth',2)
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('Error[rad/s]','FontSize',8,'FontWeight','bold')
title('Error de velocidad en la junta 2')
grid on
% saveas(fig5,strcat(int2str(tipo),int2str(ctrl))+"vel_err.png")

fig6=figure;
subplot(2,1,1)
plot(t,entrada_control(1,:), 'LineWidth',2)
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('Torque[Nm]','FontSize',8,'FontWeight','bold')
title('Torque aplicado en la junta 1')
grid on
subplot(2,1,2)
plot(t,entrada_control(2,:), 'LineWidth',2)
xlabel('Tiempo[s]','FontSize',8,'FontWeight','bold')
ylabel('Torque[Nm]','FontSize',8,'FontWeight','bold')
title('Torque aplicado en la junta 2')
grid on
% saveas(fig7,strcat(int2str(tipo),int2str(ctrl))+"par.png")
```

ANEXO 4: Función para calcular la cinemática directa del brazo robótico

```
function [x, y] = cinematica_directa(q, brazo_robot)
% Calcula la cinemática directa del robot
x = brazo_robot.l1*cos(q(1))+brazo_robot.l2*cos(q(1)+q(2));
y = brazo_robot.l1*sin(q(1))+brazo_robot.l2*sin(q(1)+q(2));
```

ANEXO 5: Función para calcular la matriz de inercia del brazo robótico

```
function M = matriz_inercia(brazo_robot,q)
% Calcula la matriz de inercia del robot
m1 = brazo_robot.m1;
lc1 = brazo_robot.lc1;
I1 = brazo_robot.I1;
m2 = brazo_robot.m2;
l1 = brazo_robot.l1;
lc2 = brazo_robot.lc2;
q2 = q(2);
I2 = brazo_robot.I2;

M11 = m1*lc1^2 + I1 +m2*(l1^2 + lc2^2 + 2*l1*lc2*cos(q2)) + I2;
M12 = m2*l1*lc2*cos(q2) + m2*lc2^2 + I2;
M22 = m2*lc2^2 + I2;

M = [M11, M12; M12, M22];
```

ANEXO 6: Función para calcular la matriz de fuerzas del brazo robótico

```
function C = matriz_fuerzas(robot_arm,q, q_dot)
% Calcula la matriz de fuerzas del robot
m2 = robot_arm.m2;
l1 = robot_arm.l1;
lc2 = robot_arm.lc2;

h = m2*l1*lc2*sin(q(2));

C11 = h*q_dot(2);
C12 = -h*(q_dot(1)+q_dot(2));
C21 = h*q_dot(1);
C22 = 0;

C = [C11, C12; C21, C22];
```

ANEXO 7: Función para calcular la matriz gravitacional del brazo robótico

```
function G = matriz_gravitacional(brazo_robot,q)
% Calcula la matriz gravitacional del robot
m1 = brazo_robot.m1;
lc1 = brazo_robot.lc1;
m2 = brazo_robot.m2;
l1 = brazo_robot.l1;
lc2 = brazo_robot.lc2;
q1 = q(1);
q2 = q(2);
g = 9.8;

g1 = m1*lc1*g*cos(q1) + m2*g*(lc2*cos(q1+q2)+ l1*cos(q1));
g2 = m2*lc2*g*cos(q1+q2);

G = [g1;g2];
```

ANEXO 8: Función para calcular la posición, velocidad y aceleración deseada del brazo robótico

```
function [qdx, qdy]=trayectoria(tipo, t, selector)
% Calcula la posición, velocidad y aceleración deseada del robot
R = 0.75;
% qd
if selector == 1
% trayectoria lenta
if tipo == 1
qdx = -R*cos(pi*t/2)-R;
qdy = -R*cos(pi*t/2)+5*R;
% trayectoria rapida
elseif tipo == 2
qdx = -R*cos(pi*t)-R;
qdy = -R*cos(pi*t)+5*R;
end
% qd'
elseif selector == 2
% trayectoria lenta
if tipo == 1
qdx = pi*R*sin(pi*t/2)/2;
qdy = pi*R*sin(pi*t/2)/2;
% trayectoria rapida
```

```
elseif tipo == 2
    qdx = pi*R*sin(pi*t);
    qdy = pi*R*sin(pi*t);
end
% qd''
elseif selector == 3
%   trayectoria lenta
    if tipo == 1
        qdx = pi^2*R*cos(pi*t/2)/4;
        qdy = pi^2*R*cos(pi*t/2)/4;
%   trayectoria rapida
    elseif tipo == 2
        qdx = pi^2*R*cos(pi*t);
        qdy = pi^2*R*cos(pi*t);
    end
end
end
```

