

A Distributed Model Predictive Control (D-MPC) for modular robots in chain configuration

T.G. 1016

**RAÚL FERNANDO CORTES VILLAMIL
DIEGO FERNANDO LINARES BAQUERO**



PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
BOGOTÁ D.C.

2011

A Distributed Model Predictive Control (D-MPC) for modular robots in chain configuration

RAÚL FERNANDO CORTES VILLAMIL
DIEGO FERNANDO LINARES BAQUERO

Trabajo de grado para optar al título de
Ingeniero Electrónico

Directores:

ING. KAMILO ANDRÉS MELO BECERRA M.Sc.
ING. DIEGO ALEJANDRO PATIÑO GUEVARA M.Sc. PhD

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA

2011

Advertencia

“La Universidad Javeriana no se hace responsable de los conceptos emitidos por sus alumnos en sus trabajos de tesis. Solo velará porque no se publique nada contrario al dogma y la moral católica y porque la tesis no contenga ataques o polémicas puramente personales; antes bien, se vea en ella el anhelo de buscar la verdad y la justicia”.

Reglamento de la Pontificia Universidad Javeriana, Artículo 23, de la Resolución 13, de Junio de 1946

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

INGENIERÍA ELECTRÓNICA

RECTOR MAGNÍFICO:	JOAQUÍN EMILIO SÁNCHEZ GARCÍA S.J
DECANO ACADÉMICO:	ING. FRANCISCO J. REBOLLEDO MUÑOZ
DECANO DEL MEDIO UNIVERSITARIO:	PADRE SERGIO BERNAL RESTREPO S.J
DIRECTOR DE CARRERA:	ING. JUAN MANUEL CRUZ BOHÓRQUEZ
DIRECTORES DEL PROYECTO:	ING. KAMILO ANDRES MELO BECERRA M.S.c. ING. DIEGO A. PATIÑO GUEVARA M.Sc. PhD

Tabla de Contenido

1.	INTRODUCCIÓN	10
1.1	OBJETIVO GENERAL	10
1.2	OBJETIVOS ESPECIFICOS	10
2.	MARCO TEÓRICO	12
2.1	CADENAS CINEMÁTICAS.....	12
2.2	EULER –LAGRANGE [4]	12
2.3	CONTROL PREDICTIVO MPC [5]	13
2.3.1	Control predictivo generalizado GPC	14
2.4	CONTROL DISTRIBUIDO	14
3.	MODELO DINÁMICO	15
3.1	EULER-LAGRANGE.....	15
3.2	MODELO IMPLEMENTADO CON SIMMECHANICS.....	17
3.3	VALIDACIÓN DEL MODELO	20
4.	CONTROL PREDICTIVO	26
4.1	GPC SIN PERTURBACIONES ($dk = 0$).....	26
4.2	GPC CON PERTURBACIONES ($dk \neq 0$).....	30
4.3	UMPC (Unconstrained Model Predictive Control).....	33
4.3.1	Conexión del controlador a la planta e identificación del tipo de señales con las que está interactuando.....	34
4.3.2	Reconocimiento e Identificación de la planta por parte del Controlador	35
4.3.3	Comparación de resultados por parte del Toolbox.....	35
4.3.4	Ajuste de los parámetros del controlador	36
4.3.5	Diseño del controlador	38
5.	CONTROL DISTRIBUIDO	41
5.1	CONTROL GPC DISTRIBUIDO 2 MÓDULOS	41
5.2	CONTROL UMPC DISTRIBUIDO 2 MÓDULOS	43
5.3	CONTROL UMPC DISTRIBUIDO 6 MÓDULOS	44
6.	PRUEBAS DE DESEMPEÑO.....	49
6.1	DESEMPEÑO PARA DIFERENTES HORIZONTES DE PREDICCIÓN	49
6.1.1	Control GPC y un solo módulo modelo teórico	49
6.1.2	UMPC	50
6.2	DESEMPEÑO CONTROLADOR CON DIFERENTES PERTURBACIONES	52

6.2.1	Modelo teórico y control GPC con perturbaciones tipo escalón en la salida de posición	52
6.2.2	Modelo mecánico y control GPC con perturbaciones tipo escalón en la salida de velocidad.....	53
6.2.3	Modelo mecánico y control GPC con perturbaciones de valor constante en la salida de posición	55
6.3	DESEMPEÑO CONTROL DISTRIBUIDO PARA DIFERENTES HORIZONTES DE PREDICCIÓN.....	56
6.3.1	Desempeño control GPC distribuido 2 módulos y diferentes horizontes de predicción	56
6.3.2	Desempeño control UMPC distribuido a 2 módulos y diferentes horizontes de predicción.....	58
6.4	DESEMPEÑO DEL CONTROLADOR UMPC VARIANDO LOS PESOS DE SALIDA Y LOS PESOS DE ENTRADA.....	60
6.4.1	Control UMPC con diferentes pesos de entrada	60
6.4.2	Control UMPC con diferentes pesos de Salida	62
7.	CONCLUSIONES	64
8.	BIBLIOGRAFÍA.....	66

Lista de Figuras

Figura 1. Cadena Cinemática compuesta por Seis Links y Cinco Joints	12
Figura 2. Estructura Básica del Control Predictivo MPC	14
Figura 3. Modelo dinámico de un solo módulo.....	16
Figura 4. Body, Revolute, Joint Sensor, Joint Actuator	17
Figura 5. Bloque Body de SimMechanics.....	17
Figura 6. Módulo base.....	17
Figura 7. Bloque Revolute de SimMechanics.....	19
Figura 8. Bloque Joint Actuator de SimMechanics.....	19
Figura 9. Bloque Joint Sensor de SimMechanics.....	19
Figura 10. Representación del Módulo Base usando SimMechanics.....	20
Figura 11. Posición Modelo Teórico y Mecánico	20
Figura 12. Posición Modelo Mecánico	21
Figura 13. Posición Modelo Teórico.....	21
Figura 14. Velocidad Angular Modelo Teórico y Modelo Mecánico	22
Figura 15. Aceleración Angular Modelo Teórico y Modelo Mecánico	22
Figura 16. Posición Modelo Teórico y Mecánico Entrada Sinusoidal	23
Figura 17. Posición Modelo Teórico Entrada Sinusoidal.....	23
Figura 18. Posición Modelo Mecánico Entrada Sinusoidal	24
Figura 19. Velocidad Angular Modelo Teórico y Modelo Mecánico Entrada Sinusoidal	24
Figura 20. Aceleración Angular Modelo Teórico y Modelo Mecánico Entrada Sinusoidal	24
Figura 21. Salida de Posición con Control GPC Para modelo Teórico y mecánico.....	28
Figura 22. Salida de Velocidad con Control GPC Para modelo Teórico y mecánico.....	29
Figura 23. Señal de Control GPC para el Modelo Mecánico y para el Modelo Teórico	29
Figura 24. Diagrama del sistema de control en lazo cerrado con el Observador de Estados.....	31
Figura 25. Comparación Posición control GPC con una Referencia igual 1 y una Perturbación igual a 0.5.....	31
Figura 26. Ampliación Comparación Posición con una Referencia igual 1 y una Perturbación igual a 0.5.....	32
Figura 27. Comparación Velocidad Angular control GPC con una Referencia igual 0 y una perturbación de Posición igual a 0.5	32
Figura 28. Diagrama Control UMPC usando el Modelo Mecánico.....	34
Figura 29. Control UMPC. Clasificación Variables de Entrada y Salida de la Planta.....	34
Figura 30. Secuencia de pasos de inicialización del MPC Controller.....	35
Figura 31. Representación del sistema que es identificado por el toolbox	36
Figura 32. Resultados de la linealización del toolbox.....	36
Figura 33. Perturbaciones a la salida de la planta	37
Figura 34. Perturbaciones a la entrada de la planta	37
Figura 35. Ruido Medido a la salida de la planta.....	38
Figura 36. Control UMPC. Salidas de posición, velocidad y aceleración para una entrada paso.....	38
Figura 37. Control UMPC. Salida de posición en respuesta a una entrada paso	39
Figura 38. Señal de Control GPC para el Modelo Mecánico y para el Modelo Teórico	39
Figura 39. Control UMPC. Respuesta a perturbaciones NO medidas	40
Figura 40. Control GPC Distribuido Para Dos Módulos.....	41
Figura 41. Posición y Velocidad Para el Módulo1.....	42
Figura 42. Posición y Velocidad Para el Módulo 2.....	42

Figura 43. Diagrama de bloques para dos módulos con control UMPC	43
Figura 44. Gráficas de Posición, Velocidad y Aceleración del Primer Módulo	43
Figura 45. Gráficas de Posición, Velocidad y Aceleración del Segundo Módulo	44
Figura 46. Diagrama de bloques para 6 Módulos	45
Figura 47. Gráficas de posición para cada módulo	46
Figura 48. Diagrama de bloques con ajuste de ganancia en cada controlador	47
Figura 49. Gráficas de posición para cada módulo luego de Ajustar la Ganancia	48
Figura 50. Posición Modelo Teórico Control GPC para Diferentes Horizontes de Predicción	49
Figura 51. Velocidad Modelo Teórico Control GPC para Diferentes Horizontes de Predicción	50
Figura 52. Gráficas de Posición Modelo Mecánico usando Control UMPC con diferentes Horizontes de Predicción	51
Figura 53. Gráficas de Velocidad Modelo Mecánico usando Control UMPC con diferentes Horizontes de Predicción	51
Figura 54. Gráficas de Aceleración Modelo Mecánico usando Control UMPC con diferentes Horizontes de Predicción	51
Figura 55. Gráfica Posición con Perturbación Posición Amplitud 1	52
Figura 56. Ampliación Gráfica Posición con Perturbación Posición Amplitud 1	53
Figura 57. Gráfica Velocidad Perturbación Posición Amplitud 1	53
Figura 58. Posición con Perturbación Velocidad Amplitud 0.3	54
Figura 59. Ampliación Posición con Perturbación Velocidad Amplitud 0.3	54
Figura 60. Velocidad con Perturbación Velocidad Amplitud 0.3	55
Figura 61. Posición con Perturbación Constante Amplitud 0.5	55
Figura 62. Ampliación Posición con Perturbación Constante Amplitud 0.5	56
Figura 63. Gráfica Posición Módulo 1 Para diferentes Horizontes de Predicción	56
Figura 64. Gráfica Velocidad Módulo 1 Para diferentes Horizontes de Predicción	57
Figura 65. Gráfica Posición Módulo 2 Para diferentes Horizontes de Predicción	57
Figura 66. Gráfica Velocidad Módulo 2 Para diferentes Horizontes de Predicción	58
Figura 67. Salida de Posición Módulo 1 Control UMPC y diferentes horizontes de predicción	58
Figura 68. Salida de Velocidad Módulo 1 Control UMPC y diferentes horizontes de predicción	59
Figura 69. Salida de Posición Módulo 2 Control UMPC y diferentes horizontes de predicción	59
Figura 70. Salida de Velocidad Módulo 1 Control UMPC y diferentes horizontes de predicción	59
Figura 71. Módulo 1: Señal de control con varios horizontes de predicción	60
Figura 72. Módulo 1: Señal de control con varios horizontes de predicción	60
Figura 73. Salida de posición Control UMPC diferentes pesos de Entrada	61
Figura 74. Señal de Control UMPC para diferentes pesos de Entrada	61
Figura 75. Salida de posición Control UMPC diferentes pesos de Salida	62
Figura 76. Control UMPC distribuido a seis módulos con los mejores parámetros de simulación	63

Lista de Tablas

Tabla 1. Equivalencias Ecuaciones Modelo Dinámico	15
Tabla 2. Tabla detallada de masa y volumen del módulo	18
Tabla 3. Parámetros Tren de Pulsos	20
Tabla 4. Error, varianza y desviación estándar Velocidad y Aceleración Angular.....	22
Tabla 5. Parámetros Señal Sinusoidal	23
Tabla 6. Error, varianza y covarianza Velocidad y Aceleración Angular Entrada Sinusoidal.....	25
Tabla 7. Equivalencias Ecuaciones Control Predictivo.....	26
Tabla 8. Error Absoluto y Varianza en la salida del Controlador para los modelos Mecánico y Teórico.	29
Tabla 9. Parámetros Escogidos Para el diseño del controlador UMPC	38
Tabla 10. Valores Obtenidos con el Control UMPC distribuido para 6 Módulos.....	48
Tabla 11. Desempeño del Controlador GPC Para diferentes Horizontes De Predicción	50
Tabla 12. Desempeño del Controlador UMPC Para diferentes Horizontes De Predicción	52
Tabla 13. Desempeño del controlador UMPC variando los pesos de entrada.	61
Tabla 14. Desempeño del controlador UMPC variando los pesos de entrada.	62

1. INTRODUCCIÓN

Los prototipos robóticos usados en labores de búsqueda y rescate han tratado de imitar el movimiento de algunos animales como serpientes, gusanos entre otros [1]. De esta manera, si se quieren diseñar robots que sean muy parecidos a la realidad, es necesario elaborar modelos matemáticos que permitan analizar de una forma práctica y eficaz el comportamiento de algunos seres vivos. Se debe resaltar, que estos modelos deben ser lo más sencillos posibles de tal forma que su implementación resulte fácil y económica.

Una vez se modela el sistema, es necesario incluir un controlador que permita al robot ejecutar los movimientos deseados lo más cercanos posibles a la realidad. Como dichos movimientos son conocidos, la implementación de controladores que hagan uso del control predictivo resulta de gran utilidad debido a que las referencias futuras son conocidas y las salidas actuales pueden ser estimadas haciendo uso de las salidas futuras.

Este informe final describe los aspectos más importantes sobre el modelado de una cadena cinemática así como la implementación de un control predictivo distribuido el cual se aplica a cada uno de los módulos que componen a la cadena robótica. En este orden de ideas, la obtención del modelo resulta ser una de las partes más importantes, no solo por ser el primer paso de todo este proceso sino porque de su complejidad dependerá la complejidad del controlador y por consiguiente de su implementación.

Como resultado de esto, se logra modelar y validar un módulo de la cadena cinemática con ayuda del toolbox SimMechanics, perteneciente a Simulink® el cual arroja resultados muy cercanos a la realidad, lo que nos permite concluir que para sistemas más complejos puede ser de gran ayuda reduciendo los tiempos de ejecución.

El control predictivo que es diseñado posteriormente, hace uso del modelo obtenido y permite observar el comportamiento del sistema para distintas señales de referencia así como para distintas perturbaciones que en el mundo real podrían llegar a afectar dicho controlador.

Por último, con ayuda de SimMechanics se construye una cadena compuesta por seis módulos aplicando a cada uno de ellos el control diseñado para un solo módulo y se observa cómo el mismo control trabaja de forma independiente para lograr un objetivo común que podría llegar a ser una trayectoria en particular.

Es importante anotar que cada uno de los desarrollos anteriores han servido para ganar experiencia sobre un tema que ha sido poco trabajado en la universidad, abriendo la puerta a futuros estudiantes para que hagan uso de estos resultados que les permitan desarrollar nuevas versiones y diferentes controles aplicados a otro tipo de sistemas que no sean solo robóticos.

A continuación se presentan los objetivos del proyecto:

1.1 OBJETIVO GENERAL

Diseñar un controlador predictivo distribuido aplicado a una plataforma robótica modular en configuración de cadena.

1.2 OBJETIVOS ESPECIFICOS

- Establecer un modelo dinámico en variables de estado del módulo base de la cadena articulada y de la interacción entre módulos.
- Evaluar estrategias de control predictivo (MPC) sobre el modelo obtenido tales como GPC¹ entre otras, y seleccionar la más adecuada según el desempeño del controlador sujeto a criterios de tiempo de cálculo, memoria y desempeño del controlador.

¹ Generalized Predictive Control

- Sintetizar un controlador predictivo-distribuido D-MPC para controlar una cadena robótica de 6 módulos (6 DOF²).
- Evaluar el desempeño del controlador distribuido mediante simulaciones, teniendo en cuenta tiempos de establecimiento, sobrepico y en general índices de desempeño que se escogerán en el desarrollo del proyecto.

² Degree of Freedom

2. MARCO TEÓRICO

Son varios los conceptos que deben tener en cuenta a la hora de realizar un proyecto de esta naturaleza. El primero de ellos hace referencia a las cadenas cinemáticas y cómo pueden ser modeladas en el espacio de estados. De igual manera, se debe tener un conocimiento general acerca del control predictivo y del control distribuido que será aplicado a una plataforma robótica modular, que para este caso en particular, se encuentra en configuración de cadena.

2.1 CADENAS CINEMÁTICAS

Se considera una cadena cinemática al conjunto de enlaces (links) interconectados por medio de articulaciones (joints). Estas articulaciones pueden ser muy simples como la articulación rotacional (revolute joint) o la articulación prismática o deslizante (prismatic joint), ó pueden ser consideradas más complejas como la articulación en forma de esfera (ball joint). Una articulación se considera simple o compleja de acuerdo a sus grados de libertad³, por ejemplo, las articulaciones rotacionales tienen un solo grado de libertad mientras que las articulaciones en forma de esfera tienen 2 grados de libertad. Las acciones de las articulaciones con un solo grado de libertad se pueden describir por medio de un número real, para el caso de las articulaciones rotacionales es el ángulo de rotación y para el caso de las articulaciones deslizantes es el desplazamiento de la misma.

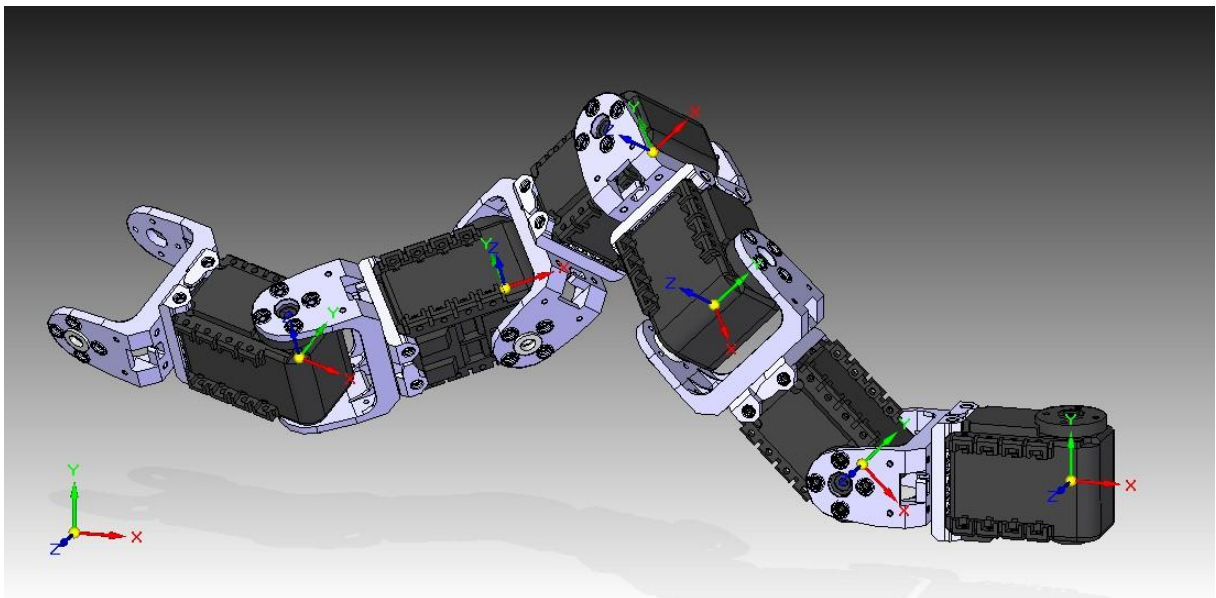


Figura 1. Cadena Cinemática compuesta por Seis Links y Cinco Joints⁴

En conjunto, lo anterior se considera como un robot manipulador [2] el cual cuenta con n articulaciones y $n+1$ enlaces. Por ejemplo en la figura 1 se observa una cadena cinemática con 6 enlaces y cinco articulaciones. A cada articulación le es asociada una variable como el ángulo de rotación y partir de estas variables se puede describir un sistema usando métodos como el de Euler-Lagrange. Se debe aclarar que existen robots con múltiples grados de libertad, los cuales superan el número máximo de variables de posición y orientación en la cinemática de robots, estos son llamados arreglos cinemáticos multi-modulares o redundantes [3].

2.2 EULER –LAGRANGE [4]

Los modelos matemáticos de sistemas físicos se pueden derivar a partir del análisis de la energía de un sistema sin hacer uso de las leyes de Newton o de leyes de Kirchhoff. Para calcular las ecuaciones de movimiento de Lagrange es necesario definir un sistema de coordenadas generalizado. El sistema de coordenadas generalizado no es más que el grupo de coordenadas independientes que son necesarias

³ Grado de Libertad: número de variables independientes necesarias para definir exactamente la posición de un sistema

⁴ Figura Tomada de [1]

para describir el movimiento de la planta completamente. Cabe resaltar que el número de coordenadas generalizadas necesarias para describir un sistema es igual al número de grados de libertad. El Lagrangiano de un sistema se define como:

$$L = T - U \quad (1)$$

Donde T es la energía cinética y U es la energía potencial de la planta. Un sistema en el cual no se disipa energía, se conoce como sistema conservativo. De esta manera, un sistema mecánico conservativo es el cual donde la energía aparece solo en forma de energía cinética y energía potencial. La ecuación de movimiento de Lagrange para un sistema conservativo aparece a continuación.

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \left(\frac{\partial L}{\partial q_i} \right) = 0 \quad (2)$$

Ahora bien, si a una planta en particular se le aplica una fuerza de entrada, la ecuación de movimiento de Lagrange es de la forma:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \left(\frac{\partial L}{\partial q_i} \right) = Q_i \quad (3)$$

Donde Q_i es la fuerza de entrada correspondiente a la i -ésima coordenada generalizada.

2.3 CONTROL PREDICTIVO MPC [5]

El control predictivo MPC se ha desarrollado considerablemente durante los últimos años. La razón para el éxito de esta clase de control, se puede atribuir al hecho de que el control predictivo MPC es quizás la forma más general de plantear el problema de los procesos de control en el dominio del tiempo. La formulación del control predictivo integra control óptimo, control estocástico, control de procesos con tiempo muerto, control multivariable y futuras referencias cuando están disponibles. Otra ventaja del MPC es que debido a que se usa un horizonte de control finito los procesos no lineales, que tan frecuentemente son encontrados en la industria, pueden ser manejados fácilmente. Aunque el control predictivo ha demostrado ser bastante robusto en la mayoría de las aplicaciones, la robustez y la estabilidad son difíciles de garantizar debido al uso del horizonte de predicción finito. Cabe resaltar, que este tipo de control no es tan popular en la industria, una de las razones para esto es que la implementación requiere algunas complejidades matemáticas.

El control predictivo MPC presenta una serie de ventajas respecto a otros métodos de control entre las cuales se encuentran:

- Se puede usar para controlar una gran variedad de procesos, desde aquellos con una dinámica relativamente simple hasta sistemas con un gran tiempo muerto o sistemas inestables.
- Introduce el control con alimentación hacia adelante (Feedforward) para compensar perturbaciones apreciables.
- Tiene intrínsecamente una compensación para tiempos muertos.
- El controlador resultante es fácil de implementar [5].
- Es muy útil cuando las referencias futuras son conocidas (Robótica).

Todos los algoritmos usados en el control predictivo cuentan elementos comunes como los que se pueden observar en la figura 2 para el cálculo del controlador, tales como la obtención del modelo, la minimización de la función objetivo y el desarrollo de la ley de control.

La obtención del modelo es quizás uno de los pasos más importantes para el diseño del controlador, debido a que de este paso depende la facilidad o complejidad de la implementación del mismo. Este modelo debe ser capaz de capturar toda la dinámica del proceso y también debe permitir el uso de las predicciones que serán calculadas.

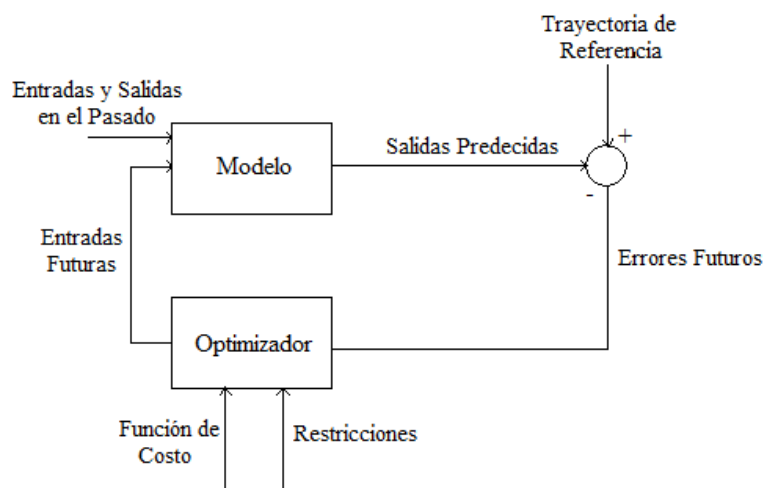


Figura 2. Estructura Básica del Control Predictivo MPC⁵

El objetivo general para esta clase de control es que la salida futura para un horizonte de predicción cualquiera debe seguir a una determinada señal de referencia y al mismo tiempo debe ser aplicada la señal de control. Para llevar todo esto a cabo, se plantea una función objetivo la cual debe ser minimizada para así plantear el sistema de control [6].

2.3.1 Control predictivo generalizado GPC

El control predictivo generalizado se ha convertido en uno de los métodos más populares de MPC, debido a que tiene un gran desempeño y cierto grado de robustez. Esta clase de control puede ser usado con una gran variedad de plantas con un número razonable de variables de diseño, dichas variables se deben especificar por la persona que se encuentre diseñando el controlador de acuerdo a los requerimientos del mismo.

El control predictivo generalizado se puede usar con plantas inestables incorporando el concepto de horizonte de control, así como la consideración de los pesos en la función de costo.

2.4 CONTROL DISTRIBUIDO

La gran ventaja de este tipo de control consiste en lograr un rendimiento óptimo en todo el sistema de control centralizado, mientras se opera en un sistema descentralizado, es decir, los controladores trabajan de forma individual pero cooperativamente para lograr un objetivo común: el control del sistema [7]. El controlador de cada módulo trabaja con las entradas locales basadas en su estado pero sin olvidar la interacción con los módulos inmediatamente vecinos, de esta forma el gasto computacional disminuye y el desempeño aumenta.

Un sistema modular en configuración de cadena necesita dividir el problema centralizado en sub problemas más sencillos, de esta manera, este tipo de sistemas usan un modelo centralizado, un modelo descentralizado y un modelo compuesto. El modelo centralizado modela cada uno de los subsistemas teniendo en cuenta las entradas, salidas y los vectores de estado para cada subsistema. El modelo descentralizado no tiene en cuenta el efecto de los subsistemas externos en el subsistema local mientras que el modelo compuesto sí lo hace [8].

⁵ Figura Tomada de [2]

3. MODELO DINÁMICO

Como se mencionó anteriormente, la obtención del modelo es quizás el paso más importante en el desarrollo de este proyecto. Durante este capítulo, se describirá como fue obtenido el modelo a partir de la formulación de Euler Lagrange y a partir el uso de SimMechanics®. Las variables que son usadas en cada una de las ecuaciones que se plantean en este capítulo se describen en la tabla 1.

Variable	Equivalencias
J_m	Momento de Inercia del Motor
J_L	Momento de Inercia del Módulo
θ_L	Ángulo del Módulo
M	Masa del Módulo
g	Gravedad
l	Longitud al Centro de Masa
θ_m	Ángulo del Eje del Motor
U	Energía Potencial
τ_L	Fuerza Generalizada
r	Radio del Engranaje
u	Torque del Motor reflejado en el Módulo
τ_m	Torque del Motor
B_m	Coefficiente de Amortiguamiento del Motor
B_L	Coefficiente de Amortiguamiento del Módulo

Tabla 1. Equivalencias Ecuaciones Modelo Dinámico

3.1 EULER-LAGRANGE⁶

Por medio de Euler Lagrange se realiza un análisis de la energía el módulo base de la cadena, dicho módulo tiene un grado de libertad.

De esta manera, la energía cinética del módulo base está dada por:

$$K = \frac{1}{2}J_m\dot{\theta}_m^2 + \frac{1}{2}J_L\dot{\theta}_L^2 \quad (4)$$

Se tiene que $\theta_m = r\theta_L$, con $r=1$ es el radio del engranaje, entonces:

$$K = \frac{1}{2}(r^2J_m + J_L)\dot{\theta}_L^2 \quad (5)$$

La energía potencial está dada por:

$$U = Mgl(1 - \cos\theta_L) \quad (6)$$

Se tiene $J = J_m + J_L$, por la definición de Lagrangiano $L= K - U$, entonces:

$$L = \frac{1}{2}J\dot{\theta}_L^2 - Mgl(1 - \cos\theta_L) \quad (7)$$

La ecuación de Euler-Lagrange está definida como:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = F_{ext} \quad (8)$$

Por lo que para este caso en particular la ecuación de Euler-Lagrange queda como:

⁶ Los Subíndices m y L hacen referencia a las variables asociadas al motor y al módulo respectivamente.

$$J\ddot{\theta}_L + Mgl\sin\theta_L = \tau_L \quad (9)$$

Donde τ_L representa fuerzas externas y torques que para este módulo serán:

$$\begin{aligned} \text{Torque motor reflejado en el módulo: } u &= r\tau_m \\ \text{Torques de Amortiguamiento: } B_m\dot{\theta}_m &\text{ y } B_L\dot{\theta}_L \end{aligned}$$

En general se tendrá lo siguiente:

$$\tau = u - B\dot{\theta}_L \quad (10)$$

Donde, $B = rB_m + B_L$

La expresión completa para la dinámica de este sistema está dada por:

$$J\ddot{\theta}_L + B\dot{\theta}_L + Mgl\sin\theta_L = u \quad (11)$$

Para plantear este modelo en Simulink ® se reescribe la ecuación anterior de la siguiente manera:

$$\ddot{\theta}_L = \frac{1}{J}(u - B\dot{\theta}_L - Mgl\sin\theta_L) \quad (12)$$

Teniendo en cuenta esto, el modelo de Simulink ® se presenta en la figura 3:

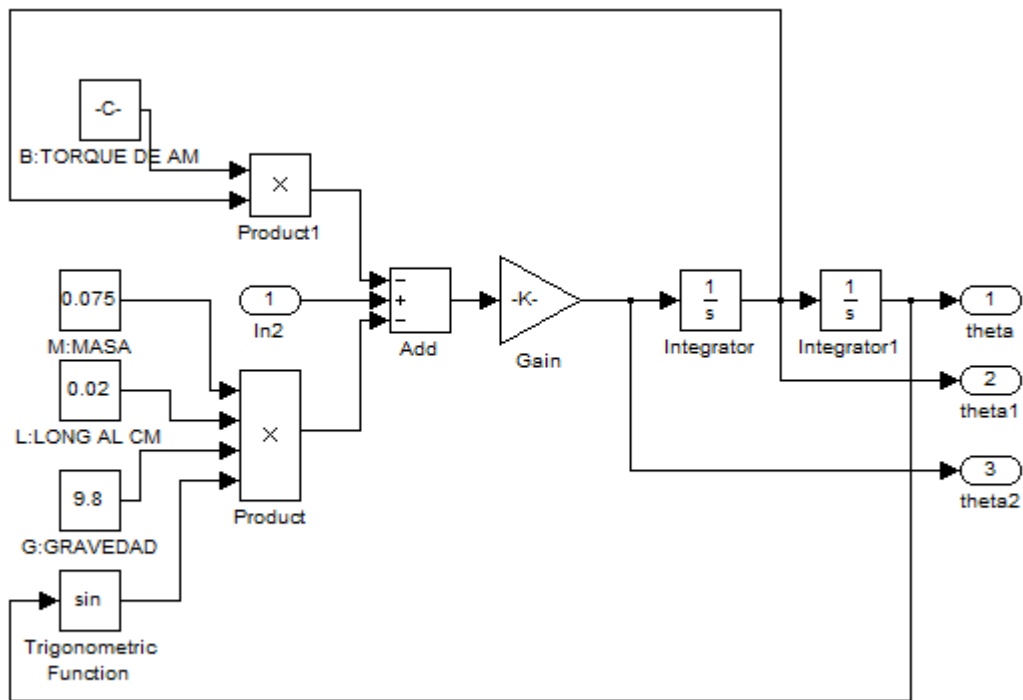


Figura 3. Modelo dinámico de un solo módulo⁷.

Los valores usados para este modelo de acuerdo con la hoja de especificaciones del servomotor Dynamixel AX-12⁸ son:

Masa: 0.075 Kg
Gravedad: 9.8 m/s²
Longitud al centro de Masa =L= 0.02 m (Ver Figura 6)

⁷ Teniendo en cuenta el carácter práctico y funcional de este trabajo de investigación, se utilizaron las especificaciones de los servos *Dynamixel AX-12*. (hoja de especificaciones adjunta)

⁸ Hoja de Especificaciones Dynamixel AX-12. Disponible en: [http://robosavvy.com/RoboSavvyPages/Support/Bioloid/AX-12\(english\).pdf](http://robosavvy.com/RoboSavvyPages/Support/Bioloid/AX-12(english).pdf)

$$\text{Momento de Inercia} = J = \frac{ml^2}{2} = 8 \times 10^{-4} \text{ kg} \cdot \text{m}^2$$

3.2 MODELO IMPLEMENTADO CON SIMMECHANICS

En este capítulo se utiliza la herramienta *SimMechanics*® de *MATLAB*, con la que se realiza una simulación mecánica del primer módulo la cual se compara posteriormente con el modelo teórico obtenido en la sección 3.1.

Los bloques básicos para la simulación mecánica son:

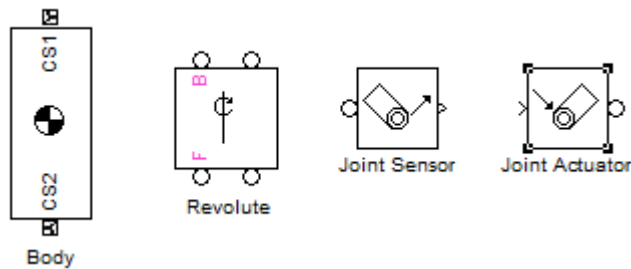


Figura 4. Body, Revolute, Joint Sensor, Joint Actuator

A continuación se explicarán cada uno de los bloques:

Body: Representa un cuerpo rígido (Ver Figura 5) cuyas propiedades se pueden ajustar dependiendo las necesidades, para esta simulación se cambió la masa, el tensor de inercia, la orientación y los sistemas de coordenadas de acuerdo a las características de los servos mencionados anteriormente⁹.

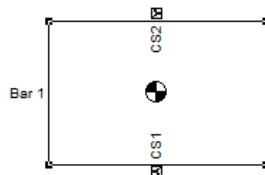


Figura 5. Bloque Body de SimMechanics

En la figura 6 se puede ver de forma más detallada el módulo.

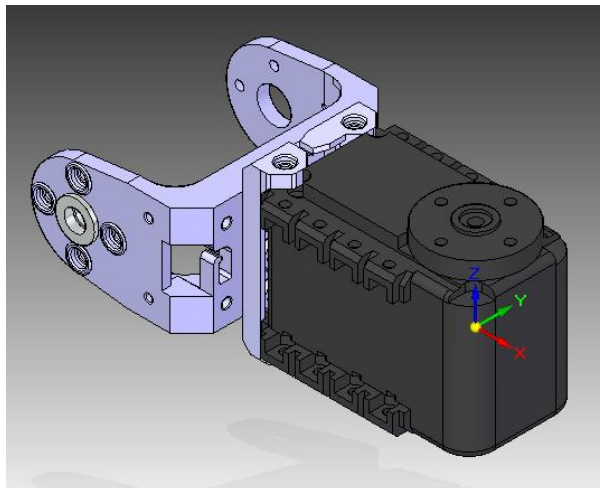


Figura 6. Módulo base

⁹ Además de los servos *Dynamixel*, también se utilizaron más partes del kit *Bioloïd*, de tal forma que la estructura y las propiedades del módulo base son totalmente reales.

Para obtener las características exactas del módulo completo (servos y otras partes) se utiliza un software de sistema de diseño asistido por ordenador (CAD), el cual permite saber las propiedades físicas con exactitud.

En la tabla 2, se observa el volumen de un solo módulo el cual se calcula con ayuda del software de diseño anteriormente mencionado.

Document Name	Up-To-Date	Mass	Volume	Surface Area
ModBase.asm	Yes	0,075 kg	54457,117 mm ³	
AX-12.par	Yes	0,060 kg	43626,109 mm ³	10804,24 mm ²
BIOarandela.par	Yes	0,000 kg	258,200 mm ³	402,12 mm ²
BIOarandelaO.par	Yes	0,000 kg	125,664 mm ³	251,33 mm ²
OF-12S.par	Yes	0,004 kg	3172,435 mm ³	3517,01 mm ²
OF-12SH.par	Yes	0,010 kg	7274,709 mm ³	7518,10 mm ²

Tabla 2. Tabla detallada de masa y volumen del módulo

Además de la masa y las dimensiones del módulo, es necesario calcular el tensor de inercia tal y como se muestra a continuación.

El tensor de inercia, está definido como:

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (13)$$

Las componentes del tensor de inercia de la ecuación (13) se encuentran en una base ortonormal y se calculan a partir de los tres momentos de inercia según esos tres ejes perpendiculares. De esta manera, se asume que ésta es una distribución de masa uniforme por lo que la densidad no varía en las ecuaciones 14, 15 y 16 (Densidad Constante) por lo que sólo es necesario calcular los elementos de la diagonal de la matriz I, estos elementos se calculan de la siguiente manera:

$$I_{xx} = \iiint (y^2 + z^2)\rho(x, y, z)dx dy dz \quad (14)$$

$$I_{xx} = \int_0^{0.047} \int_0^{0.026} \int_0^{0.089} (y^2 + z^2) \frac{0.074 \text{ kg}}{5.4 \times 10^{-5} \text{ m}^3} dx dy dz$$

$$I_{xx} = 1.42 \times 10^{-4}$$

$$I_{yy} = \iiint (x^2 + z^2)\rho(x, y, z)dx dy dz \quad (15)$$

$$I_{yy} = \int_0^{0.047} \int_0^{0.026} \int_0^{0.089} (x^2 + z^2) \frac{0.074 \text{ kg}}{5.4 \times 10^{-5} \text{ m}^3} dx dy dz$$

$$I_{yy} = 4.99 \times 10^{-4}$$

$$I_{zz} = \iiint (x^2 + y^2)\rho(x, y, z) dx dy dz \quad (16)$$

$$I_{zz} = \int_0^{0.047} \int_0^{0.026} \int_0^{0.089} (x^2 + y^2) \frac{0.074 \text{ kg}}{5.4 \times 10^{-5} \text{ m}^3} dx dy dz$$

$$I_{zz} = 4.24 \times 10^{-4}$$

Las integrales de las ecuaciones (14), (15) y (16) se calculan sobre la región del espacio ocupada por el módulo, los elementos de la diagonal del tensor de inercia son llamados momentos de inercia principales alrededor de los ejes x , y y z . Además de esto, se halla la densidad ρ del módulo usando valores de la tabla 2. Las dimensiones que se usan son extraídas de la hoja de especificaciones de los servomotores *Dynamixel AX-12¹⁰*.

Los valores calculados son ingresados a bloque body de SimMechanics.

Revolute: es el segundo bloque fundamental (ver Figura 7), éste representa un grado único de libertad de rotación sobre un eje determinado entre dos cuerpos o en este caso, entre el cuerpo y la tierra; el sentido de rotación está definido por la regla de la mano derecha.

En este bloque sólo se definió el sentido de rotación y el número de entradas y salidas; las entradas de este bloque son la salida del cuerpo anterior a éste y la señal que viene del bloque *Joint Actuator*; las salidas son la entrada del siguiente cuerpo y la señal que va al bloque *Joint Sensor*.

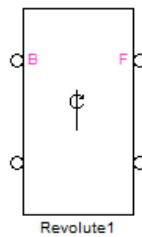


Figura 7. Bloque Revolute de SimMechanics

Joint Actuator: La función de este bloque (Ver Figura 8) es convertir una señal de entrada en una fuerza, por lo tanto este bloque sólo tiene una entrada y una salida; la entrada puede ser una señal sinusoidal, un tren de pulsos, una rampa, entre otros; La salida es la forma de onda convertida en una señal de fuerza a la cual estará conectada al bloque *Revolute*.

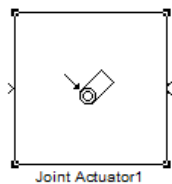


Figura 8. Bloque Joint Actuator de SimMechanics

Joint Sensor: La función de este bloque (Ver Figura 9) es transformar una señal física en una forma de onda que pueda ser vista en un *scope*, al igual que el *Joint Actuator* éste solo tiene una entrada y una salida, la entrada es una de las salidas del bloque *Revolute*, y la salida va conectada a un *Scope*. En este bloque sólo se seleccionan las señales que queremos ver.

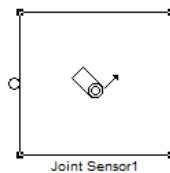


Figura 9. Bloque Joint Sensor de SimMechanics

Después de ver y caracterizar cada uno de los bloques se conformó el modelo del módulo base, el cual estaba conectado de forma paralela con la simulación del modelo teórico. La interconexión de todos los elementos anteriormente mencionados para conformar el modelo del módulo base se ve en la figura 10.

¹⁰ Hoja de Especificaciones Dynamixel AX-12. Disponible en: [http://robosavvy.com/RoboSavvyPages/Support/Bioloid/AX-12\(english\).pdf](http://robosavvy.com/RoboSavvyPages/Support/Bioloid/AX-12(english).pdf)

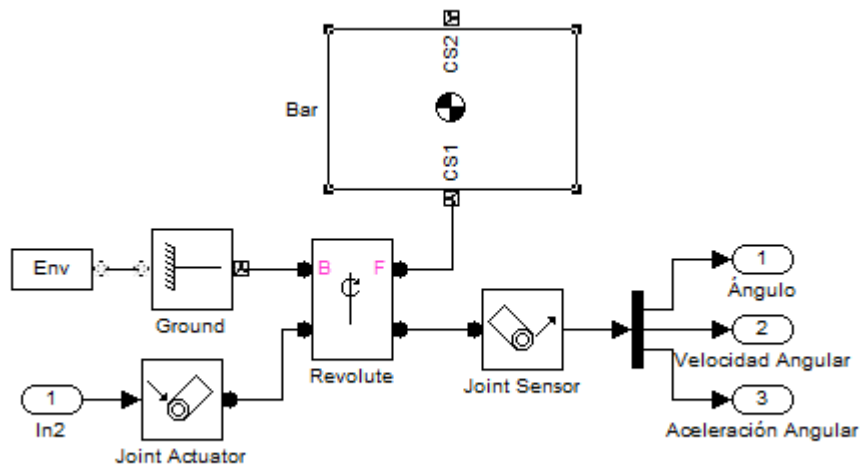


Figura 10. Representación del Módulo Base usando SimMechanics

3.3 VALIDACIÓN DEL MODELO

Entrada Tren de Pulsos: Se simuló el sistema con un tren de pulsos como entrada, de esta manera para el modelo en *SimMechanics*® la señal pulsante está conectada al *Joint Actuator* y para el modelo teórico la señal pulsante es la variable u .

Las características de la señal de entrada son:

Parámetro	Valor
Amplitud	0.1
Periodo	4 s
Ancho del Pulso	50 %

Tabla 3. Parámetros Tren de Pulsos

Se escogió esta señal debido a que es relativamente lenta y los cambios se pueden observar con claridad cada 4 segundos. Las gráficas para el modelo teórico y para el modelo mecánico se pueden observar a continuación.

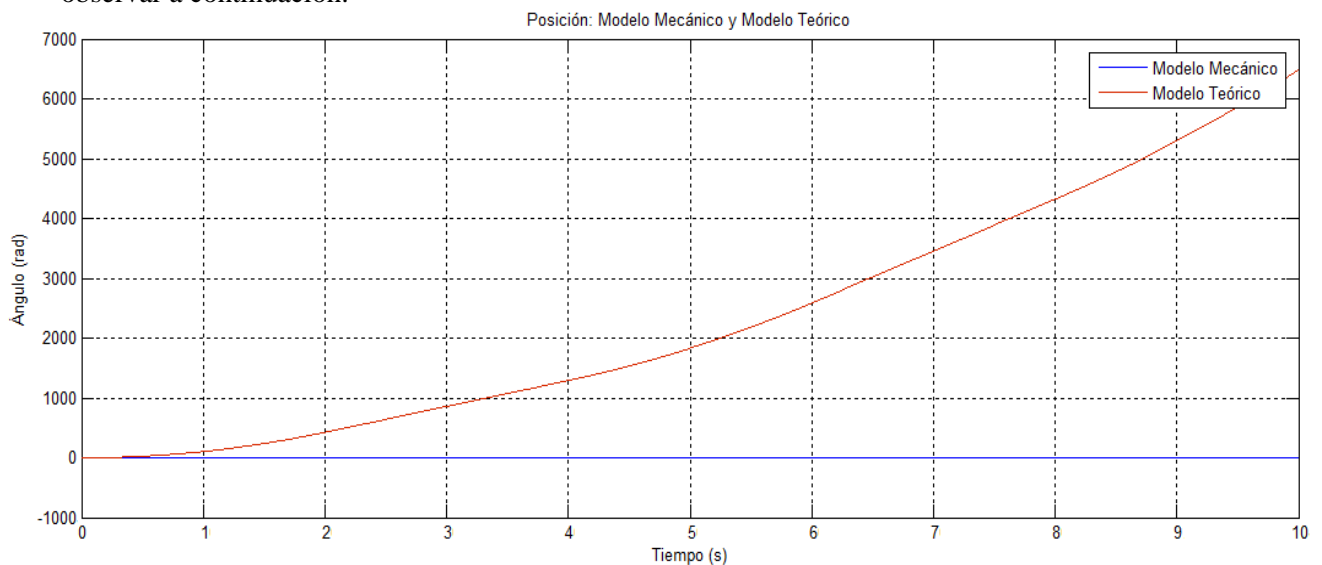


Figura 11. Posición Modelo Teórico y Mecánico

En la figura 11 se pueden observar las gráficas de posición tanto del modelo teórico como para el modelo mecánico. Sin embargo, existe una diferencia en como SimMechanics muestra dicha posición

ya que una vez el ángulo aumenta, SimMechanics lo muestra variando entre 3.14 y -3.14 tal y como se ve en la figura 12.

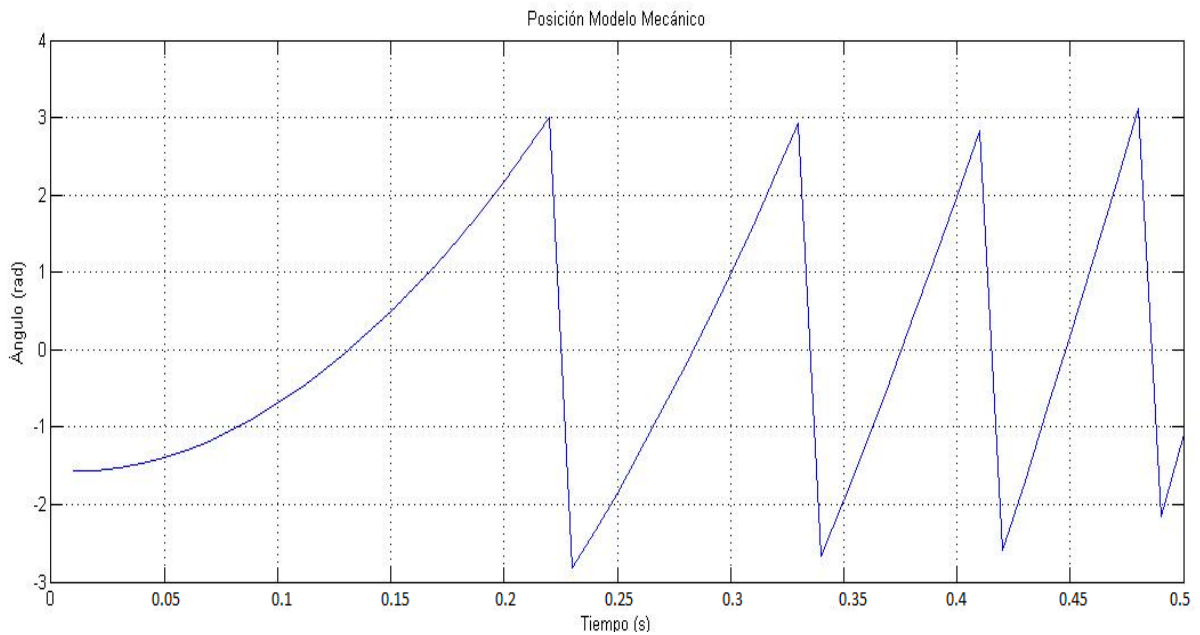


Figura 12. Posición Modelo Mecánico

Para el modelo teórico, la forma de visualización de dicho ángulo es completamente diferente, de esta manera si el módulo gira varias veces sobre su propio eje, Matlab lo identifica como 6.28 rad, 12.56 rad y así sucesivamente, por lo que la gráfica de posición crece a medida que el tiempo lo hace tal y como se ve en la figura 13.

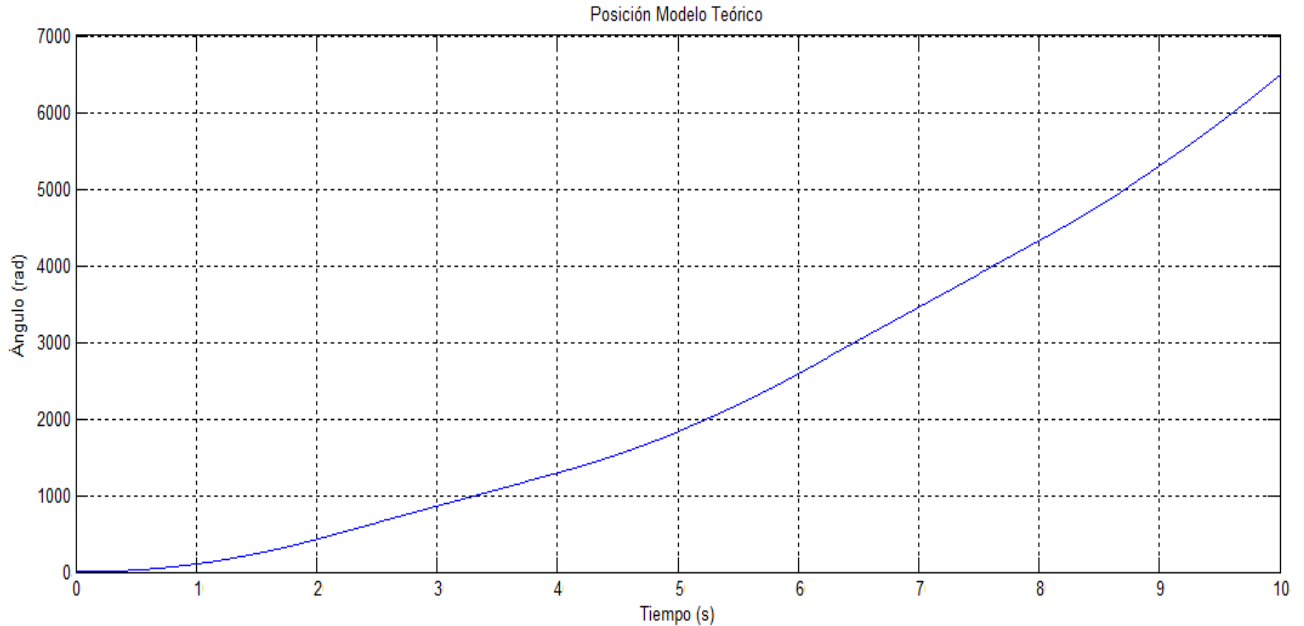


Figura 13. Posición Modelo Teórico

La velocidad angular y la aceleración angular son graficadas con el mismo método tanto en SimMechanics como en la simulación teórica de Matlab, por lo que dichas gráficas si pueden ser observadas en una misma ventana tal y como se ve en las figuras 14 y 15.

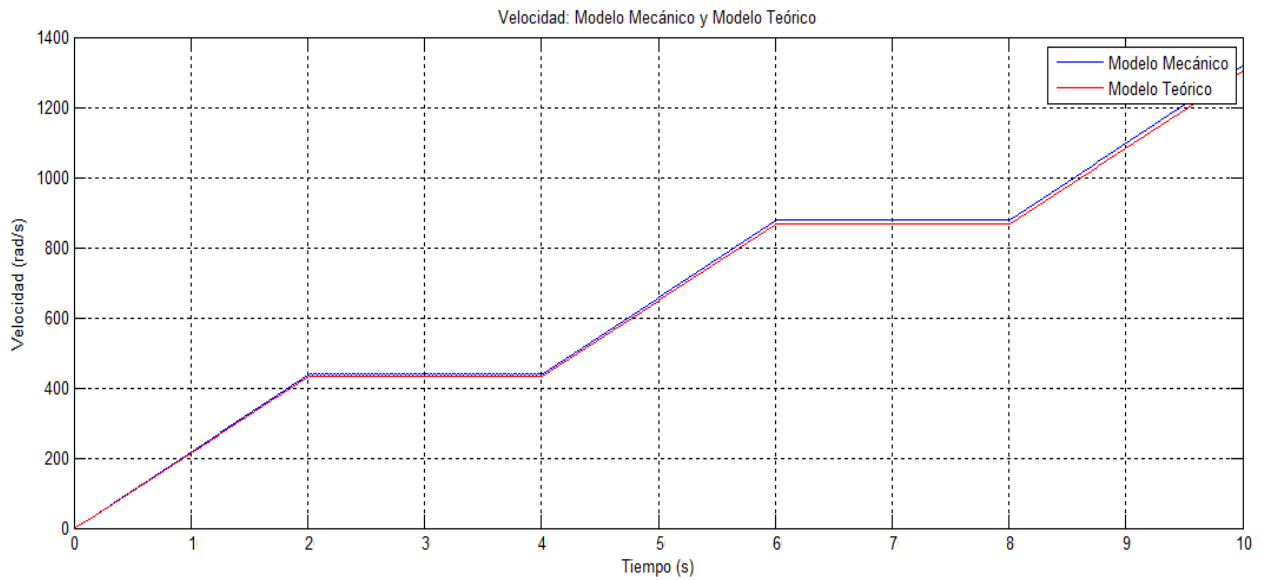


Figura 14. Velocidad Angular Modelo Teórico y Modelo Mecánico

Por último se grafica la aceleración angular obteniendo los resultados de la figura 15.

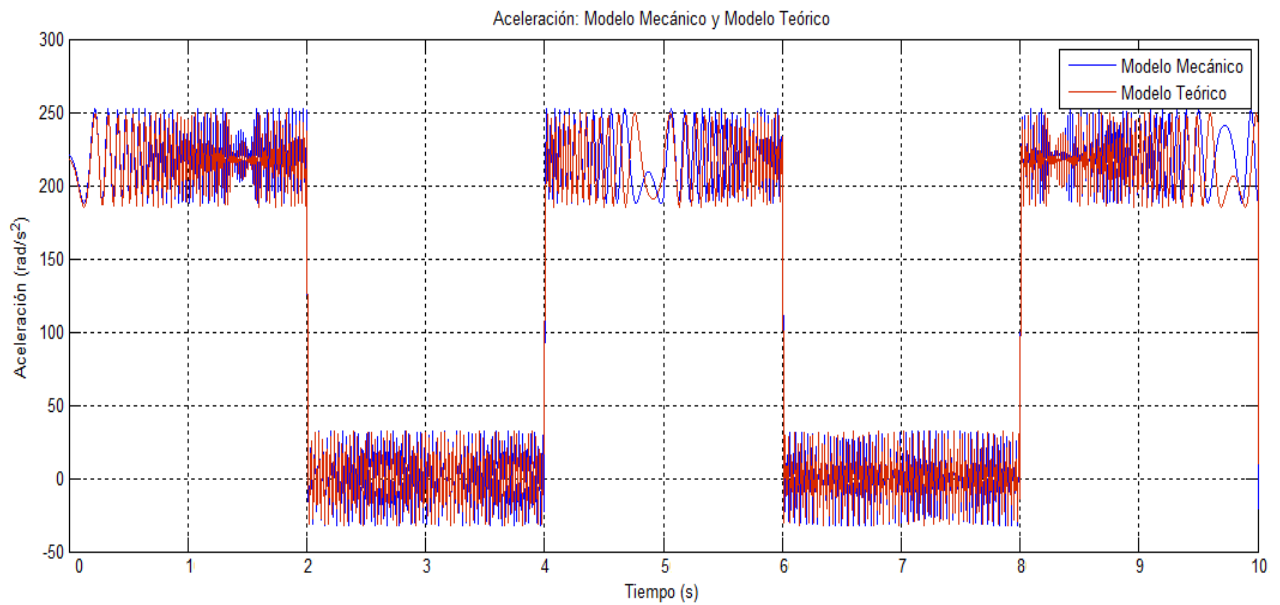


Figura 15. Aceleración Angular Modelo Teórico y Modelo Mecánico

Para comparar los resultados obtenidos en la figura 14 y figura 15, se halla el error absoluto, la varianza y la desviación estándar (Ver Tabla 4).

	Error Absoluto (%)	Varianza	Desviación Estándar
Velocidad	0.64	0.098	0.31
Aceleración	18.61	7.87	2.8

Tabla 4. Error, varianza y desviación estándar Velocidad y Aceleración Angular

Como se observa en la tabla 4, el error¹¹ en la aceleración es mucho mayor respecto al error en la velocidad, esto se debe a que los valores de aceleración cada 2 segundos tienen grandes variaciones lo que hace que el error tenga un valor mayor.

¹¹ El error se calcula usando la siguiente fórmula $Error = \frac{|Valor Teórico - Valor Mecánico|}{Valor Teórico} \times 100$

Entrada Sinusoidal: Se simuló el sistema con una entrada sinusoidal, de esta manera para el modelo en SimMechanics® la señal pulsante está conectada al **Joint Actuador** y para el modelo teórico la señal pulsante es la variable u .

Las características de la señal sinusoidal son:

Parámetro	Valor
Amplitud	1
Frecuencia	1 rad/s

Tabla 5. Parámetros Señal Sinusoidal

Se escogió esta señal debido a que es relativamente lenta y los cambios se pueden observar con claridad cada segundo. Las gráficas para el modelo teórico y para el modelo mecánico se pueden observar a continuación.

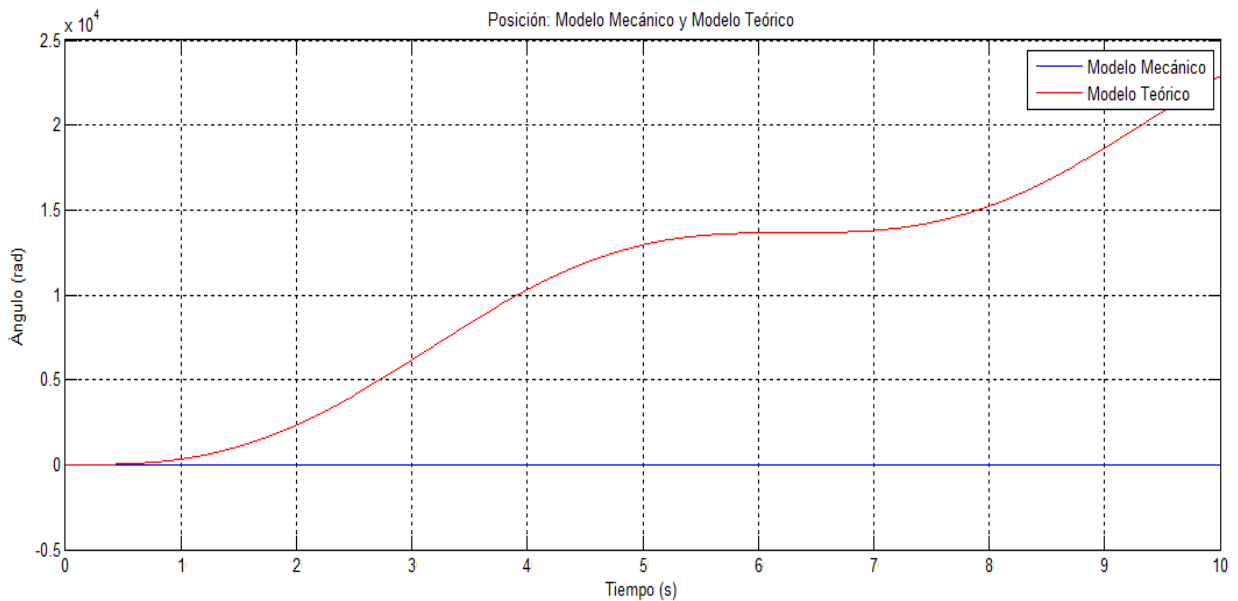


Figura 16. Posición Modelo Teórico y Mecánico Entrada Sinusoidal

En la figura 16 al igual que en la figura 11, las posiciones no pueden ser comparadas en la misma escala debido a la forma como se grafican en cada uno de los modelos. De esta manera, las gráficas de posición para una entrada sinusoidal se muestran por aparte en las figuras 17 y 18.

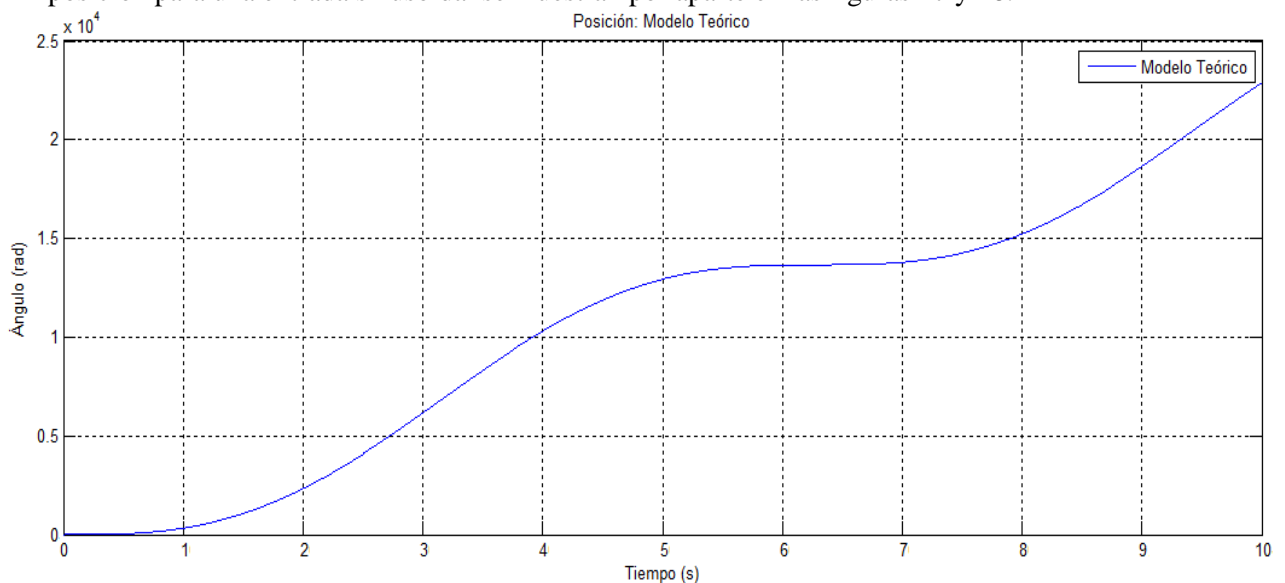


Figura 17. Posición Modelo Teórico Entrada Sinusoidal

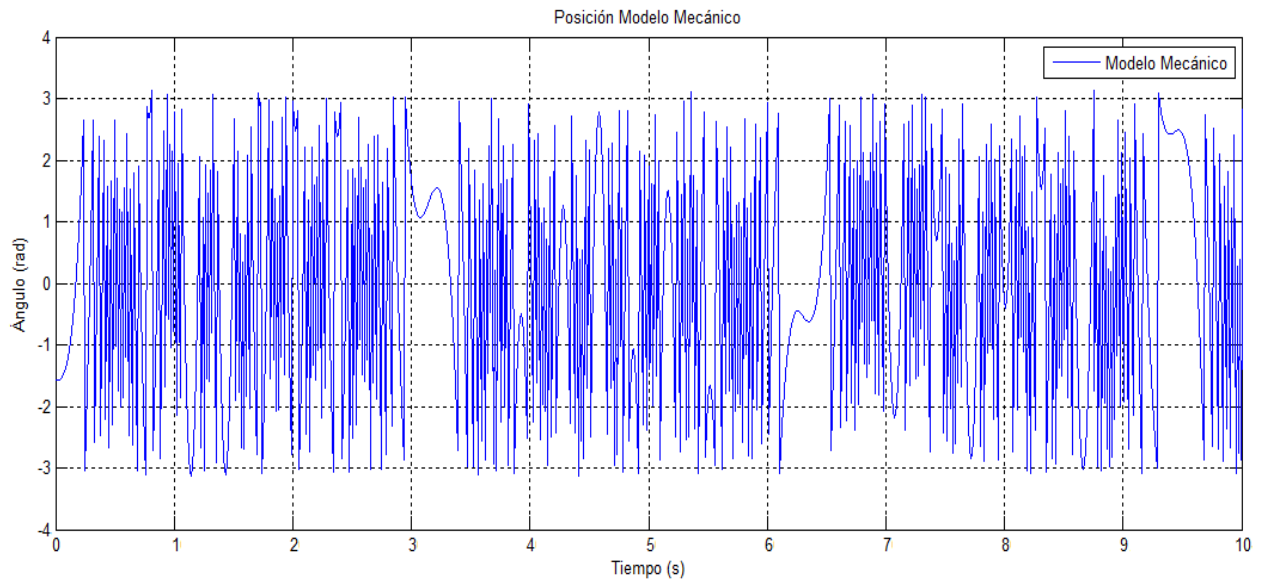


Figura 18. Posición Modelo Mecánico Entrada Sinusoidal

La velocidad angular y la aceleración angular con una entrada sinusoidal se pueden observar en la figura 19 y 20 respectivamente.

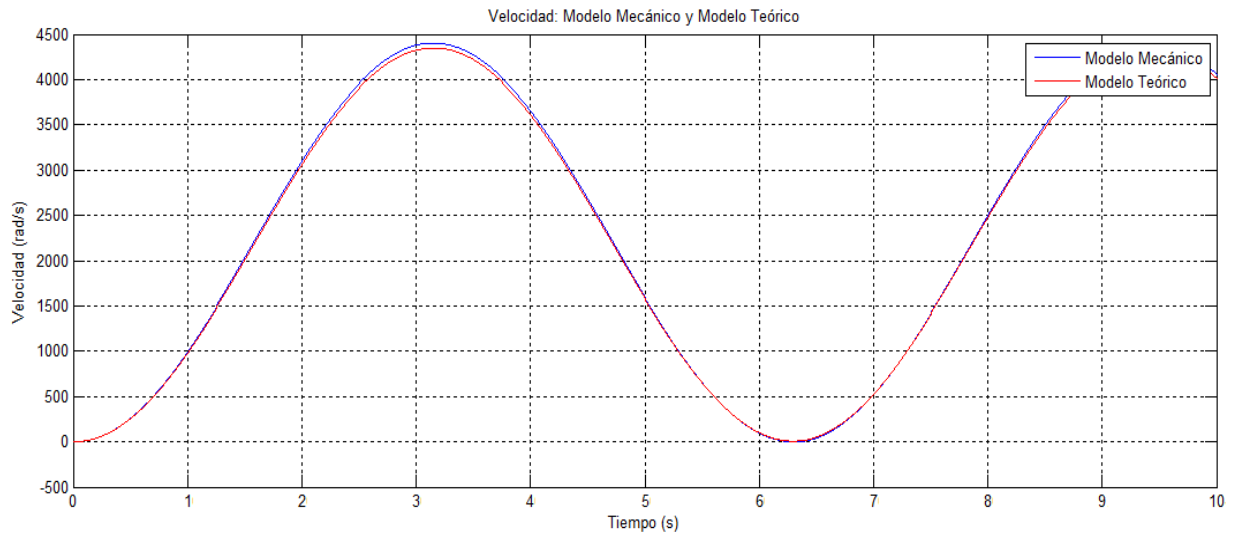


Figura 19. Velocidad Angular Modelo Teórico y Modelo Mecánico Entrada Sinusoidal

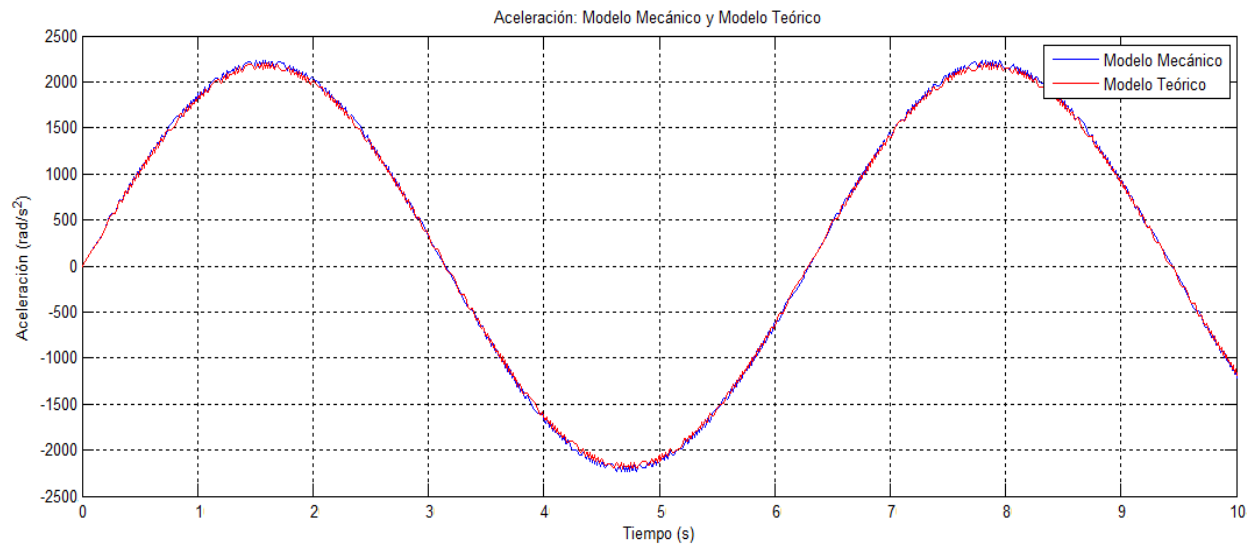


Figura 20. Aceleración Angular Modelo Teórico y Modelo Mecánico Entrada Sinusoidal

Al igual a como se hizo con el tren de pulsos, se calcula el error absoluto, la varianza y la covarianza registrando los resultados en la tabla 6.

	Error Absoluto (%)	Varianza	Desviación Estándar
Velocidad	1.18	0.019	0.14
Aceleración	7.75	0.025	0.159

Tabla 6. Error, varianza y covarianza Velocidad y Aceleración Angular Entrada Sinusoidal

Como se pudo observar en las figuras anteriores, el modelo teórico y modelo mecánico tienen un comportamiento dinámico muy similar, esto fue comprobado posteriormente hallando el error¹² entre las dos medidas, dicho error es relativamente pequeño por lo que se asume que el modelo ha sido validado. De esta manera, en los capítulos posteriores se podrá hacer uso de cualquier modelo para diseñar los controladores con lo que se espera obtener resultados con poca diferencia entre ellos. Sin embargo, para efectos de simulación, se mostraran los resultados tanto para el modelo mecánico como para el modelo teórico. Por último, invitamos al lector a desarrollar sus propios modelos mecánicos con ayuda de SimMechanics®, para que puedan observar la gran utilidad de este software el cual reduce tiempos de simulación y arroja resultados realmente confiables.

¹² El error se calcula usando la siguiente fórmula $Error = \frac{|Valor\ Teórico - Valor\ Mecánico|}{Valor\ Teórico} \times 100$

4. CONTROL PREDICTIVO

En este capítulo, se muestra el desarrollo de los dos tipos de controles con los que se decidió trabajar en este proyecto, el Control predictivo generalizado o GPC y el control predictivo sin restricciones UMPC (Unconstrained Model Predictive Control) diseñado con ayuda del toolbox Model predictive control de Matlab. Las variables que son usadas en cada una de las ecuaciones que se plantean en este capítulo se describen en la tabla 7.

Variable	Equivalencias
A, B, C, D	Matrices Espacio de Estados
u_k	Entradas del Sistema en el instante k
x_k	Vector de Estados en el instante k
y_k	Salida del Sistema en el instante k
r	Referencia
d_k	Perturbación
K	Ganancia Realimentación de Estados
$x_{\rightarrow k}$	Vector de Futuros valores de x_k
H,P,Q	Matrices de Predicción
I	Matriz Identidad
n_y	Horizonte de Predicción
n_u	Horizonte de Control

Tabla 7. Equivalencias Ecuaciones Control Predictivo

4.1 GPC SIN PERTURBACIONES ($d_k = 0$)

Para empezar con el diseño del control predictivo generalizado se debe tener en cuenta que el sistema debe linealizarse y discretizarse por lo que asume un tiempo de muestreo igual a 0.01 s. Se escoge este tiempo de muestreo debido a que describe al sistema sin pérdida de información. De esta manera, el sistema es de la forma:

$$x_{k+1} = Ax_k + Bu_k \quad (17)$$

$$y_k = Cx_k + Du_k + d_k$$

Para un sistema discreto en estado estacionario se sabe que:

$$x_{k+1} = x_k = x_{ss} \quad (18)$$

De esta manera:

$$x_{ss} = Ax_{ss} + Bu_{ss} \quad (19)$$

Así, al despejar x_{ss} de la ecuación (19):

$$x_{ss} = (I - A)^{-1}Bu_{ss} \quad (20)$$

Ahora, si se quiere tener una salida deseada, por ejemplo $y_{ss} = r$ se tiene que:

$$y_{ss} = Cx_{ss} + Du_{ss} \quad (21)$$

$$r = C(I - A)^{-1}Bu_{ss} + Du_{ss} \quad (22)$$

En la ecuación (21) se asume $d_k = 0$, de esta manera, si se quiere seguir una referencia en particular, es necesario saber cuál debe ser el valor de la entrada u, por lo que:

Sea,
$$\tilde{A} = C(I - A)^{-1}B \quad (23)$$

Entonces,
$$r = \tilde{A}u_{ss} + Du_{ss} \quad (24)$$

$$r = (\tilde{A} + D)u_{ss} \quad (25)$$

$$(\tilde{A} + D)^T r = (\tilde{A} + D)^T (\tilde{A} + D)u_{ss} \quad (26)$$

$$u_{ss} = ((\tilde{A} + D)^T (\tilde{A} + D))^{-1} (\tilde{A} + D)^T r \quad (27)$$

Según la hoja de especificaciones del servomotor¹³, la entrada de torque u puede tomar valores hasta de 16.5 kgf.cm o en el SI de 1.62 Nm. Aunque para efectos de simulación se asumirá cualquier valor para dicho torque.

Además, para todas las simulaciones que se presentan en este libro se escoge una referencia $r = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, lo que equivale a un ángulo de 57 grados aproximadamente. Dicha referencia es escogida así debido a que se ajusta con las limitaciones físicas que tendrán los servomotores a la hora de moverse en el mundo real.

Según GPC, la ley de control se define como [6]:

$$u_k - u_{ss} = -e_1^T [H_x^T Q H_x + R]^{-1} H_x^T Q P_{xx} (x - x_{ss}) \quad (28)$$

Donde,

$$e_1^T = [1, 0, 0, \dots, 0] \quad (29)$$

En la ecuación (28) se hace uso de las predicciones para el modelo las cuales están descritas por:

$$x_{\rightarrow k} = P_{xx} x_k + H_x u_{\rightarrow k-1} \quad (30)$$

$$y_{\rightarrow k} = P x_k + H u_{\rightarrow k-1}$$

Y donde:

$$P_{xx} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{ny} \end{bmatrix}, H_x = \begin{bmatrix} B & 0 & 0 \\ AB & B & 0 \\ \vdots & \vdots & \vdots \\ A^{ny-1}B & A^{ny-2}B & B \end{bmatrix} \quad (31)$$

$$P = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{ny} \end{bmatrix} H = \begin{bmatrix} CB & 0 & 0 \\ CAB & CB & 0 \\ \vdots & \vdots & \vdots \\ CA^{ny-1}B & CA^{ny-2}B & CB \end{bmatrix}$$

Usando el desarrollo teórico del capítulo 1 en donde el módulo base de la cadena está descrito por las siguientes matrices:

$$A = \begin{bmatrix} 1 & 0.01 \\ 0.004 & 1 \end{bmatrix}; B = \begin{bmatrix} 0.1111 \\ 22.22 \end{bmatrix}; C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0.4047 & 0 \end{bmatrix}; D = \begin{bmatrix} 0 \\ 0 \\ 2222.2 \end{bmatrix} \quad (32)$$

Se calculó la ley de control tomando $n_y = 4$ y $n_u = 2$, por lo que las matrices de predicción son:

¹³ Hoja de Especificaciones Dynamixel AX-12. Disponible en: [http://robosavvy.com/RoboSavvyPages/Support/Bioloid/AX-12\(english\).pdf](http://robosavvy.com/RoboSavvyPages/Support/Bioloid/AX-12(english).pdf)

$$P_{xx} = \begin{bmatrix} 1 & 0.01 \\ 0.004 & 1 \\ 1 & 0.02 \\ 0.008 & 1 \\ 1 & 0.03 \\ 0.012 & 1 \\ 1 & 0.04 \\ 0.016 & 1 \end{bmatrix}; H_x = \begin{bmatrix} 0.11 & 0 \\ 22.22 & 0 \\ 0.33 & 0.11 \\ 22.22 & 22.22 \\ 0.55 & 0.33 \\ 22.22 & 22.22 \\ 0.77 & 0.55 \\ 22.22 & 22.22 \end{bmatrix}; P = \begin{bmatrix} 1 & 0.01 \\ 0.004 & 1 \\ 0.4 & 0.004 \\ 1 & 0.02 \\ 0.008 & 1 \\ 0.4 & 0.008 \\ 1 & 0.03 \\ 0.012 & 1 \\ 0.4 & 0.012 \\ 1 & 0.04 \\ 0.016 & 1 \\ 0.404 & 0.016 \end{bmatrix}; H = \begin{bmatrix} 0.111 & 0 \\ 22.22 & 0 \\ 0.045 & 0 \\ 0.33 & 0.11 \\ 22.22 & 22.22 \\ 0.134 & 0.04 \\ 0.55 & 0.33 \\ 22.22 & 22.22 \\ 0.224 & 0.134 \\ 0.77 & 0.55 \\ 22.22 & 22.22 \\ 0.314 & 0.224 \end{bmatrix} \quad (33)$$

Comúnmente se escoge $Q = C^T C$ [4] y $R = \lambda I$, donde λ es la matriz de pesos e I es la matriz identidad. De esta manera, la ganancia por realimentación de estados es:

$$K = -e_1^T [H_x^T Q H_x + R]^{-1} H_x^T Q P_{xx} = [-0.0027 \quad -0.0449] \quad (34)$$

Con estos valores se implemento la ley de control por realimentación de estados tal y como se ve en la ecuación (35).

$$u = -K(x - x_{ss}) + u_{ss} \quad (35)$$

Al simular el sistema de control con una referencia $r = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, se obtienen los siguientes resultados:

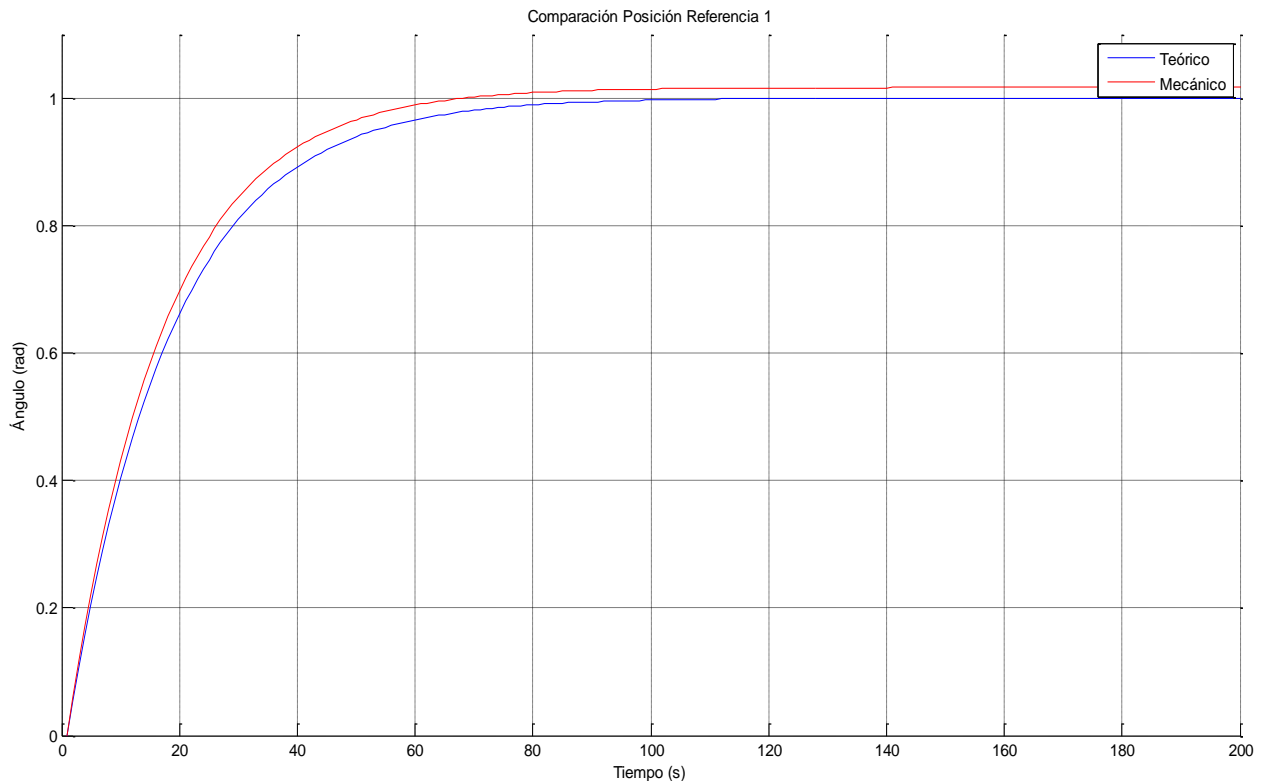


Figura 21. Salida de Posición con Control GPC Para modelo Teórico y mecánico

En la figura 21, se grafica el ángulo del controlador cuando se quiere que siga una referencia igual a 1. Como se puede ver, esta referencia se alcanza luego de 100 segundos por lo que es necesario mejorar el tiempo de respuesta. Este análisis se llevará a cabo en el capítulo 6 donde se realizaran pruebas de desempeño a los controladores. El error absoluto y la varianza para las gráficas 21 y 22 se resume en la tabla 8.

	Error Absoluto (%)	Varianza
Ángulo	1.85	0.14
Velocidad	5.56	0.37

Tabla 8. Error Absoluto y Varianza en la salida del Controlador para los modelos Mecánico y Teórico.

El error de velocidad es mayor debido a que el modelo mecánico presenta un mayor pico de velocidad respecto al modelo teórico.

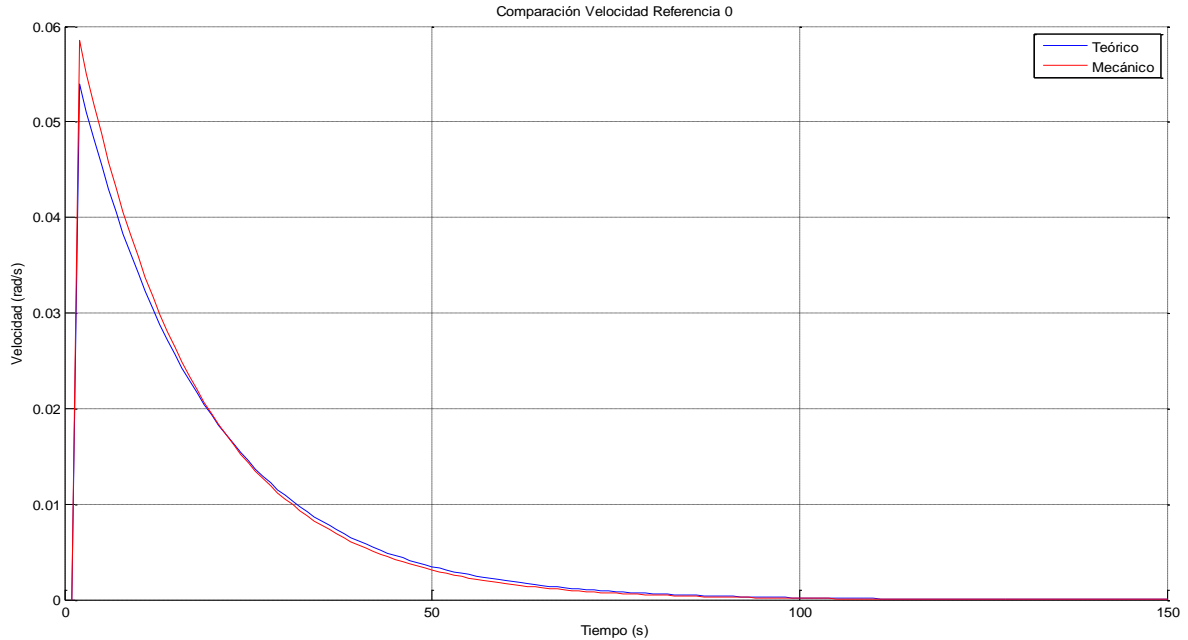


Figura 22. Salida de Velocidad con Control GPC Para modelo Teórico y mecánico

En la figura 22 se observa que la velocidad alcanza el valor cero luego de 100 segundos, esta velocidad tiene un pequeño pico de amplitud 0.06 durante los primeros segundos pero poco a poco alcanza el valor deseado.

La señal de control tanto para el modelo teórico como para el modelo mecánico se puede ver en la figura 23.

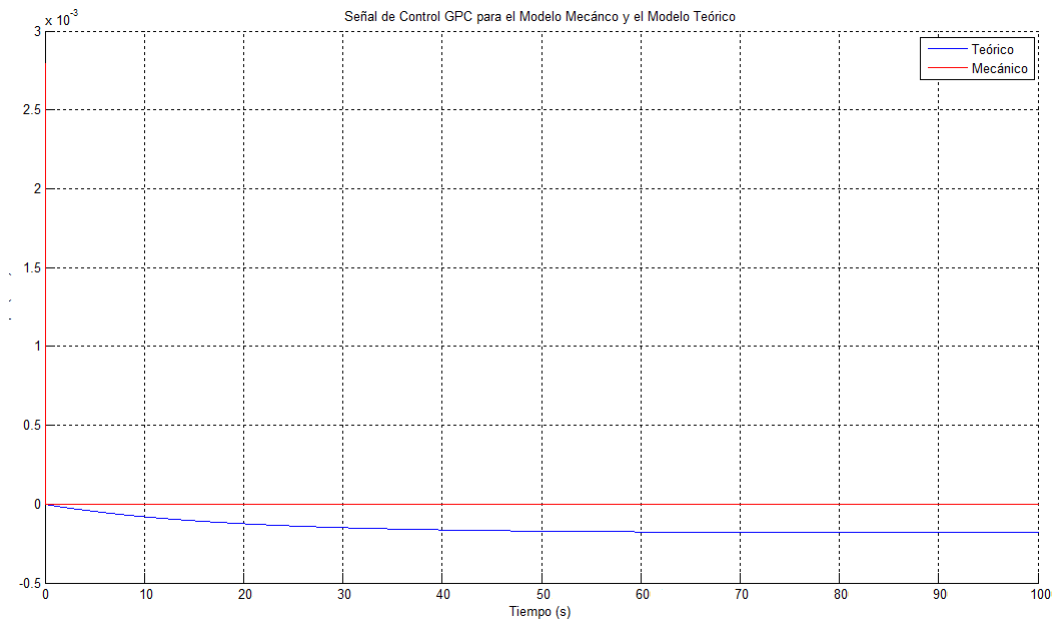


Figura 23. Señal de Control GPC para el Modelo Mecánico y para el Modelo Teórico

A continuación se muestra el desarrollo que fue llevado a cabo cuando el sistema es sometido a una perturbación.

4.2 GPC CON PERTURBACIONES ($d_k \neq 0$)

En el numeral anterior, se asumió que las perturbaciones d_k son iguales a cero. Sin embargo, el controlador debe funcionar correctamente así sea sometido a una perturbación como una constante, una señal escalón entre otras. Es así, como el término d_k ahora hace parte de la ecuación (21) llegando a:

$$y_{ss} = Cx_{ss} + Du_{ss} + d_k \quad (36)$$

En esta parte, se diseñará un observador de estados [9] para estimar los estados y las perturbaciones. Dicho observador de estados será de la forma:

$$z_{k+1} = \tilde{A}z_k + \tilde{B}u_k - L(\tilde{C}z_k + Du_k - y_k) \quad (37)$$

Donde,

$$z_{k+1} = \begin{bmatrix} x_{k+1} \\ d_{k+1} \end{bmatrix}; \quad \tilde{A} = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}; \quad \tilde{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}; \quad \tilde{C} = [C \quad I] \quad (38)$$

Antes de diseñar el observador verificamos que el sistema sea observable, la matriz de observabilidad es:

$$O = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0.4047 & 0 \\ 1 & 0.01 \\ 0.004 & 1 \\ 0.4047 & 0.004 \end{bmatrix} \quad (39)$$

La matriz O es de rango completo por lo que el sistema es completamente observable. Para calcular la ganancia del observador L que aparece en la ecuación (37), se usa la función *place* de Matlab, la cual calcula dicha ganancia a partir de \tilde{A}, \tilde{C} y los polos del sistema. Los polos escogidos son [0.001,0.01,0.05,0.07]. Estos polos se encuentran cerca del origen para que el observador tenga una respuesta mucho más rápida respecto a la del sistema.

La ganancia L es:

$$L = \begin{bmatrix} 4.14 & 92.78 & -2.21 & -92.76 \\ 232.77 & 2.27 & -232.75 & -0.33 \end{bmatrix} \quad (40)$$

Dado que la referencia y las perturbaciones son conocidas, es necesario calcular x_{ss} y u_{ss} para obtener la salida deseada.

De la ecuación (20) sabemos que $x_{ss} = (I - A)^{-1}Bu_{ss}$ y la ecuación (21) ahora es de la forma

$$y_{ss} = Cx_{ss} + Du_{ss} + d_k \quad (41)$$

Por lo que si $y_{ss} = r$ se tiene:

$$r - d_k = C(I - A)^{-1}Bu_{ss} + Du_{ss} \quad (42)$$

Al despejar u_{ss} de la ecuación (36) encontramos que:

$$u_{ss} = ((\tilde{A} + D)^T(\tilde{A} + D))^{-1}(\tilde{A} + D)^T(r - d_k) \quad (43)$$

Con estos valores, se plantea un nuevo diagrama para el controlador y el observador tal y como se ve en la figura 24:

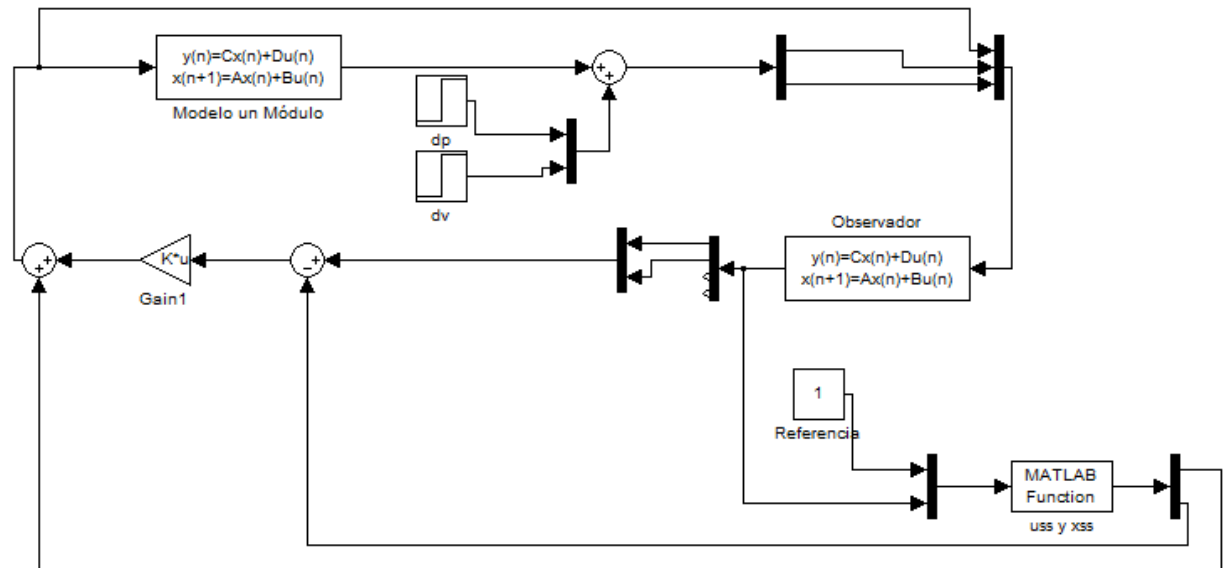


Figura 24. Diagrama del sistema de control en lazo cerrado con el Observador de Estados.

Al diagrama de la figura 24 le es aplicada a los 100 segundos una señal escalón de amplitud 0.5 como perturbación y una referencia $r = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, obteniendo los siguientes resultados:

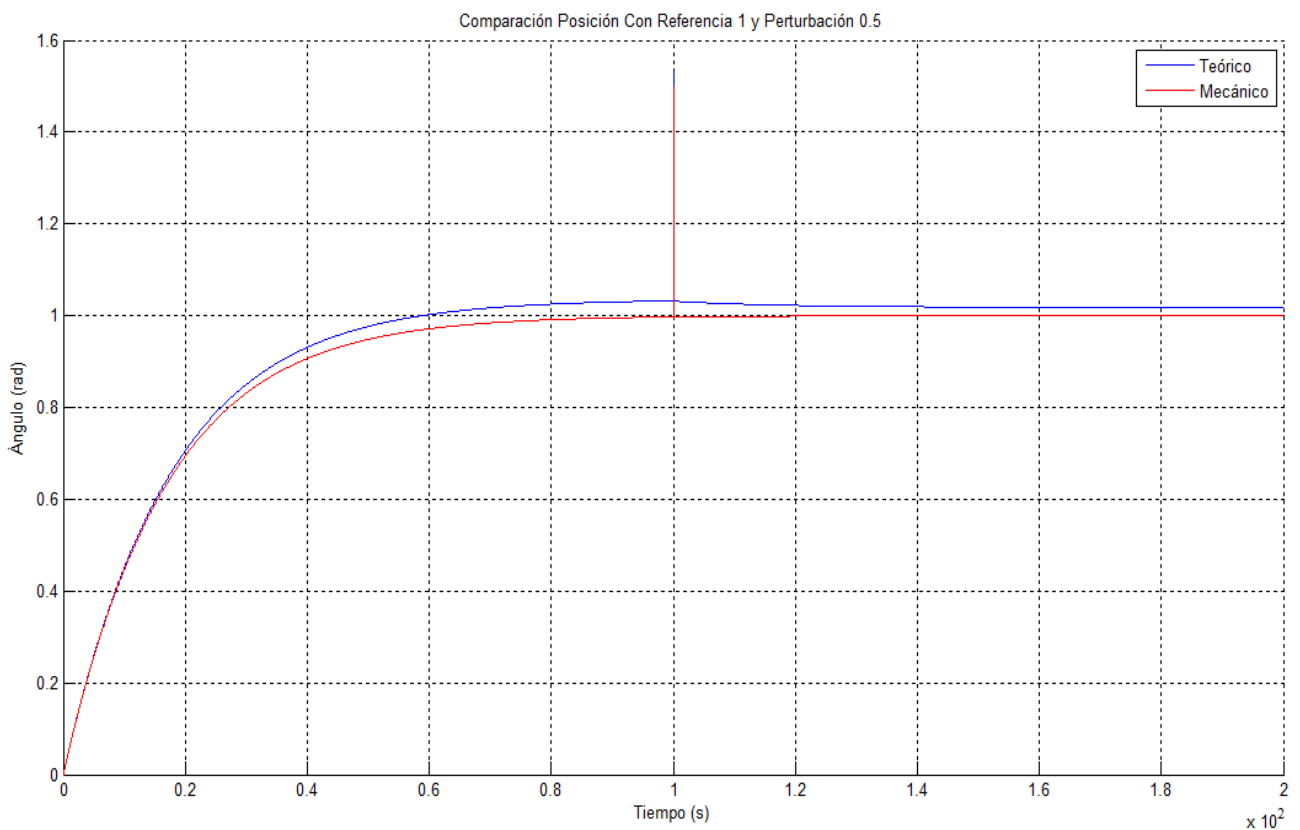


Figura 25. Comparación Posición control GPC con una Referencia igual 1 y una Perturbación igual a 0.5

Para ver como se corrige el error, se grafica la figura 25 pero solo el intervalo de tiempo donde es aplicada la perturbación. Los resultados se pueden observar en la figura 26.

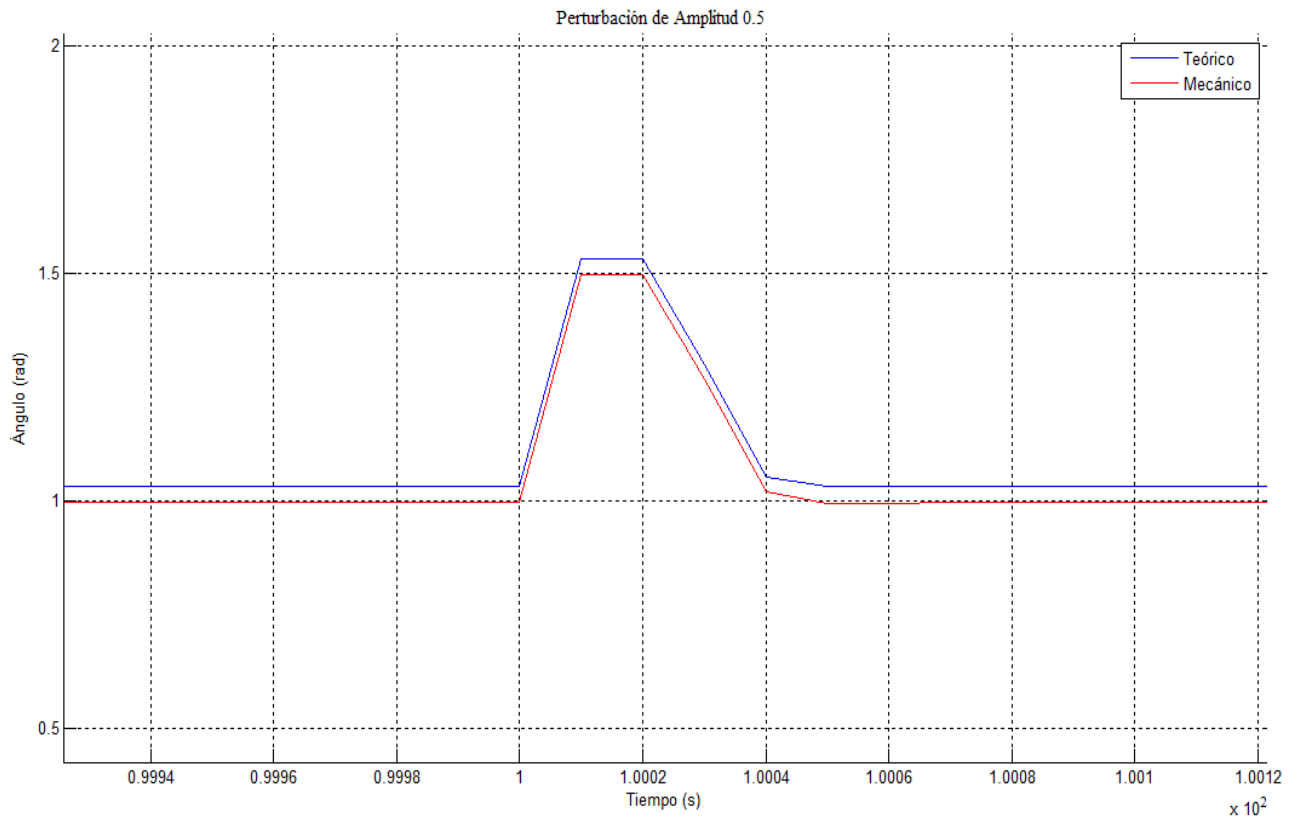


Figura 26. Ampliación Comparación Posición con una Referencia igual 1 y una Perturbación igual a 0.5

En la figura 26, se observa que el error es corregido en 0.04 segundos volviendo rápidamente a su valor original. Además, el control diseñado con el modelo mecánico muestra un mejor desempeño debido a que sigue el valor 1 de la referencia.

A continuación, se muestran las graficas de velocidad cuando es aplicada una perturbación de amplitud 0.5 en la salida de posición.

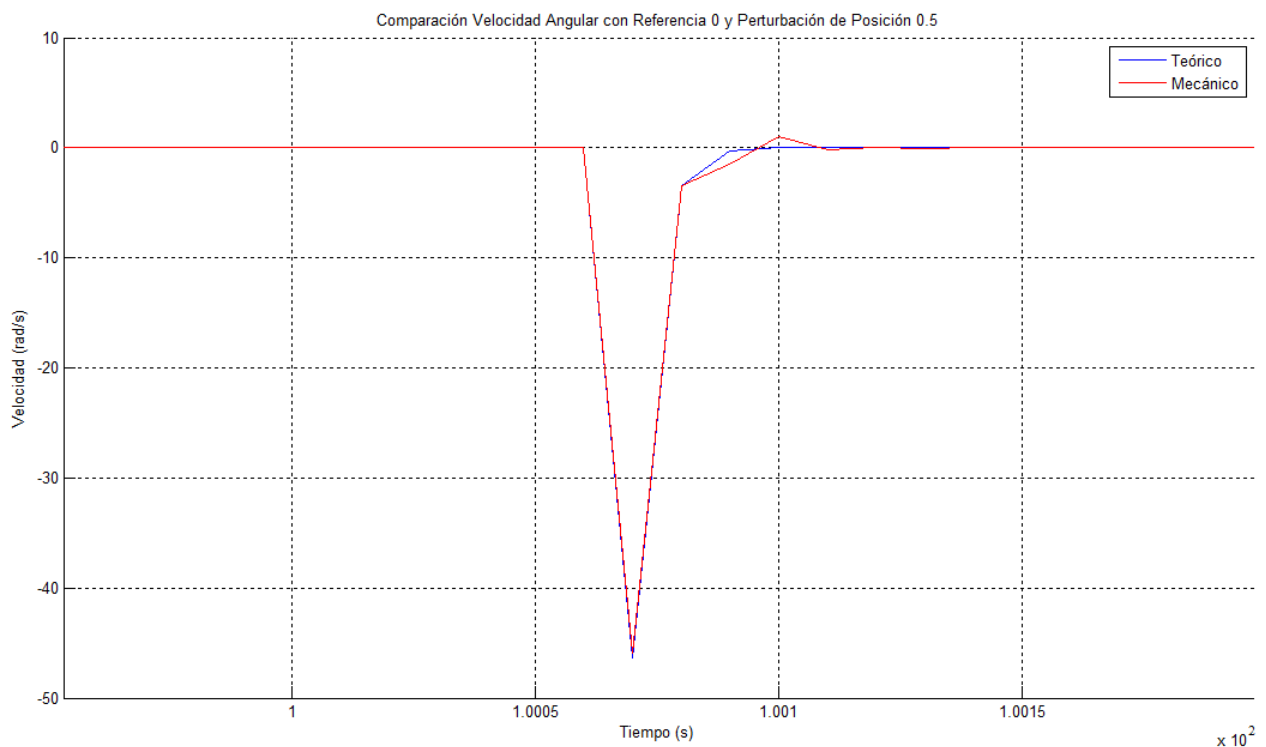


Figura 27. Comparación Velocidad Angular control GPC con una Referencia igual 0 y una perturbación de Posición igual a 0.5

En las graficas anteriores, se puede observar el desempeño del controlador tanto para el modelo teórico como para el modelo mecánico. Este desempeño no es el mismo para los dos modelos debido a que las matrices de las ecuaciones de estado que describen al sistema no son totalmente iguales, tal y como se ve a continuación. Se resalta, que estas matrices fueron obtenidas en el capítulo 3 pero no son mostradas en ningún momento. Se recomienda al lector revisar el script Datos GPC creado en Matlab el cual se encuentra en el CD adjunto a este libro, en dicho script se describe como hallar las matrices que aparecen a continuación;

Modelo Teórico:

$$A = \begin{bmatrix} 1 & 0.01 \\ 0.004 & 1 \end{bmatrix}; B = \begin{bmatrix} 0.1111 \\ 22.22 \end{bmatrix}; C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0.4047 & 0 \end{bmatrix}; D = \begin{bmatrix} 0 \\ 0 \\ 2222.2 \end{bmatrix}$$

Modelo Mecánico:

$$A = \begin{bmatrix} 1 & 0.01 \\ 0 & 1 \end{bmatrix}; B = \begin{bmatrix} 0.1101 \\ 22.02 \end{bmatrix}; C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}; D = \begin{bmatrix} 0 \\ 0 \\ 2202.6 \end{bmatrix}$$

Usando estas matrices, se calcula la ley de control cuando $d_k = 0$, encontrando algunas diferencias entre los dos modelos.

Modelo Teórico:

$$K = [-0.0027 \quad -0.0449]$$

Modelo Mecánico:

$$K = [-0.0026 \quad -0.0453]$$

Para el diseño del controlador cuando es aplicada una perturbación al sistema, la ganancia K se escoge igual tanto para el modelo mecánico como para el modelo teórico. Esto debido a como se explicó en el capítulo 3, el modelo se comporta de una manera muy similar por lo que el desempeño del controlador no debería verse afectado ya que estaría simulando una ‘misma’ planta. En el capítulo 6, estos controladores serán diseñados para diferentes horizontes de predicción y serán probados con diferentes perturbaciones.

4.3 UMPC (Unconstrained Model Predictive Control)

El segundo controlador se desarrolló utilizando el toolbox *Model Predictive Control* de MATLAB. La estrategia de control que utiliza es UMPC¹⁴, dicha estrategia se ajusta al problema de diseño que se está tratando debido a que el controlador que se va a implementar no tiene ninguna restricción.

Se decide trabajar con el control UMPC debido a que a partir de este se pueden llegar a entender las demás estrategias de control predictivo. El UMPC cuenta con los principales elementos de un control predictivo, el cual usa el modelo matemático del proceso (planta) para predecir sus salidas futuras, basándose en las señales de control futuras. Estas señales se calculan con ayuda de un programa de optimización que siempre requiere del planteamiento de una función objetivo y de ser el caso, de restricciones activas o pasivas. Este modelo matemático debe ser capaz de describir en forma muy cercana la dinámica del proceso para poder predecir las salidas futuras. Otros componentes estructurales de un control MPC son la trayectoria de referencia, los errores futuros (diferencia entre la Trayectoria de referencia y las salidas predichas) y las señales de control futuras [10].

De esta manera, para empezar con el diseño del controlador deben seguirse los siguientes pasos:

¹⁴Unconstrained Model Predictive Control

4.3.1 Conexión del controlador a la planta e identificación del tipo de señales con las que está interactuando.

Lo primero que se hace es conectar el controlador (Bloque MPC Controller) y la referencia a la planta para definir el sistema con el que se va a trabajar (Figura 28).

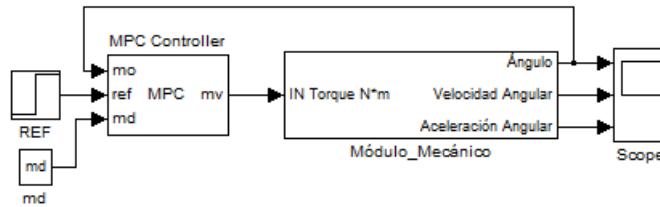


Figura 28. Diagrama Control UMPC usando el Modelo Mecánico

Después de tener el diagrama del sistema realimentado, se debe especificar qué tipo de señales interactúan con la planta, saber cuáles se van a medir y cuáles se van a controlar; cada señal tiene que clasificarse como se muestra en la Figura 29.

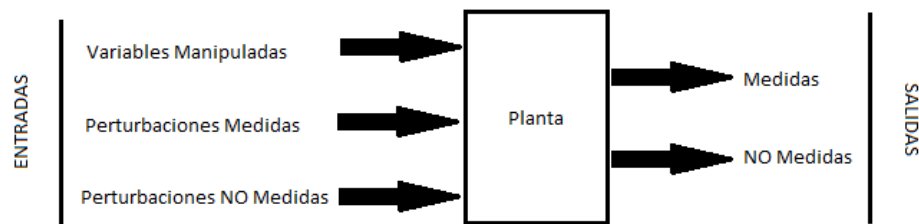


Figura 29. Control UMPC. Clasificación Variables de Entrada y Salida de la Planta

Las entradas que se ven en la figura 29 son variables que afectan a la planta, hay tres tipos de entradas tal y como se ve en la figura anterior.

Para la planta con la que se está trabajando las entradas que se identifican son las siguientes:

- **IN Torque**, además de ser una entrada también es una variable manipulada ya que corresponde a la salida del controlador.
- **MD** (Perturbaciones Medidas), el controlador utiliza las perturbaciones medidas para una compensación “feedforward”, en la Figura 28 el bloque que corresponde a este tipo de perturbaciones está conectado al controlador, esto se hace para efectos de diseño, en realidad estas perturbaciones no afectan al controlador sino a la planta.
- **Perturbaciones NO Medidas**, no aparecen en el diagrama de bloques pero hay que tenerlas en cuenta debido a que el control no siempre va a trabajar bajo las mismas condiciones, este tipo de perturbaciones son entradas independientes de las cuales el controlador no tiene conocimiento directo pero debe compensar.

Ahora bien, en la figura 29 también se pueden observar las salidas las cuales son variables dependientes que salen de la planta. Para este caso en particular, la única salida que se va a tener en cuenta es el ángulo de cada uno de los módulos. El sistema es realimentado con dichos ángulos por lo tanto se deben considerar como variables medidas. El controlador usa esta clase de variables para estimar las cantidades no medidas las cuales también son realimentadas, llegando así al control deseado.

Luego de haber definido el grupo al que corresponden cada una de las variables del sistema, se procede con el reconocimiento y la identificación de la planta por parte del controlador.

4.3.2 Reconocimiento e Identificación de la planta por parte del Controlador

En el toolbox MPC de Matlab, el controlador debe saber qué tiene conectado por lo que sigue los pasos que son ilustrados en la Figura 30.

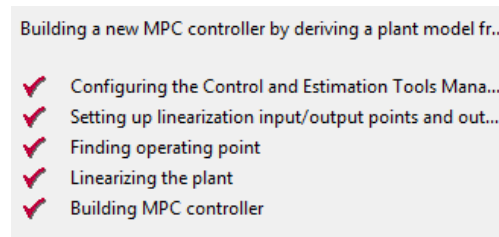


Figura 30. Secuencia de pasos de inicialización del MPC Controller¹⁵

- El primer paso consiste en la configuración de las herramientas de control y estimación, en esta etapa se crea el proyecto y se le asigna el mismo nombre que el modelo en Simulink.
- El segundo paso consiste en identificar las entradas y las salidas, éste es el momento en el cual el controlador identifica qué tipo de señales tiene conectadas.
- El objetivo del tercer paso es encontrar el punto de operación.
- En el cuarto paso se calcula el modelo linealizado de la planta, hay que aclarar que el control predictivo es capaz de controlar plantas no lineales, como la que se está estudiando, pero para efectos de diseño de este controlador se necesita tener un modelo linealizado. Para la linealización de la planta, el toolbox *MPC* hace uso del otro toolbox de Matlab *System Identification Toolbox*, éste se encarga de generar un modelo lineal e invariante en el tiempo basándose en las entradas y salidas de la planta. La forma en que el toolbox linealiza el modelo consiste en crear dos vectores, uno con una secuencia de entradas a la planta (entradas de torque, unidades Nm), y el otro con las respectivas salidas (salidas de ángulo, radianes). Después de tener los dos vectores y el tiempo de muestreo (definido en el modelo mecánico $t_s=0.01s$) el toolbox procede a crear un modelo que ajuste de la mejor forma los datos, el cual va a tener contenida la información sobre la estructura del sistema y la información sobre el algoritmo usado para estimar los parámetros. Luego de crear el modelo linealizado (System Identification Toolbox Model), este debe convertirse a un "*LTI object*" para poder ser utilizado con el Toolbox Model Predictive Control de Matlab.
- Por último, el quinto paso, consiste en consolidar toda la información para poder generar el controlador, después de esto se despliega un conjunto de opciones con las que se pueden ajustar y modificar los parámetros de dicho controlador.

4.3.3 Comparación de resultados por parte del Toolbox.

Se deben comparar dos cosas, la primera, que las variables que definimos en el paso 4.3.1 concuerden con las que el toolbox identificó, y la segunda, que las matrices que muestra el toolbox del modelo linealizado sean muy similares a las que se calcularon en el capítulo 3.

- Al inicializar el MPC toolbox aparece la ventana de la Figura 31, como se puede observar dicha figura concuerda perfectamente con lo que se había planteado en el paso 4.3.1, hay que aclarar que el toolbox muestra 3 Perturbaciones Medidas y antes sólo se había hablado de

¹⁵ Figura Tomada del Toolbox Model Predictive Control de Matlab

una (Figura 29), esto se debe a que el bloque MD de la Figura 29 lo conforman tres señales, las cuales simulan las componentes x,y,z del torque de interacción entre módulos. Estos torques son tenidos en cuenta en el capítulo 5 donde se distribuye el controlador para 6 módulos.

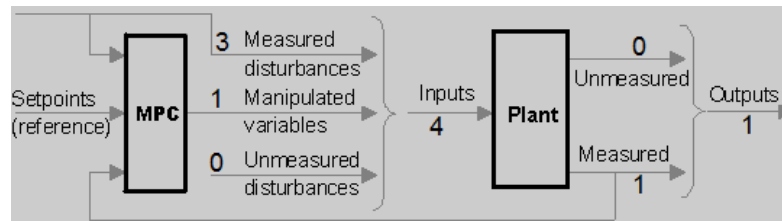


Figura 31. Representación del sistema que es identificado por el toolbox

- La linealización que se obtiene por medio del toolbox (Figura 32) es muy parecida a la del modelo teórico que se planteó en el capítulo anterior. La matriz A es igual a la del modelo mecánico, B tiene los mismos valores en la primera columna pero adicionalmente presenta tres columnas más, esto se debe a que el *System Identification Toolbox* crea automáticamente entradas de ruido correspondientes a las perturbaciones medidas. Al igual que B, el tamaño de C es diferente, debido a que para este caso en particular solo se realimenta la posición.

```

Linearization Result:

- To plot the response of this result click on the node labeled Custom Views.
- To export the result click on the Export To Workspace button below.

a =
      x1    x2
x1      1  0.01
x2      0      1

b =
      u1    u2    u3    u4
x1  0.09452  0      0      0
x2   18.9      0      0      0

c =
      x1    x2
y1      1      0

d =
      u1    u2    u3    u4
y1      0      0      0      0

Input groups:
  Name      Channels
Manipulated 1
Measured   2,3,4

```

Figura 32. Resultados de la linealización del toolbox

4.3.4 Ajuste de los parámetros del controlador

El toolbox ofrece distintos parámetros de diseño del controlador, los principales y con los que se van trabajar son:

Horizontes: hay que definir 3 sub parámetros dentro de esta categoría:

- **Intervalo de control:** establece el tiempo transcurrido entre cada ajuste de la variable manipulada de control.
- **Horizonte de predicción:** es el número de intervalos de control sobre los cuales la salida es optimizada.
- **Horizonte de control:** es el número de intervalos de control sobre las cuales la variable de control es optimizada.

Pesos: hay que definir 2 sub parámetros

- **Pesos de entrada:** esta variable define el peso para la señal de entrada, si este valor es cero no afecta en nada el diseño del controlador, pero cualquier perturbación o cambio de referencia puede desviar permanentemente la variable manipulada del valor nominal, en cambio, si es diferente de cero, hace que la variable manipulada esté más cerca a su valor nominal, eliminando así el error en estado estacionario de la salida de la planta. De forma paralela hay que definir otra variable llamada “Rate Weigth”, por defecto esta variable se define cero y no tiene ningún efecto para el diseño del controlador, pero si se le asigna un valor positivo, éste se encarga de incrementar el peso en el momento necesario para reducir el movimiento en la variable manipulada.
- **Pesos de salida:** esta variable define el peso para la señal de salida, puede tomar valores positivos o cero. Cuando hay una sola señal de salida, incrementar el peso sirve para aumentar la cercanía de la salida de la planta con la referencia; cuando hay más de una variable de salida, la que tenga el mayor peso va ser a la que el controlador le va dar más importancia y va tratar de corregir más rápidamente.

Adicionalmente, el toolbox en esta misma sección presenta una herramienta gráfica que permite escoger que característica principal se desea que tenga el controlador, ésta consiste un selector que al variarlo permite volver más rápido el controlador dejando a un lado la robustez o viceversa. El funcionamiento de esta opción radica en variar todos los pesos a la vez, por ejemplo, si la barra se mueve hacia el lado de mayor robustez aumenta el peso que corresponde a la variable manipulada y disminuye los relacionados con la referencia.

Estimación: en esta sección se definen el tipo de perturbaciones para las que el controlador se va a diseñar, hay 3 tipos:

- **Perturbaciones a la salida de la planta**



Figura 33. Perturbaciones a la salida de la planta

- **Perturbaciones a la entrada de la planta**

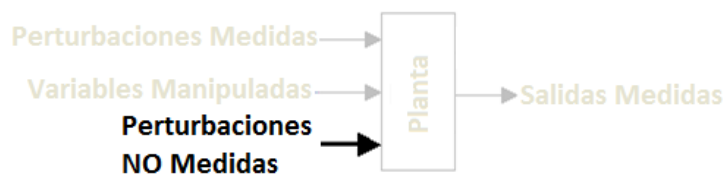


Figura 34. Perturbaciones a la entrada de la planta

- **Ruido Medido**

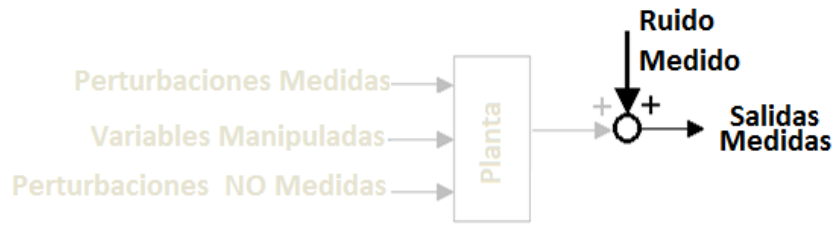


Figura 35. Ruido Medido a la salida de la planta

Para cada uno de estos se puede escoger el tipo de perturbación, pueden ser señales cuadradas, rampas o ruido blanco, adicional a esto se puede escoger la magnitud de cada señal.

4.3.5 Diseño del controlador

Después de ver y analizar cada uno de los elementos que hacen parte del toolbox, se procede a implementar el control con parámetros similares a los que se utilizaron en el control GPC.

Horizonte de predicción	4
Horizonte de control	2
Pesos de entrada	1
Pesos de salida	1

Tabla 9. Parámetros Escogidos Para el diseño del controlador UMPC

Con estos parámetros se obtienen los siguientes resultados.

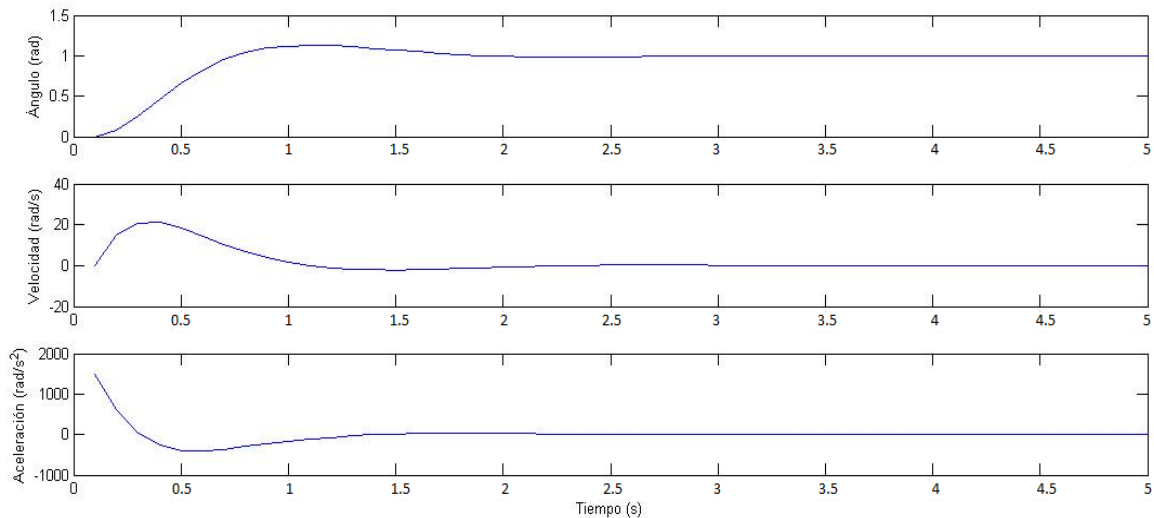


Figura 36. Control UMPC. Salidas de posición, velocidad y aceleración para una entrada paso

Como se observa en la figura 36, el control UMPC alcanza el valor de la referencia en 1.8 segundos, por lo que responde casi 80 veces más rápido que el control GPC diseñado en las secciones 4.1 y 4.2 de este capítulo.

En la figura 37 se grafica la referencia y la salida del controlador, observando el rendimiento del mismo

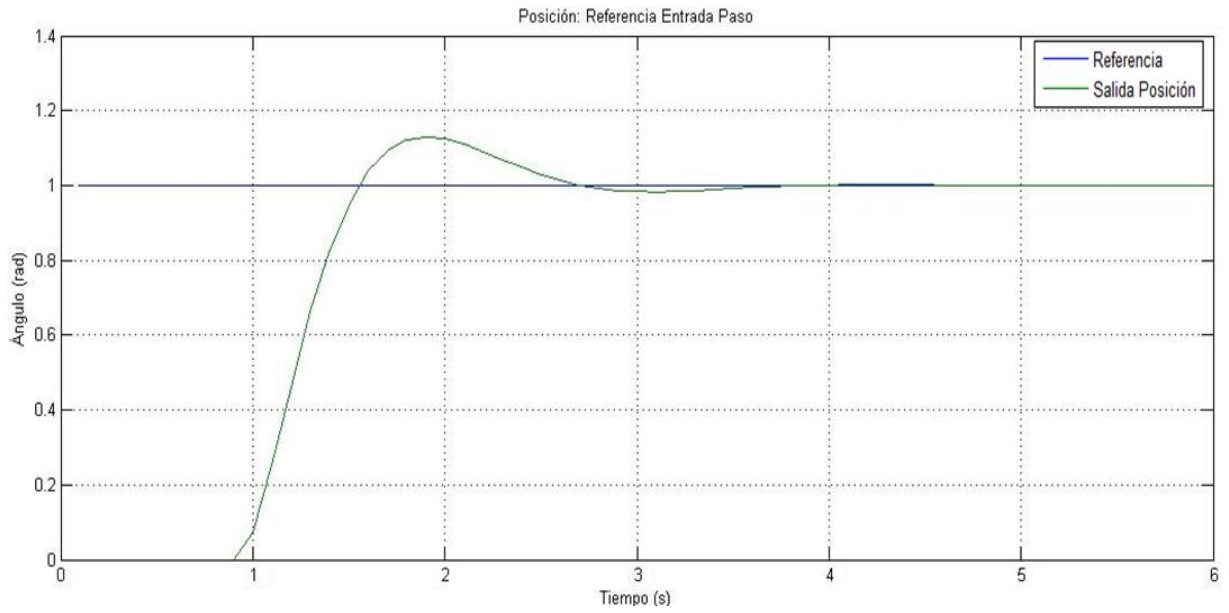


Figura 37. Control UMPC. Salida de posición en respuesta a una entrada paso

La señal de control para el controlador UMPC se puede ver en la figura 38.

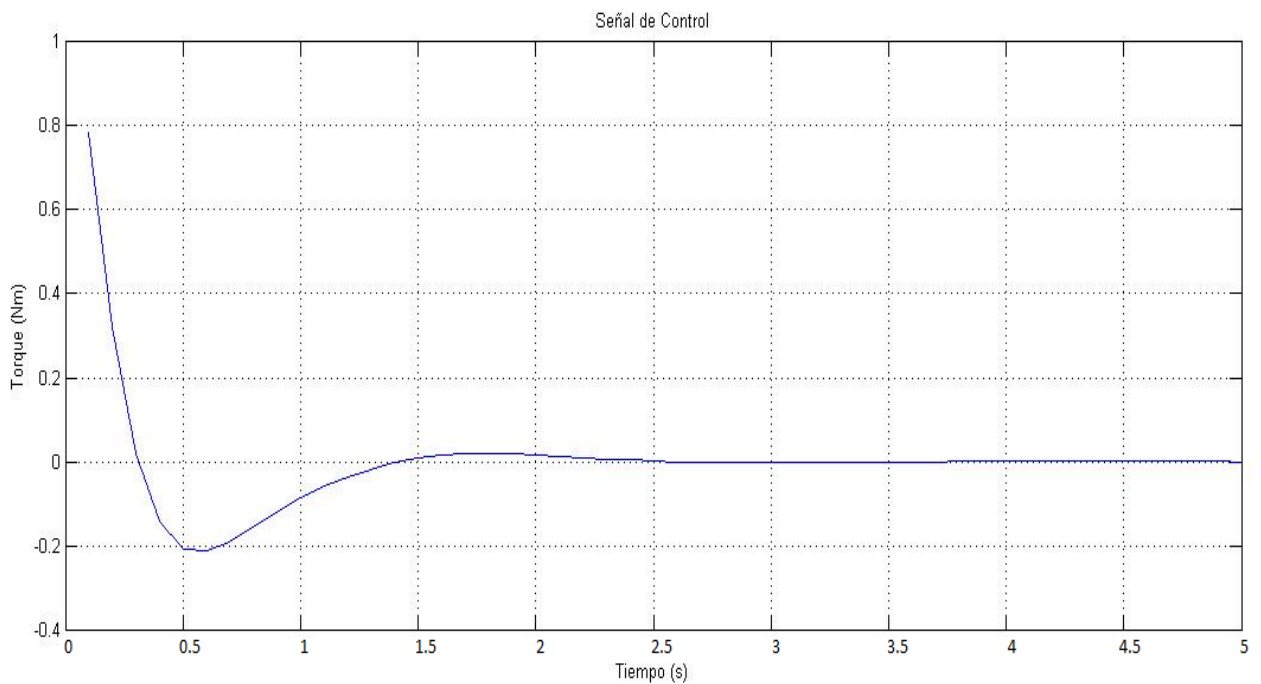


Figura 38. Señal de Control GPC para el Modelo Mecánico y para el Modelo Teórico

Por último, debe resaltarse que el toolbox MPC de Matlab diseña el controlador teniendo en cuenta que puede verse sometido a una perturbación, por lo que en esta sección no es necesario diseñar un observador de estados tal y como se hizo en la sección 4.2.

En la figura 39 se muestra la respuesta a perturbaciones no medidas:

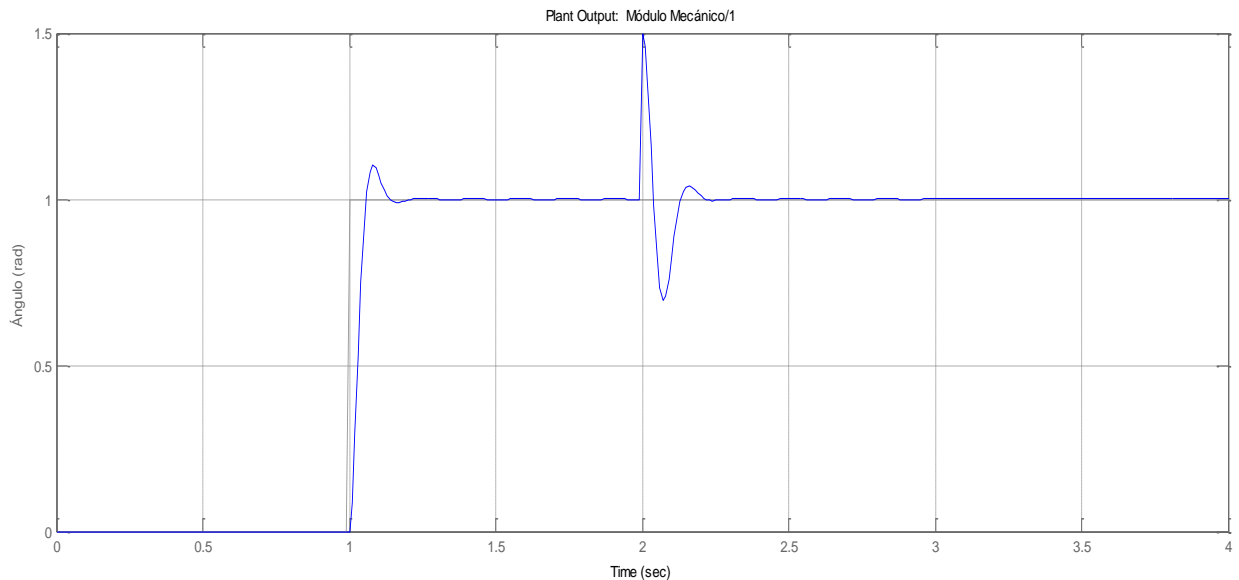


Figura 39. Control UMPC. Respuesta a perturbaciones NO medidas

Se debe aclarar al lector que en este punto no se ha escogido el mejor controlador, como se menciono anteriormente, en el capítulo 6 se harán varias pruebas cambiando los parámetros de diseño para poder escoger el que presente el mejor desempeño. Sin embargo, en este capítulo queda claro que el control UMPC responde casi 80 veces más rápido respecto al control GPC (Ver figuras 21 y 37).

Por último, es importante resaltar que la fácil implementación es otra de las grandes ventajas que ofrece el Model Predictive Toolbox y más exactamente, el bloque MPC Controller. Después de tener diseñado y listo el controlador en Simulink, basta con oprimir una combinación de teclas para invocar el programa Real-Time Workshop, el cual se encarga de generar un modelo del controlador, que puede ser descargado fácilmente en cualquiera de las siguientes tarjetas para control en tiempo real:

- Generic Real-Time Target
- Real-Time Workshop Embedded Coder
- Real-Time Windows Target
- Rapid Simulation Target
- Target Support Package for Use with Freescale® MPC5xx
- xPC Target
- dSpace Target
- Target for Infineon TriCore

5. CONTROL DISTRIBUIDO

En el capítulo anterior, se calculó un controlador predictivo para el módulo base de la cadena. Sin embargo, la cadena que se quiere controlar en este proyecto consta de 6 módulos por lo que es necesario diseñar un controlador distribuido [11] que tenga un buen desempeño en cuanto a tiempo de establecimiento y sobrepicos.

Para diseñar este controlador, es necesario tener en cuenta las fuerzas de interacción entre módulos [12], dichas fuerzas son modeladas por medio del uso de *SimMechanics* debido a que una vez se interconectan más de un módulo, la fuerza de reacción y el torque de reacción se encargan de modificar el comportamiento de cada uno de los módulos en donde son aplicadas de uno u otro modo.

De esta manera, en esta parte del trabajo se ilustrara el control distribuido para dos módulos haciendo uso del control GPC y el control MPC del Toolbox de Matlab diseñados en el capítulo inmediatamente anterior.

Por último, se extrapolara el control distribuido a los 6 módulos que componen la cadena robótica. Cabe resaltar que cada módulo trabaja de forma independiente y que la interacción entre módulos se tomara como una entrada adicional a cada uno de estos.

Es importante aclarar que como los módulos no giran en el mismo eje, los torques de reacción no afectan al módulo inmediatamente siguiente al que están conectados, sino que afectan al siguiente módulo que se encuentre girando en el mismo eje, por lo tanto, el torque de reacción del primer módulo afectará al tercero, el del tercero afectará tanto al primero como al quinto, ya que habrá un torque medido respecto al “base” y otro respecto al “follower”. En el caso de los que giran en el eje z la situación es parecida, el torque medido respecto al “follower” en el segundo módulo afectará al cuarto, y el cuarto afectará tanto al segundo como al sexto, uno con el torque medido respecto a la “base” y otro respecto al “follower”. Aunque a simple vista no es tan claro, la Figura 46 aclarará cualquier duda.

5.1 CONTROL GPC DISTRIBUIDO 2 MÓDULOS

Haciendo uso del modelo mecánico creado por medio de *SimMechanics* y con el control predictivo generalizado (GPC) se plantea el siguiente diagrama.

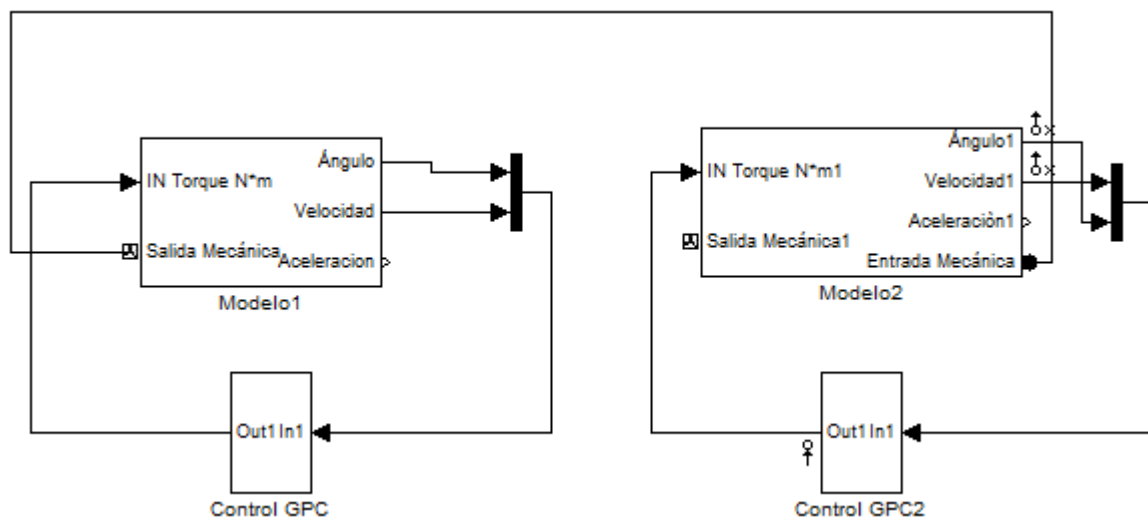


Figura 40. Control GPC Distribuido Para Dos Módulos

En el diagrama de la figura 40, uno de los módulos se encuentra rotando sobre el eje y y el otro sobre el eje z. Se escogió que rotaran de esa manera, debido a que esa clase de conexión permite una mejor locomoción de la cadena. Es importante aclararle al lector que en este trabajo de grado la cadena no se desplazara sino que seguirá una referencia en particular dada por el usuario.

Si la referencia es $r = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ para los dos módulos, el controlador muestra el siguiente desempeño:

En la figura 41 y la figura 42 se pueden observar los resultados de simulación para cada uno de los módulos, de esta manera:

Para el módulo uno (Figura 41):

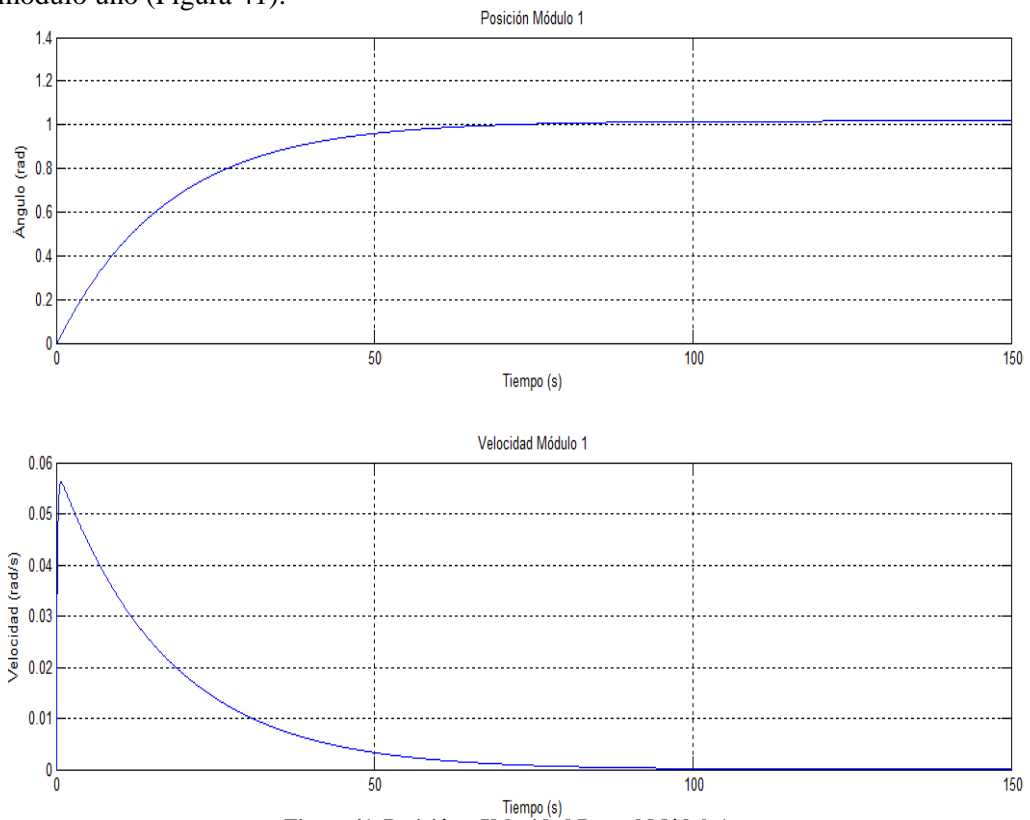


Figura 41. Posición y Velocidad Para el Módulo 1

Y para el módulo 2 (Figura 42),

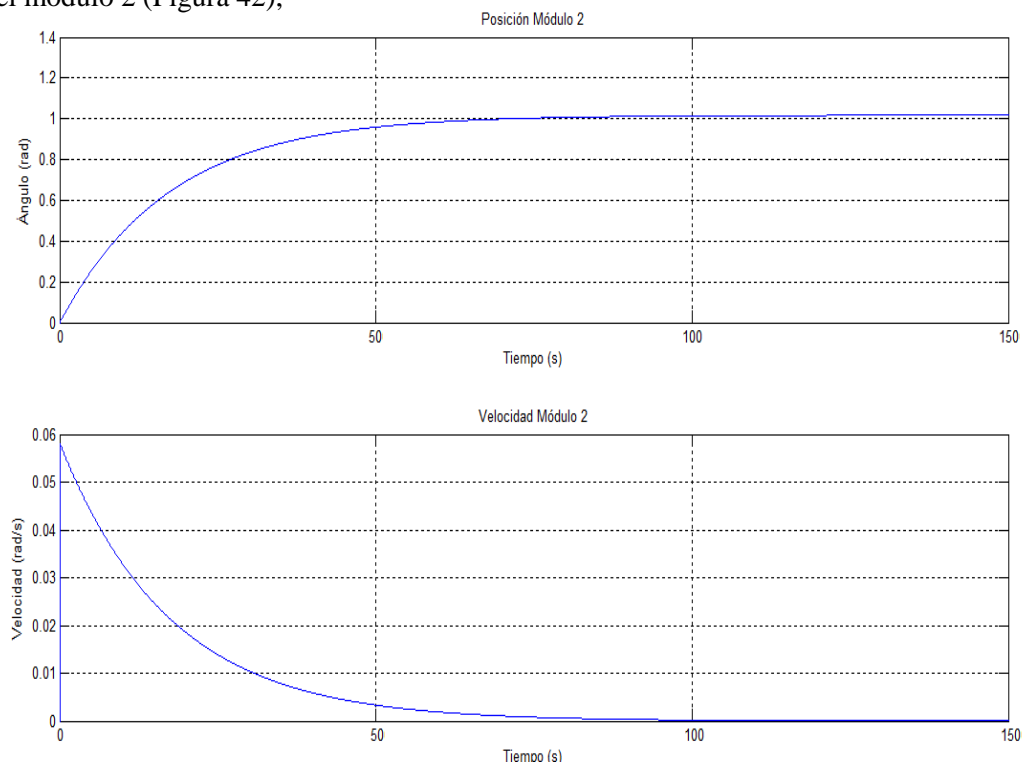


Figura 42. Posición y Velocidad Para el Módulo 2

En las figuras 41 y 42 se muestra graficas de posición y velocidad para cada uno de los módulos una vez el control es distribuido. Como se puede ver, los dos módulos alcanzan el valor de la referencia en el mismo tiempo lo que nos muestra que el desempeño del controlador es el mismo que se tenía para un solo módulo. Estas graficas nos permiten ver de una manera clara el fin de un control distribuido, ya que según se observa, cada uno de los controles actúa de forma independiente pero logrando el objetivo común el cual es la posición 1 para el módulo 1 y la posición 1 para el módulo 2.

En la siguiente sección se muestran los resultados de distribuir el control para dos módulos usando el toolbox MPC de Matlab.

5.2 CONTROL UMPC DISTRIBUIDO 2 MÓDULOS

Al igual que para el control GPC, el eje de rotación del primer módulo es y y el del segundo módulo es el eje z, esta configuración permite que el área de desplazamiento de la cadena robótica sea más grande. El diagrama de bloques que se implementó se puede ver a continuación:

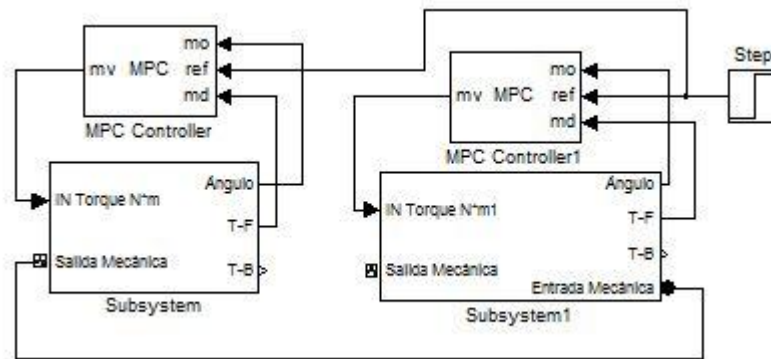


Figura 43. Diagrama de bloques para dos módulos con control UMPC

La referencia tanto para el primer módulo como para el segundo es una entrada paso de amplitud 1 rad, los resultados para el primer módulo se pueden ver en la Figura 44 y para el segundo en la Figura 45.

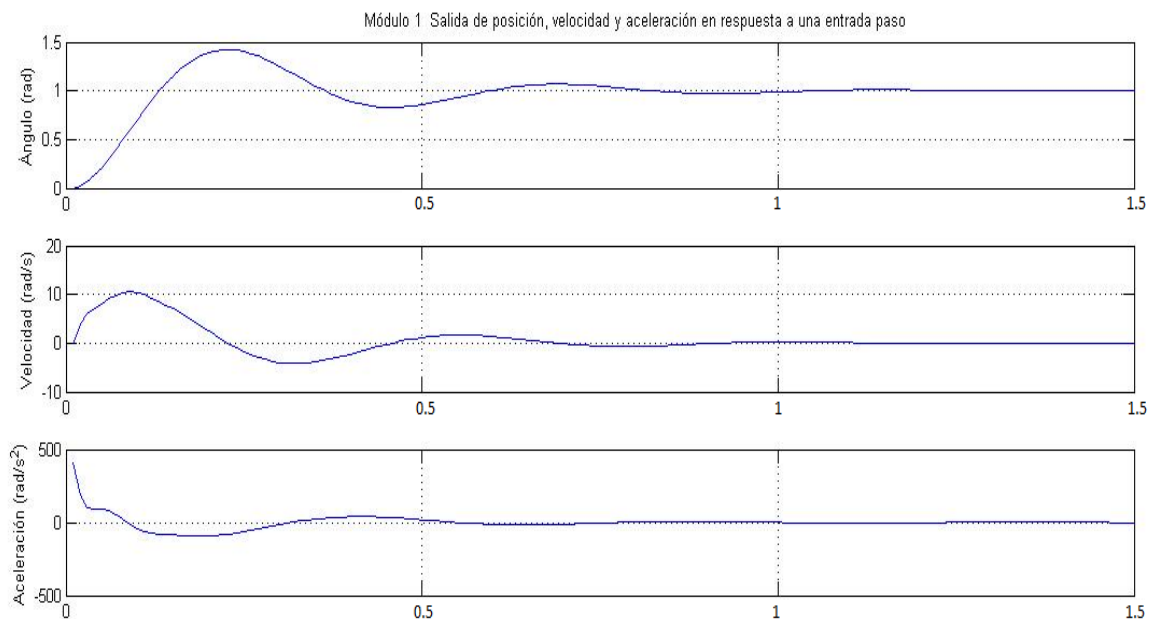


Figura 44. Gráficas de Posición, Velocidad y Aceleración del Primer Módulo

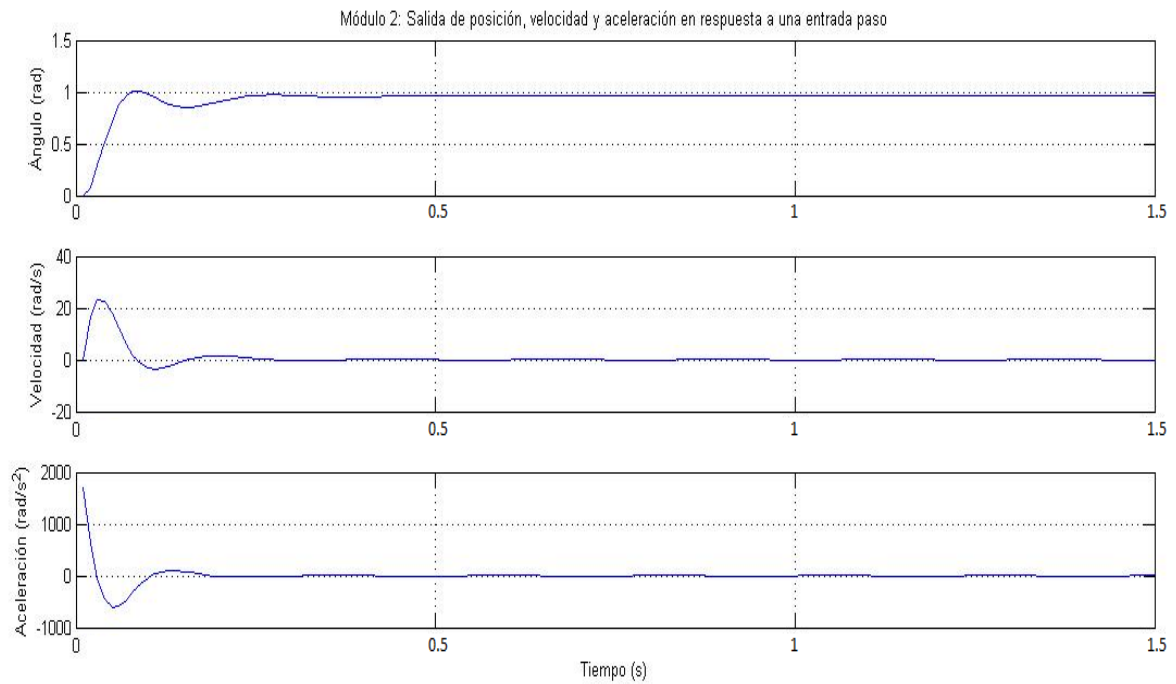


Figura 45. Gráficas de Posición, Velocidad y Aceleración del Segundo Módulo

En las graficas 44 y 45 se observa que el sobrepico para el módulo 2 tiene una menor magnitud y además alcanza la referencia en un menor tiempo. En este caso en particular, se ve como el mismo controlador al ser distribuido tiene una respuesta diferente para cada módulo logrando así el objetivo final que para este caso era la posición 1 para el módulo 1 y la posición 1 para el módulo 2.

5.3 CONTROL UMPC DISTRIBUIDO 6 MÓDULOS

Después de distribuir tanto el controlador GPC como el UMPC para los dos módulos se procede a hacer lo mismo para los 6 módulos, los módulos impares girarán en el eje z y los pares en el eje y . Es importante resaltar que en este libro sólo se hará la simulación de 6 módulos con el modelo mecánico y con el control UMPC, el cual tiene un mejor desempeño, como se vio en el capítulo 4, en cuanto a tiempos de establecimiento, respecto al control GPC, por lo que este último no se distribuirá a seis módulos. Además, se escoge el modelo mecánico debido a que en el capítulo 3 éste se validó, llegando a la conclusión que las simulaciones resultantes, usando cualquiera de los dos modelos, arrojan los mismos resultados.

Ahora bien, se sabe que el bloque “*joint sensor*” además de mostrar la posición, velocidad y aceleración de cada módulo también muestra los torques de reacción entre módulos, estos son medidos respecto al módulo “*base*”, el cual corresponde al módulo que está conectado inmediatamente anterior y respecto al “*follower*”, que corresponde al módulo que se encuentra conectado a continuación.

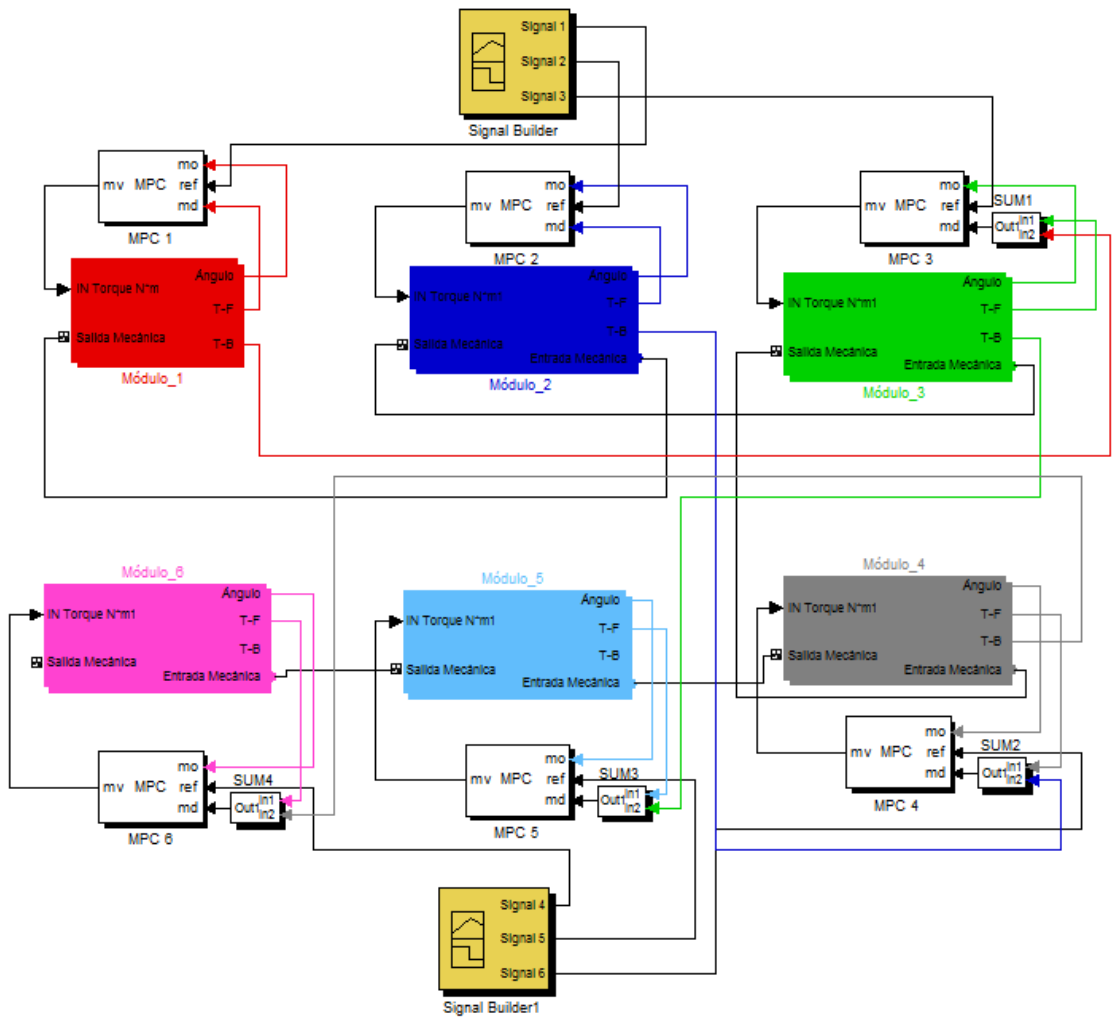
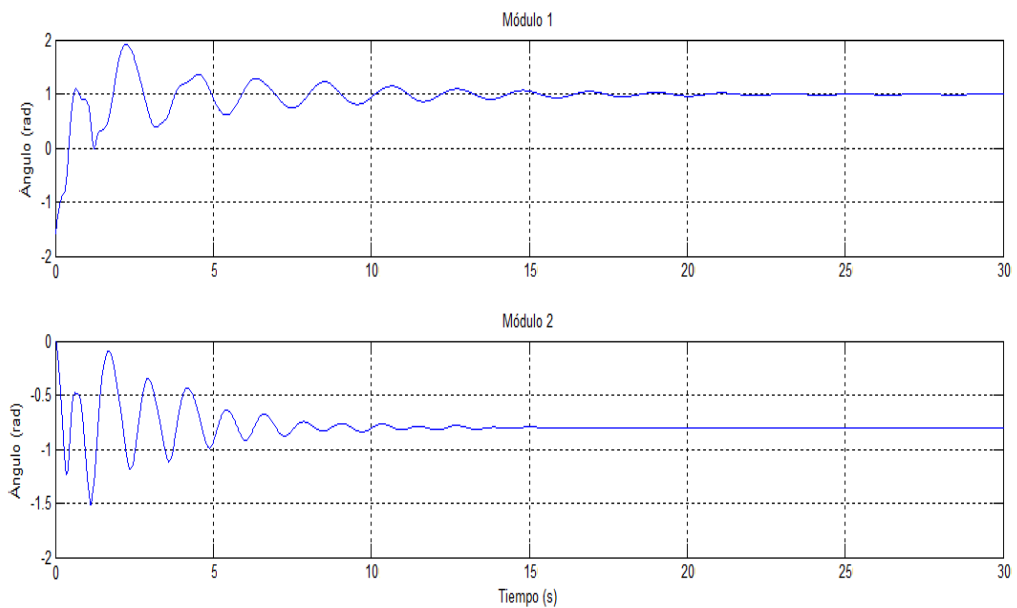


Figura 46. Diagrama de bloques para 6 Módulos

En la figura 47 se observan los resultados de esta configuración aplicando una entrada paso de amplitud 1 a cada uno de los módulos, en este caso esta entrada es la misma para todos los módulos pero puede variar de acuerdo a la trayectoria que se quiere que siga la cadena.



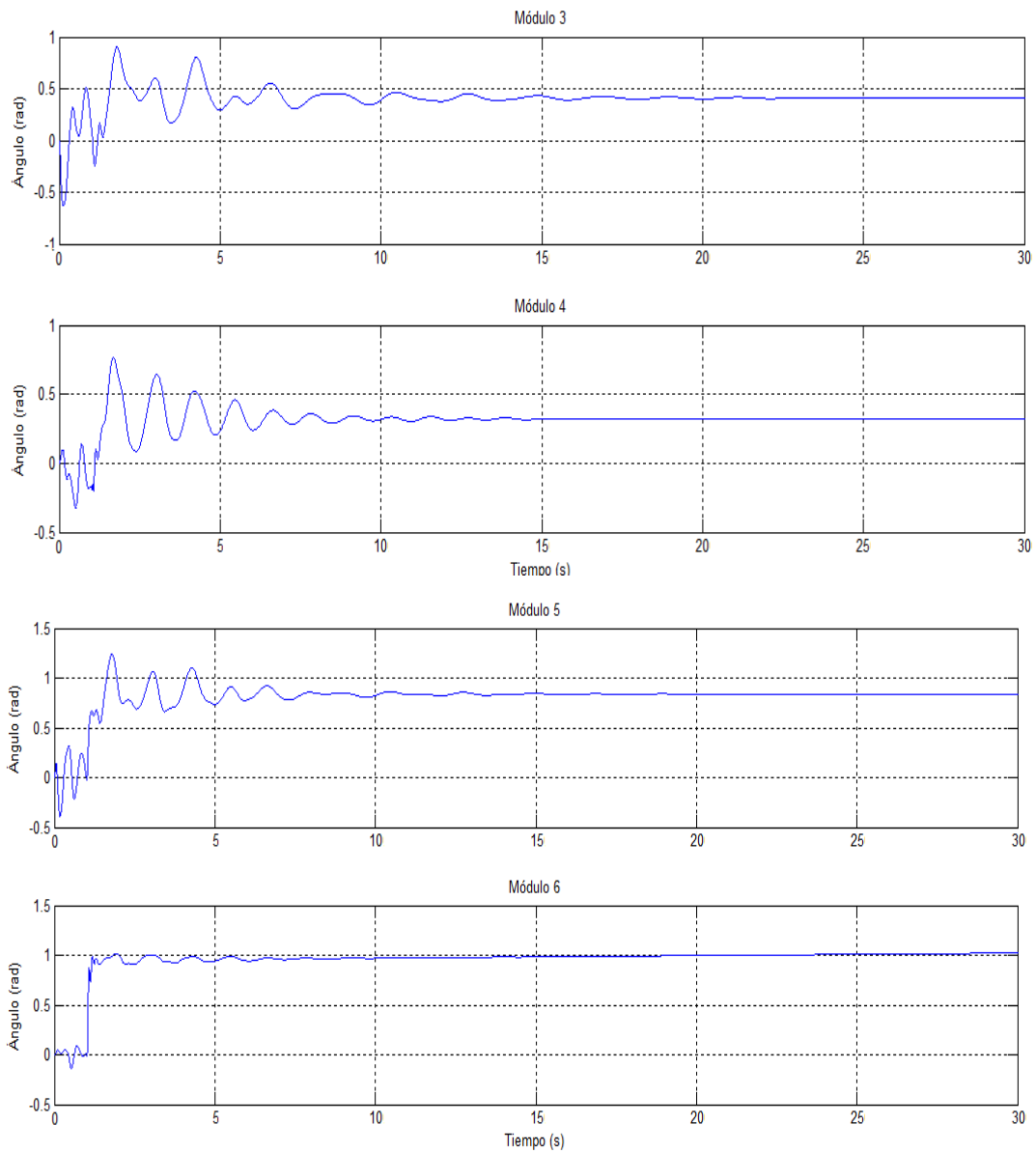


Figura 47. Gráficas de posición para cada módulo

Como se puede ver en la figura 47, la respuesta no es la misma para todos los módulos. El primer módulo llega a la posición deseada al igual que el último, aunque el primero en un tiempo mucho mayor respecto al módulo 6; los módulos 2, 3 y 4 no se acercan a la referencia y el módulo 5, aunque se acerca a la referencia, tiene un error en estado estacionario relativamente grande.

Aunque durante el diseño del controlador se tienen en cuenta los torques de interacción, esto no es suficiente y los resultados en este momento no son los esperados. Este conjunto de errores se debe a que cada controlador está en condiciones diferentes, para corregirlo se va a aumentar la velocidad de respuesta de cada controlador dependiendo la posición en la que se encuentre. Lo anterior se logra agregando una ganancia después de cada controlador de tal forma que la ganancia que tenga el primero sea mayor que la del segundo y así sucesivamente, siendo la más pequeña la del sexto controlador, si la llegará a necesitar. El nuevo diagrama de bloques es el siguiente:

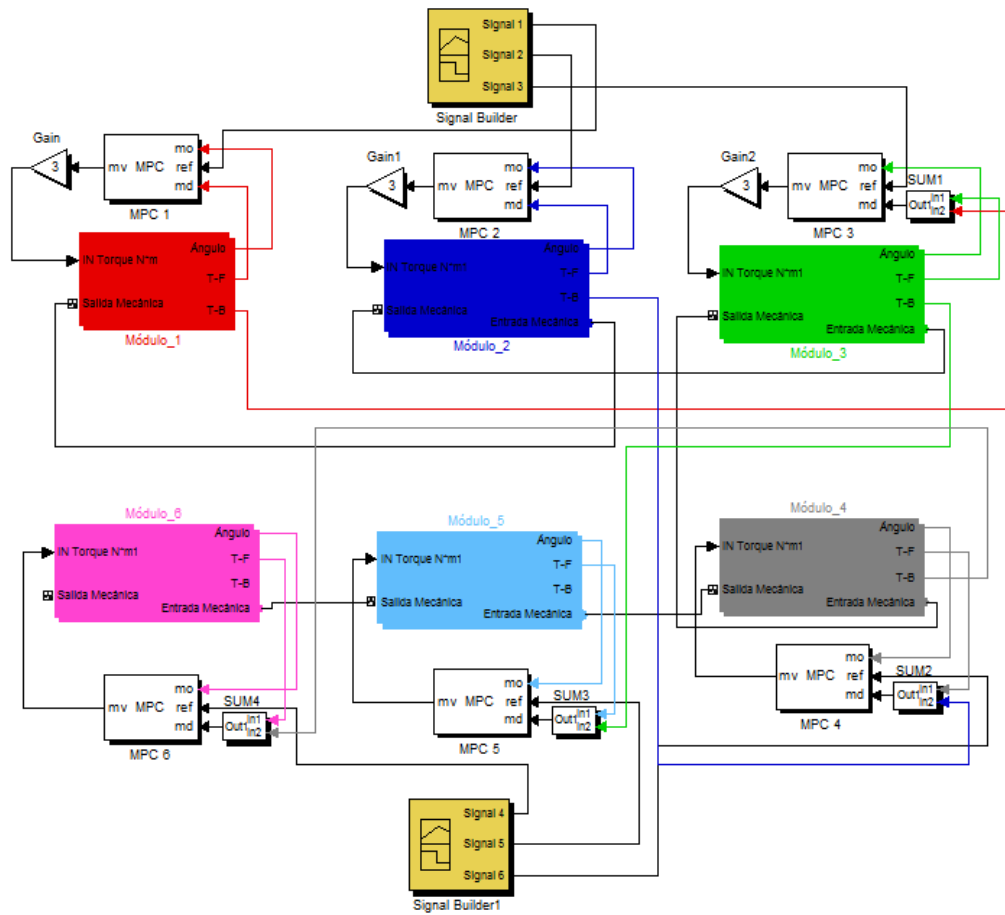
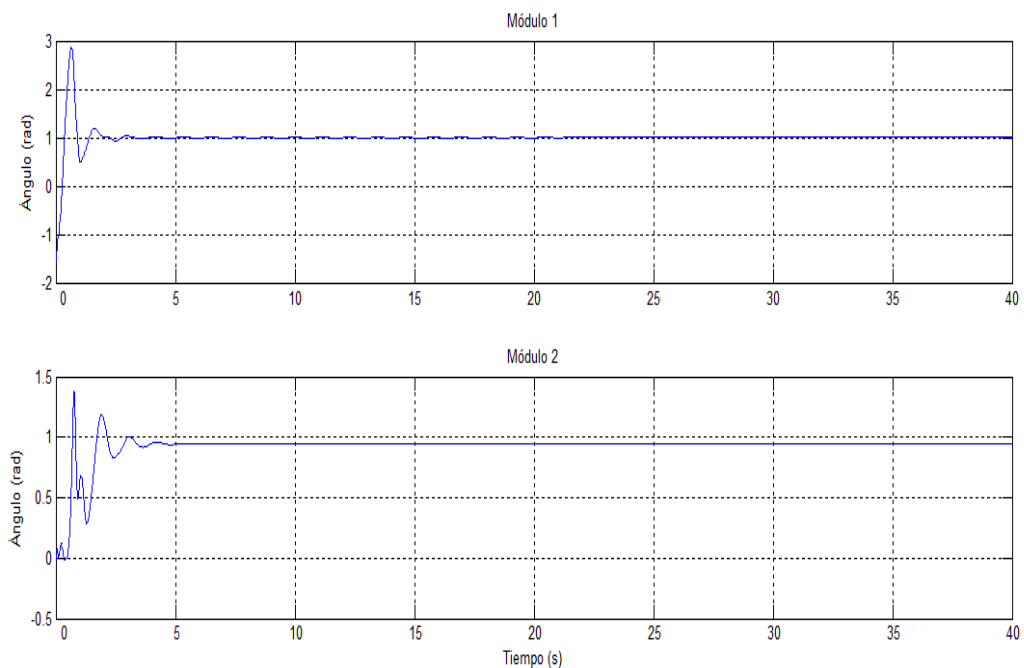


Figura 48. Diagrama de bloques con ajuste de ganancia en cada controlador

Después de analizar el comportamiento de cada controlador se concluyó que no todos necesitan la ganancia adicional, dejando así sólo los 3 primeros con esta ganancia. Los parámetros de esta simulación son, horizonte de predicción igual a 4 y horizonte de control igual a 2. Los resultados se muestran en la figura 49.



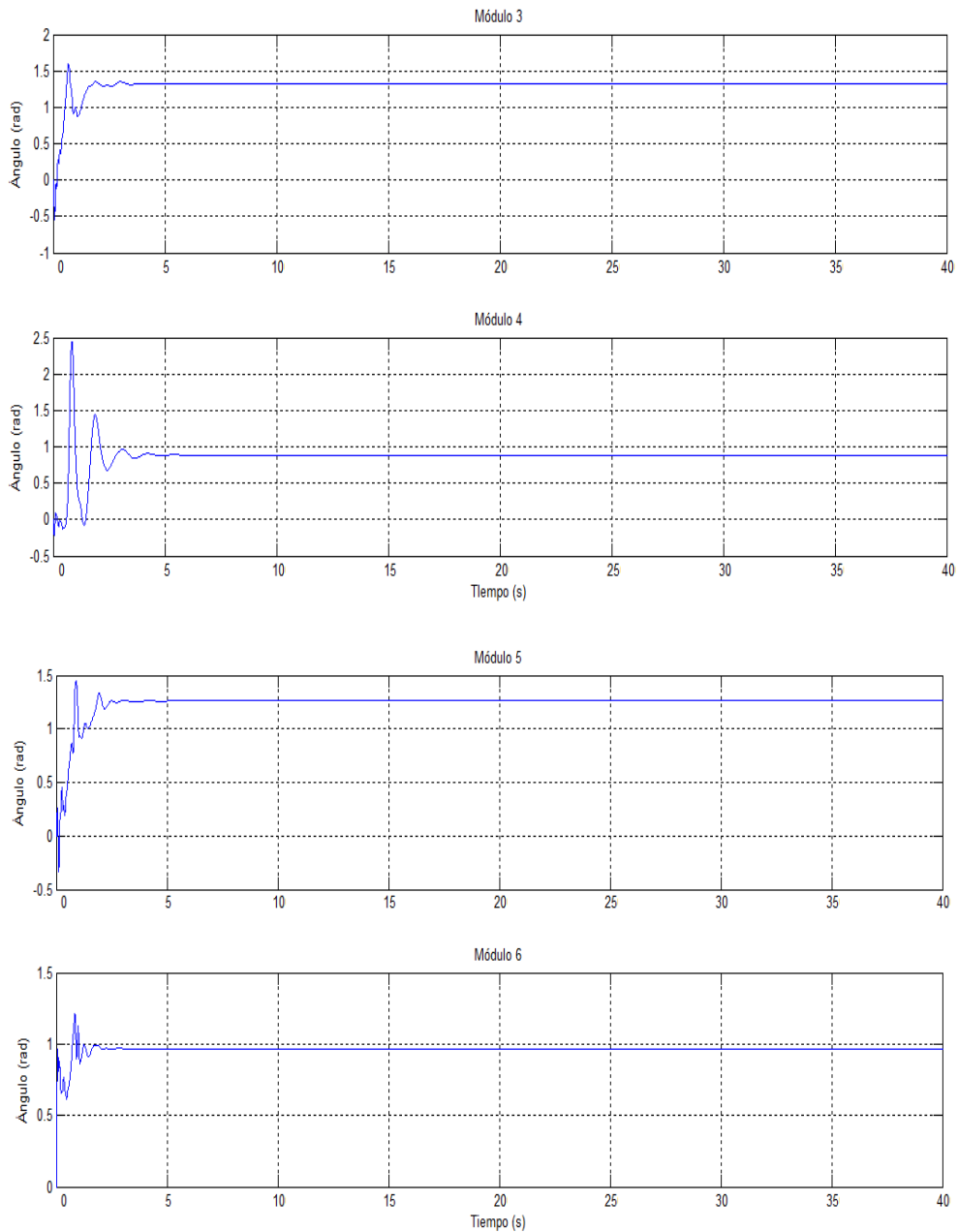


Figura 49. Gráficas de posición para cada módulo luego de Ajustar la Ganancia

Como se puede ver en la figura 49, la respuesta del sistema en general mejoró logrando valores más cercanos a la referencia y tiempos de establecimiento mucho menores tal y como se ve en la tabla 10. Sin embargo, en el siguiente capítulo se va a tratar de optimizar esto variando los parámetros de control predictivo que se presentaron el capítulo 4.

	Ts Sin Ganancia (s)	Ts con Ganancia (s)	Ref. sin Ganancia (rad)	Ref con Ganancia (rad)
Modulo 1	15	4	1	1
Modulo 2	12	5	-0,7	0,98
Modulo 3	15	3	0,5	1.2
Modulo 4	12	5	0.4	0,95
Modulo 5	10	3	0.9	1,3
Modulo 6	1,5	1,5	1	1

Tabla 10. Valores Obtenidos con el Control UMPC distribuido para 6 Módulos

6. PRUEBAS DE DESEMPEÑO

En los capítulos anteriores se muestra la implementación del controlador predictivo y de cómo se distribuyó para poder ser aplicado a la cadena robótica. Sin embargo, es necesario evaluar el desempeño de dichos controladores bajo distintas condiciones las cuales serán propuestas a lo largo de este capítulo.

6.1 DESEMPEÑO PARA DIFERENTES HORIZONTES DE PREDICCIÓN

En esta sección, se evaluará el desempeño del controlador GPC y UMPC para diferentes horizontes de predicción.

6.1.1 Control GPC y un solo módulo modelo teórico

Con el modelo dinámico teórico obtenido en el capítulo 3 y con el control GPC diseñado en el capítulo 4 se escogen horizontes de predicción $n_y = 4, 6, 10$ y 20 y un horizonte de control $n_u = 2$. Aunque en el capítulo 4 y el capítulo 5 se llega a la conclusión que el control UMPC muestra un mejor desempeño, se simula el control GPC nuevamente para poder hacer un análisis sobre el desempeño de esta clase de control predictivo, llegando a los resultados de las figuras 50 y 51.

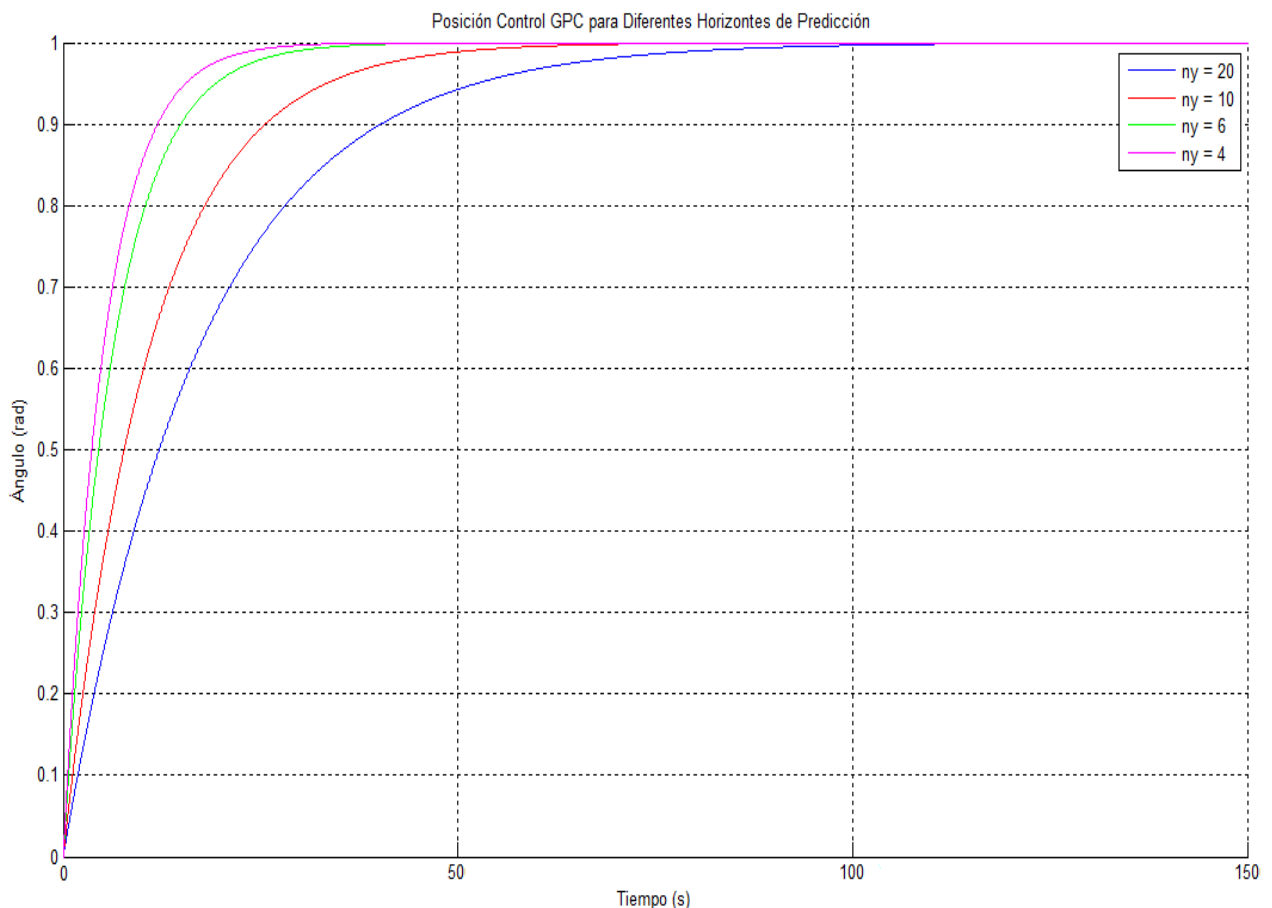


Figura 50. Posición Modelo Teórico Control GPC para Diferentes Horizontes de Predicción

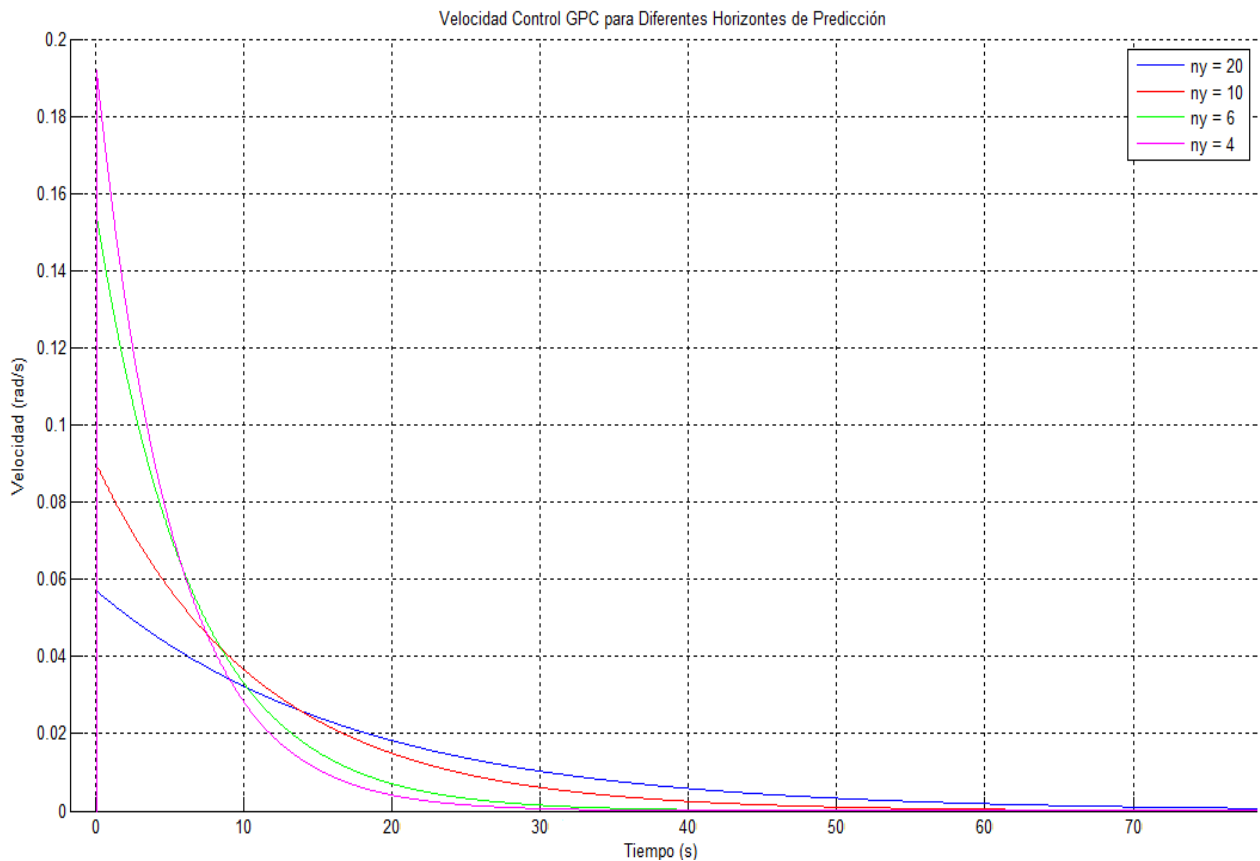


Figura 51. Velocidad Modelo Teórico Control GPC para Diferentes Horizontes de Predicción

Como se ve en la figuras 50 y 51 a medida que el horizonte de predicción aumenta, el control sigue la referencia de una manera más lenta y el pico de velocidad es mucho menor (Ver Tabla 11). Se escogieron solo 4 horizontes de predicción debido a que las matrices de predicción a mayor horizonte mayor tamaño, haciendo que el cálculo de la ganancia del controlador tenga un mayor costo computacional. Sin embargo, se puede inferir que esta tendencia seguirá presentándose, de la misma manera, por lo tanto se puede concluir que a un mayor horizonte un mayor tiempo de respuesta.

Horizonte	Tiempo de Establecimiento (s)	Sobrepico de Velocidad	Tiempo de Cálculo del controlador (s)
4	35	0.19	0.0042
6	40	0.152	0.06
10	70	0.089	0.1155
20	105	0.058	0.2944

Tabla 11. Desempeño del Controlador GPC Para diferentes Horizontes De Predicción

6.1.2 UMPC

Ahora, para el control UMPC y el modelo mecánico se escogen los mismos horizontes de predicción usados en la sección 6.1.1. Obteniendo los resultados que pueden observarse en las figuras 52, 53 y 54.

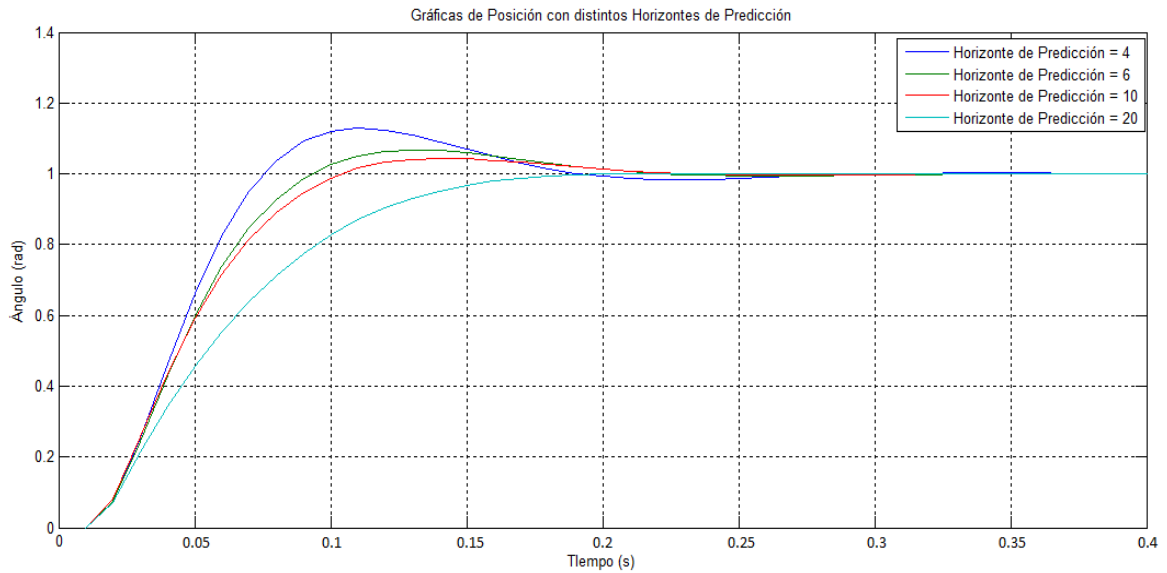


Figura 52. Gráficas de Posición Modelo Mecánico usando Control UMPC con diferentes Horizontes de Predicción

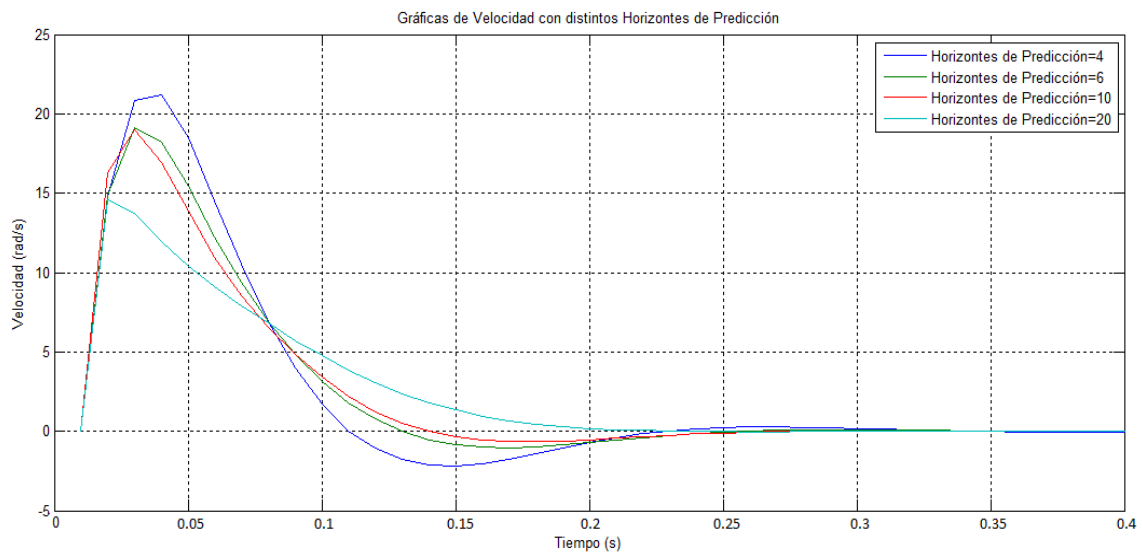


Figura 53. Gráficas de Velocidad Modelo Mecánico usando Control UMPC con diferentes Horizontes de Predicción

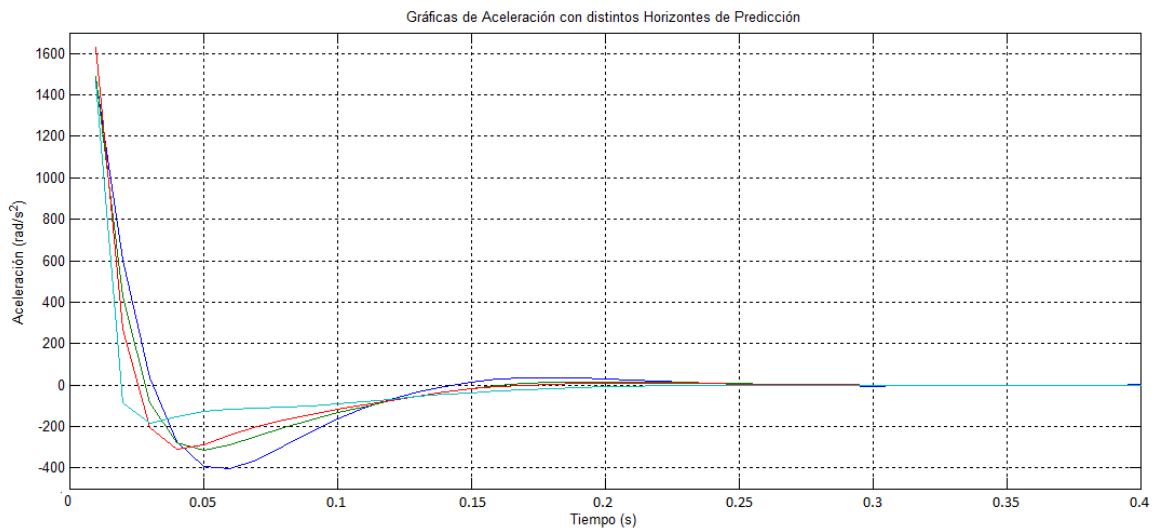


Figura 54. Gráficas de Aceleración Modelo Mecánico usando Control UMPC con diferentes Horizontes de Predicción

Como se puede ver en las figuras 52, 53 y 54 para el control UMPC a medida que el horizonte de predicción aumenta también lo hace el tiempo en el que se alcanza el máximo sobrepico. (Ver tabla 12).

Horizonte de Predicción	Ts (s)	Mp	Tp (s)
4	0.2	13%	0.1
6	0.25	6.8%	0.125
10	0.25	4.4%	0.13
20	0.2	0%	NA

Tabla 12. Desempeño del Controlador UMPC Para diferentes Horizontes De Predicción

6.2 DESEMPEÑO CONTROLADOR CON DIFERENTES PERTURBACIONES

Como se mencionó en el capítulo 4, el controlador que fue diseñado estará frente a distintas perturbaciones, por lo que es necesario ver el desempeño del mismo en distintos posibles escenarios.

6.2.1 Modelo teórico y control GPC con perturbaciones tipo escalón en la salida de posición

- Referencia $r = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ con una perturbación de amplitud 1 efectiva desde el tiempo $t = 100$ segundos.

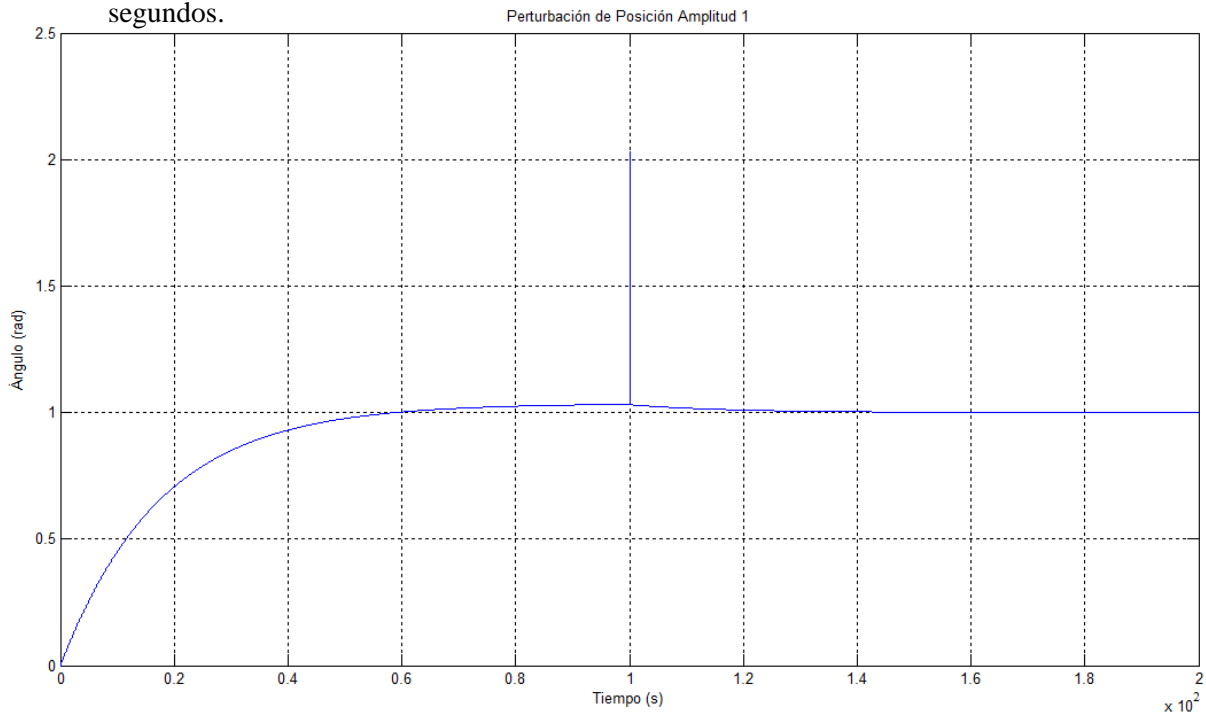


Figura 55. Gráfica Posición con Perturbación Posición Amplitud 1

Ahora si se hace una ampliación de la perturbación de la figura 55, se obtiene:

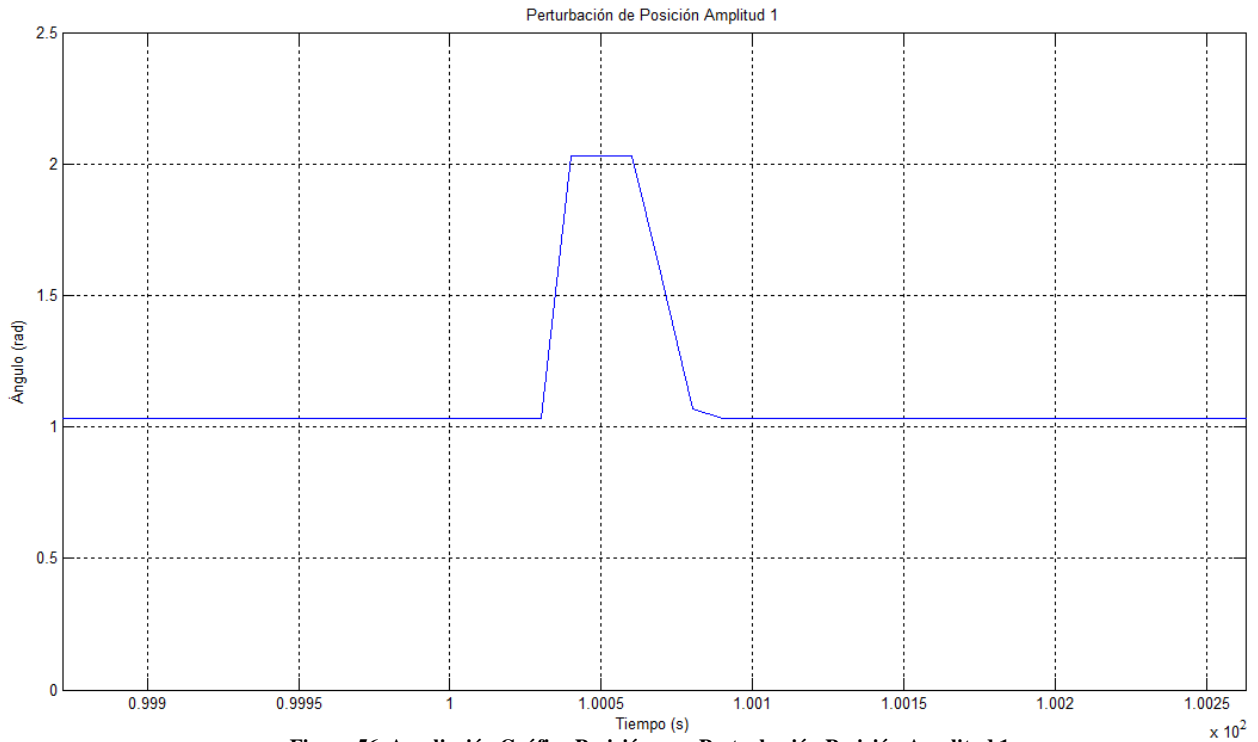


Figura 56. Ampliación Gráfica Posición con Perturbación Posición Amplitud 1

Y la velocidad es:

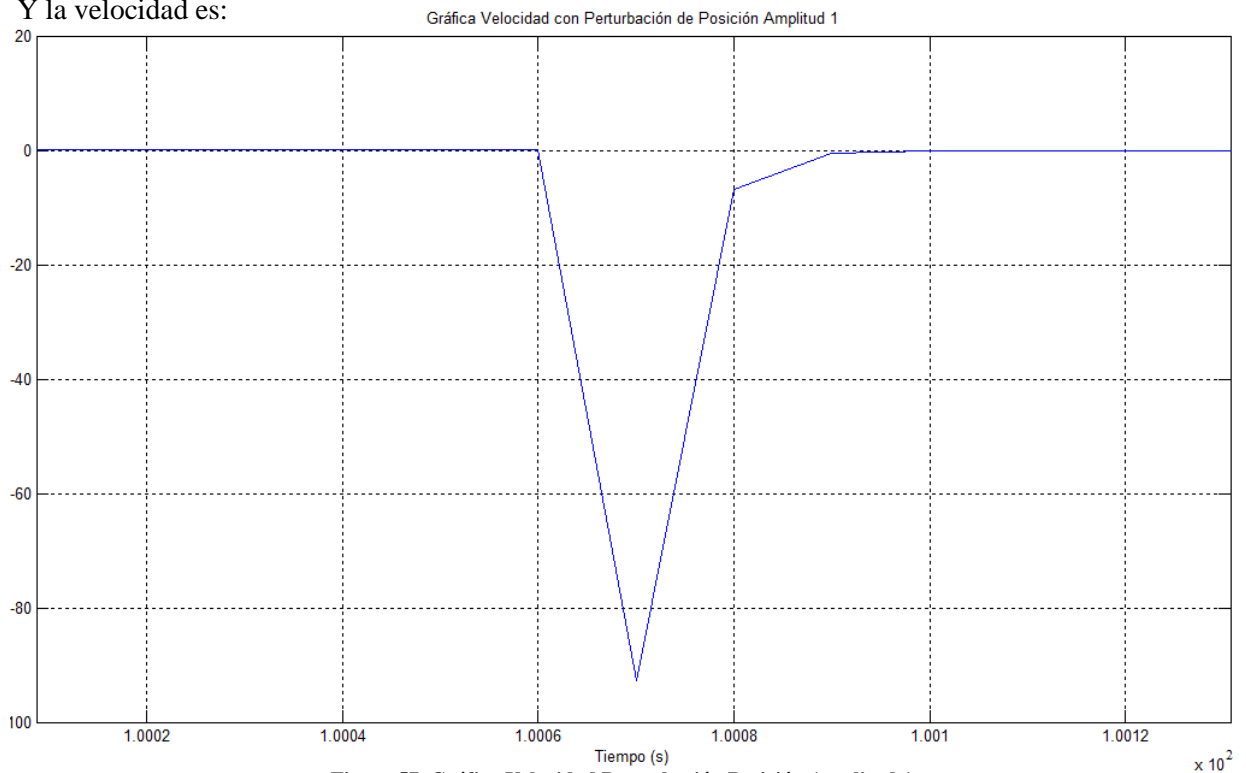


Figura 57. Gráfica Velocidad Perturbación Posición Amplitud 1

6.2.2 Modelo mecánico y control GPC con perturbaciones tipo escalón en la salida de velocidad.

Al controlador GPC, también se le pueden presentar perturbaciones de velocidad por lo que en esta sección se hace una prueba con una señal $r = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ y una perturbación en la velocidad de amplitud 0.3 a los 90 segundos. Obteniendo los resultados que se ven en las figuras 58,59 y 60.

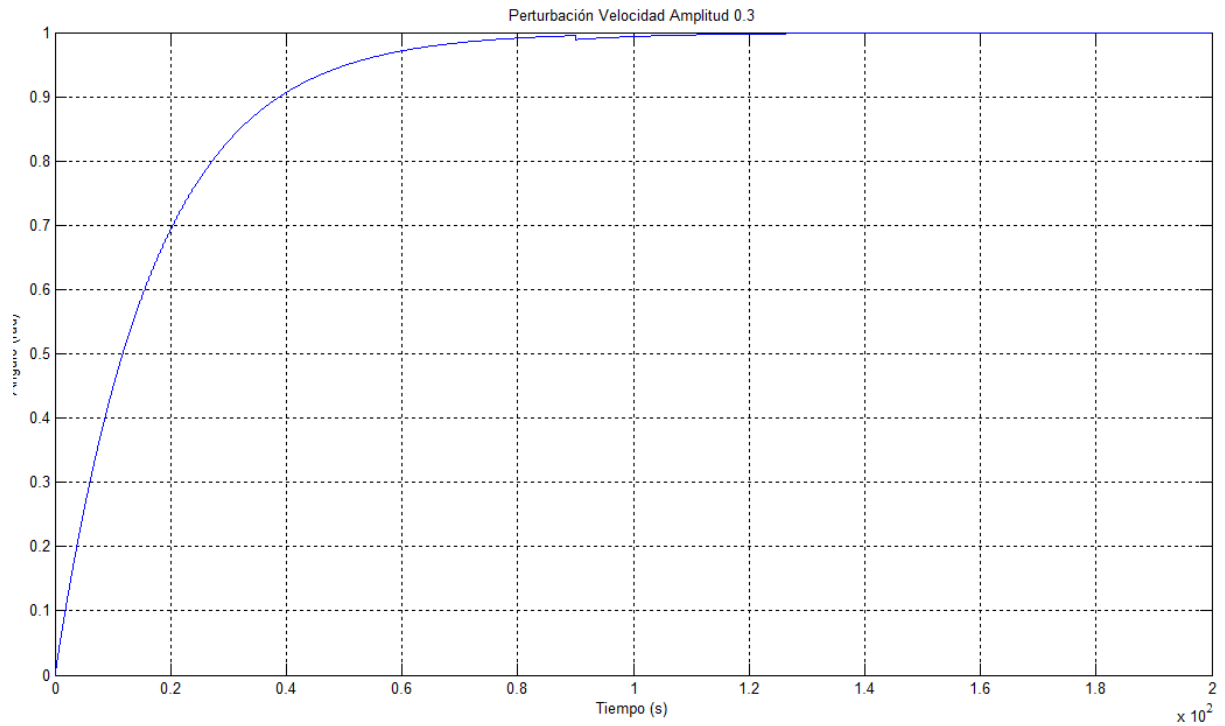


Figura 58. Posición con Perturbación Velocidad Amplitud 0.3

En la figura 58, se puede observa como a los 90 segundos cuando la perturbación de velocidad es aplicada, la posición tiene una pequeña caída la cual es corregida en 0.6 segundos tal y como se ve en la figura 59.

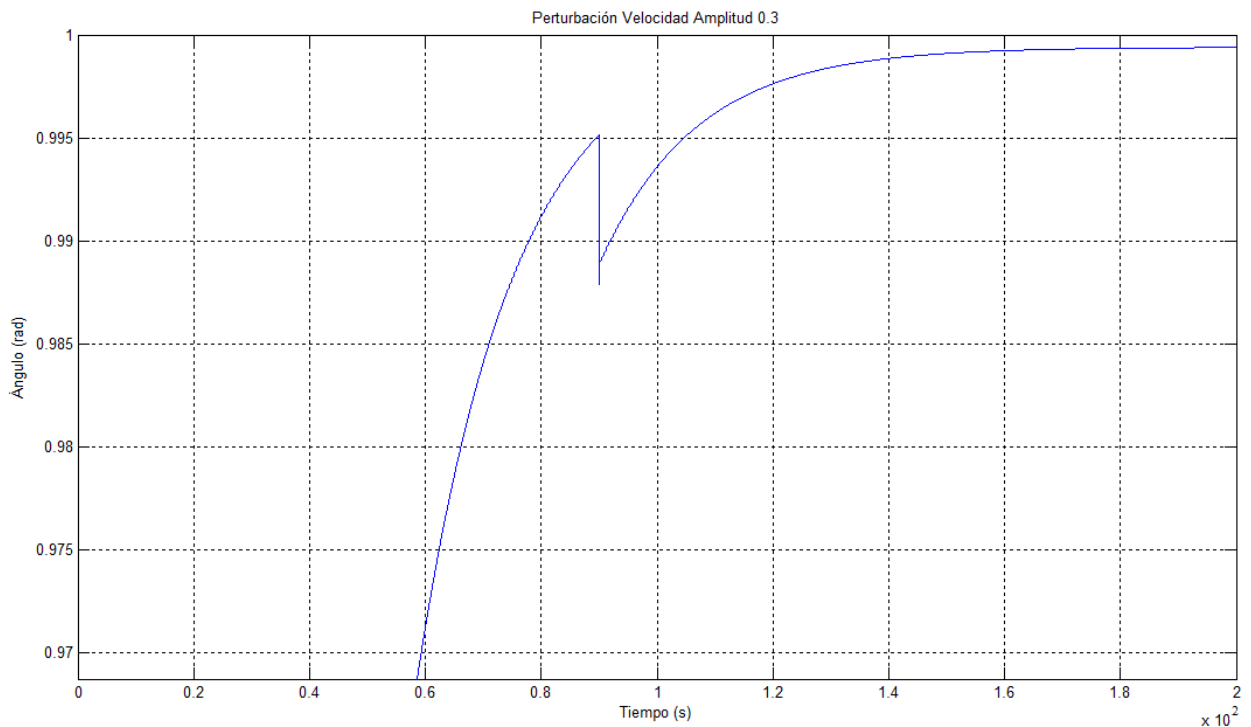


Figura 59. Ampliación Posición con Perturbación Velocidad Amplitud 0.3

Y la grafica para la velocidad del sistema se observa en la figura 60.

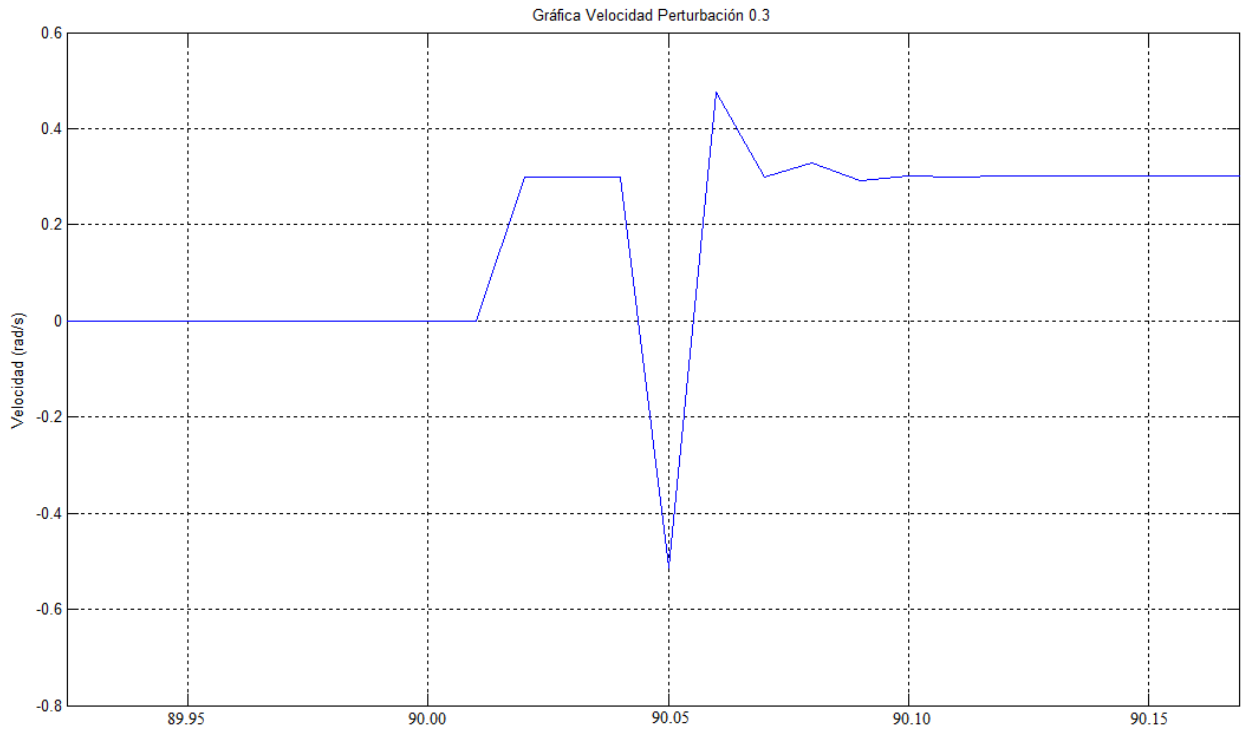


Figura 60. Velocidad con Perturbación Velocidad Amplitud 0.3

En las figuras 58, 59 y 60, se observa que una vez se aplica una perturbación de velocidad al controlador, este se encarga de corregir la posición rápidamente alcanzando nuevamente el valor de la referencia. Para el caso de la velocidad, una vez es aplicada dicha perturbación, esta toma el valor de la misma.

6.2.3 Modelo mecánico y control GPC con perturbaciones de valor constante en la salida de posición

Al controlador GPC y el modelo mecánico con una referencia $r = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ y una perturbación de posición constante de amplitud 0.5 se obtienen los resultados de las figuras 61 y 62.

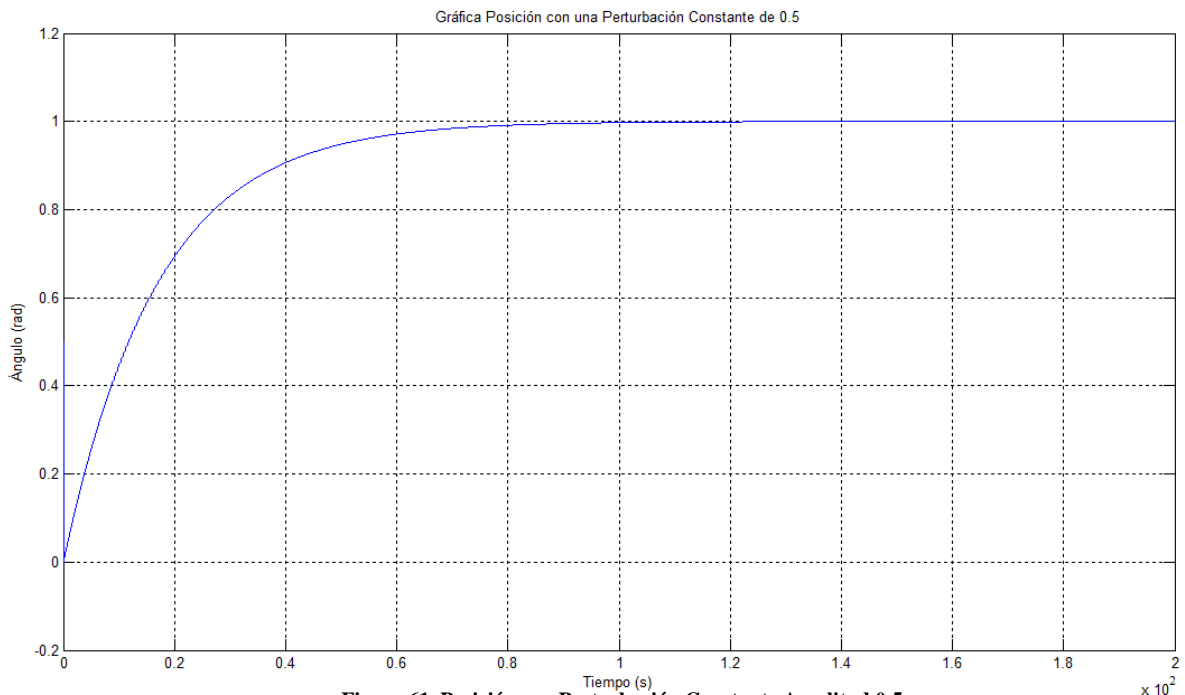


Figura 61. Posición con Perturbación Constante Amplitud 0.5

En la figura 61 se observa que el sistema sigue la referencia sin importar el valor de la perturbación constante. Sin embargo, este tiene un pico del valor de dicha perturbación a los 0 segundos, el cual es corregido rápidamente tal y como se ve en la figura 62.

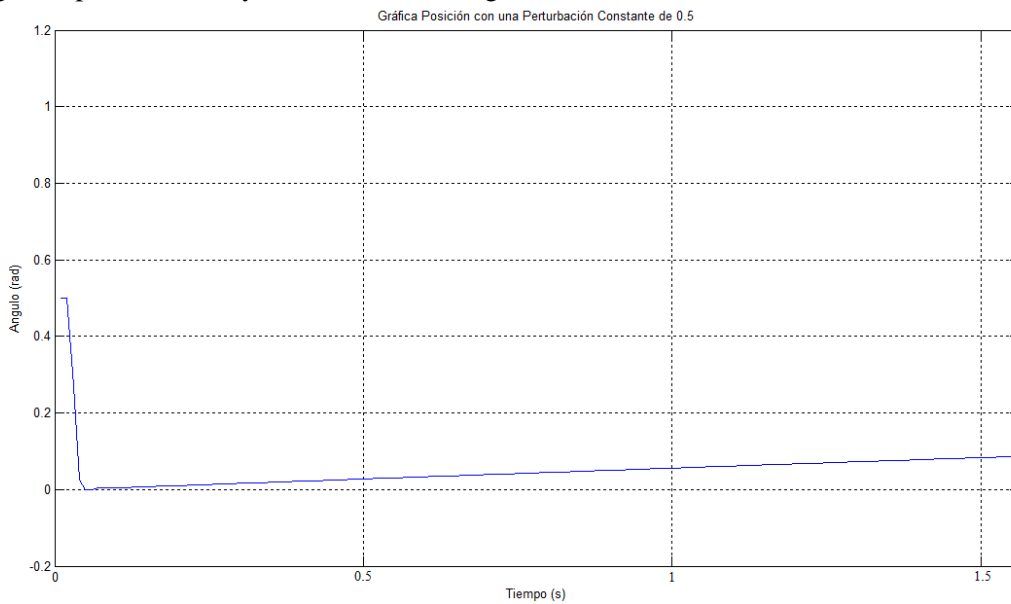


Figura 62. Ampliación Posición con Perturbación Constante Amplitud 0.5

6.3 DESEMPEÑO CONTROL DISTRIBUIDO PARA DIFERENTES HORIZONTES DE PREDICCIÓN

En esta sección se evaluara el desempeño de los controladores distribuidos para diferentes horizontes de predicción.

6.3.1 Desempeño control GPC distribuido 2 módulos y diferentes horizontes de predicción

El modelo mecánico con el controlador GPC, será simulado con diferentes horizontes de predicción. De esta manera, para una referencia $r = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ y horizontes de predicción iguales a 4, 6 y 10 se obtienen los resultados de las figuras 63,64, 65 y 66.

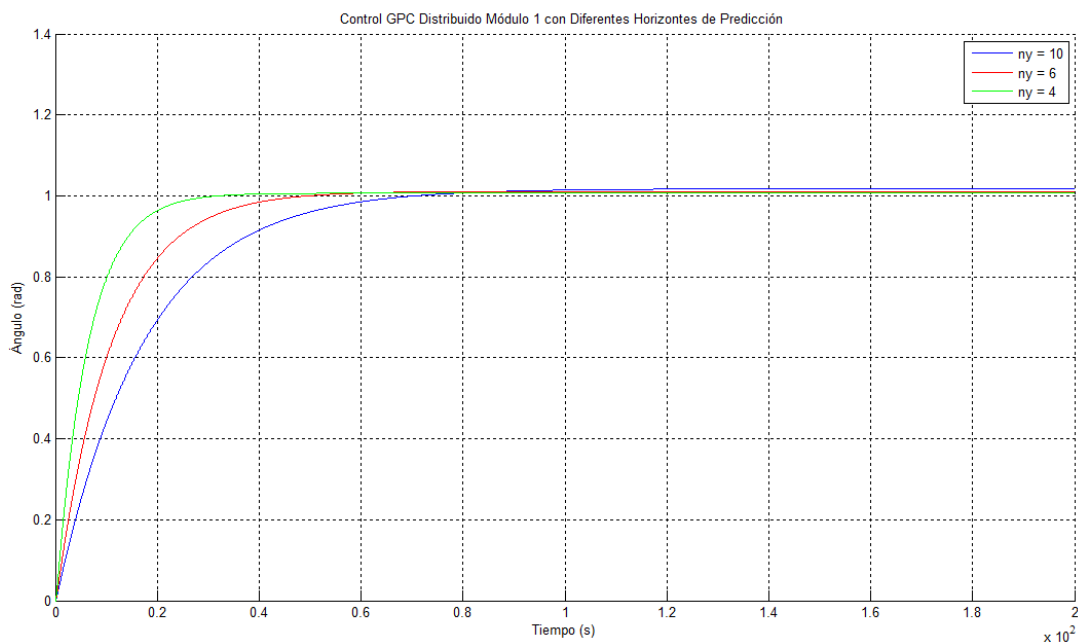


Figura 63. Gráfica Posición Módulo 1 Para diferentes Horizontes de Predicción

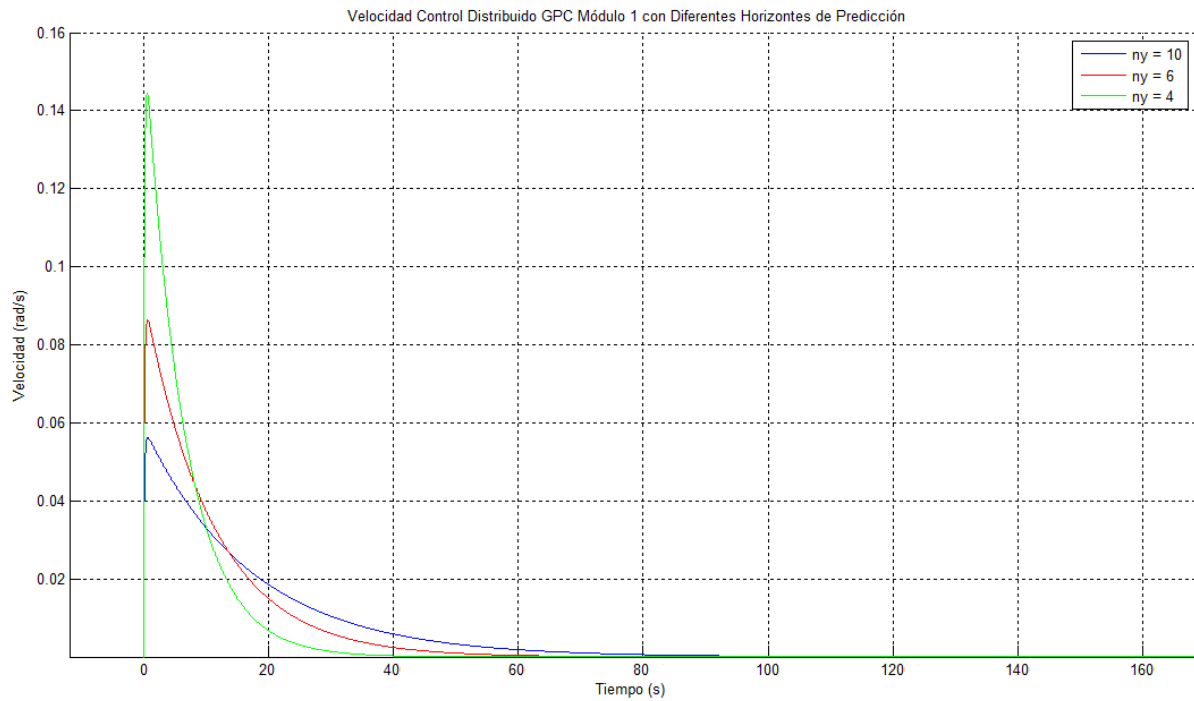


Figura 64. Gráfica Velocidad Módulo 1 Para diferentes Horizontes de Predicción

Los resultados para el módulo 2 se ven en las figuras 65 y 66.

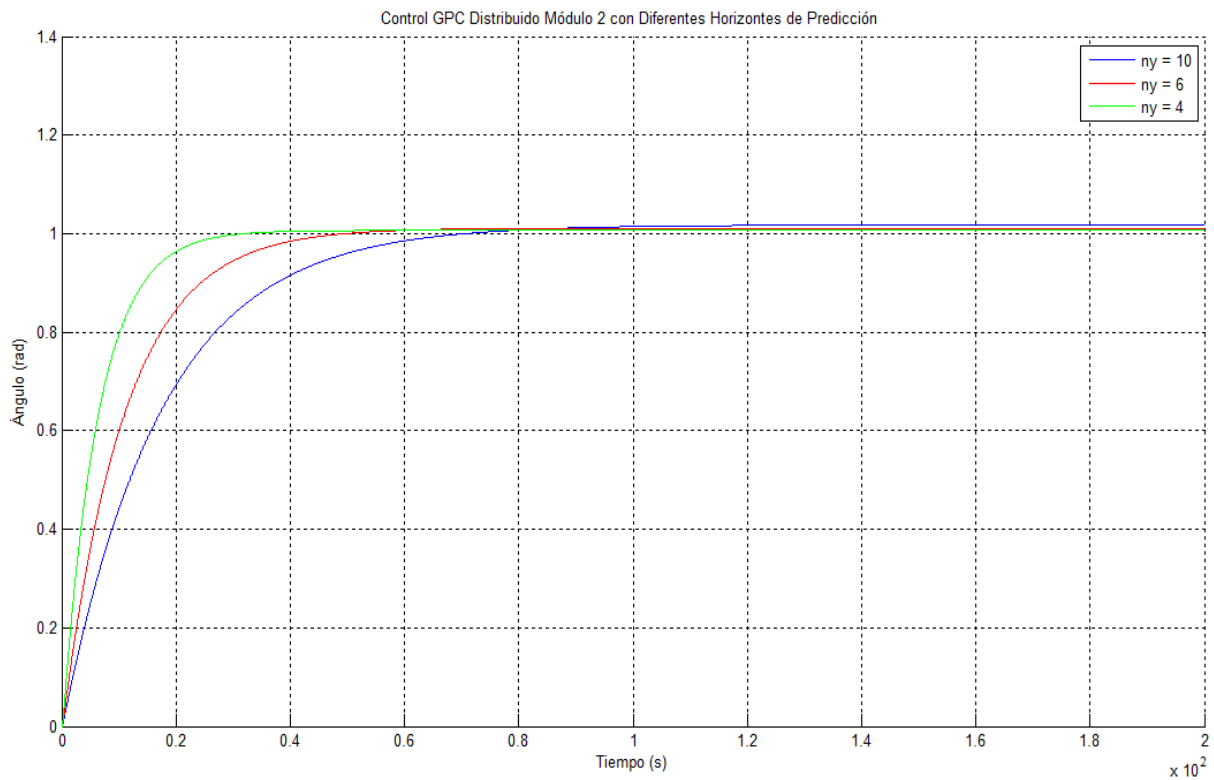


Figura 65. Gráfica Posición Módulo 2 Para diferentes Horizontes de Predicción

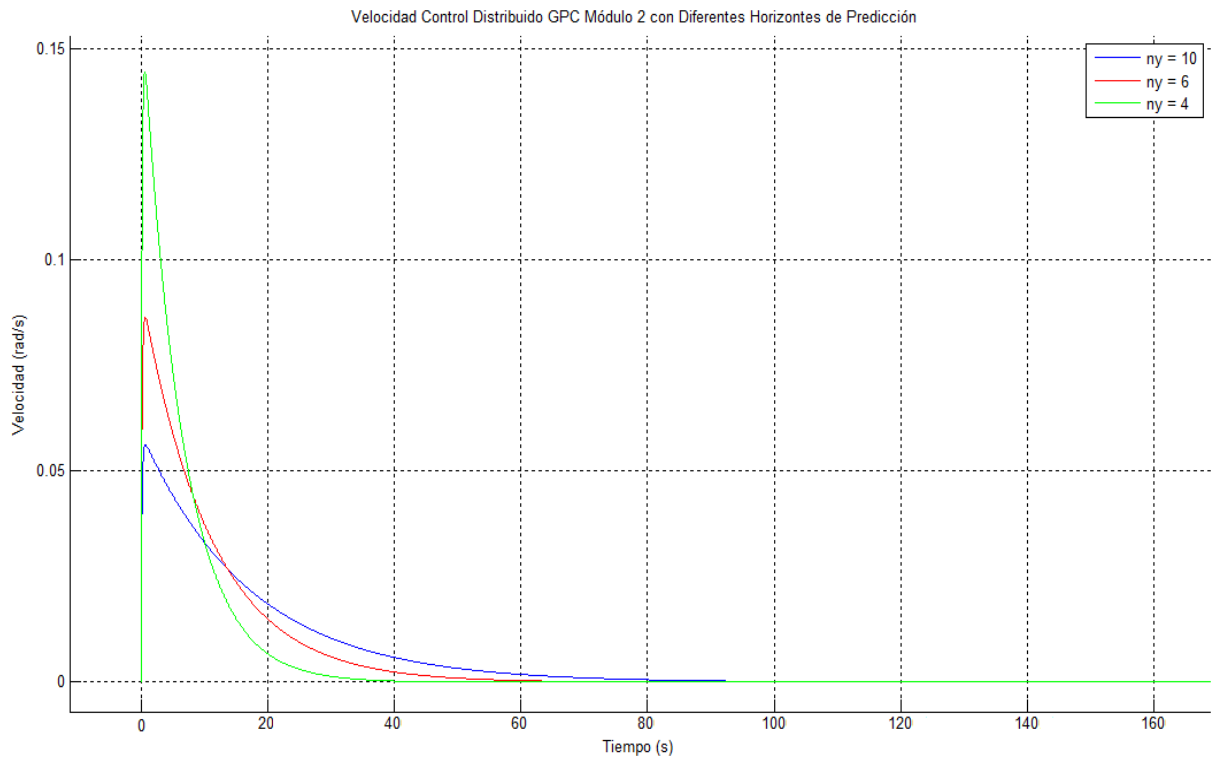


Figura 66. Gráfica Velocidad Módulo 2 Para diferentes Horizontes de Predicción

Para dos módulos, a medida que el horizonte de predicción aumenta el tiempo de establecimiento también lo hace. Este comportamiento ya se había observado anteriormente para un solo módulo donde los resultados obtenidos nos llevan a la misma conclusión.

6.3.2 Desempeño control UMPC distribuido a 2 módulos y diferentes horizontes de predicción

El modelo mecánico con el controlador UMPC, será simulado con diferentes horizontes de predicción, tomando los mismos valores y las mismas condiciones del numeral 6.3.1, llegando a los resultados de las figuras 67, 68, 69, 70,71 y 72.

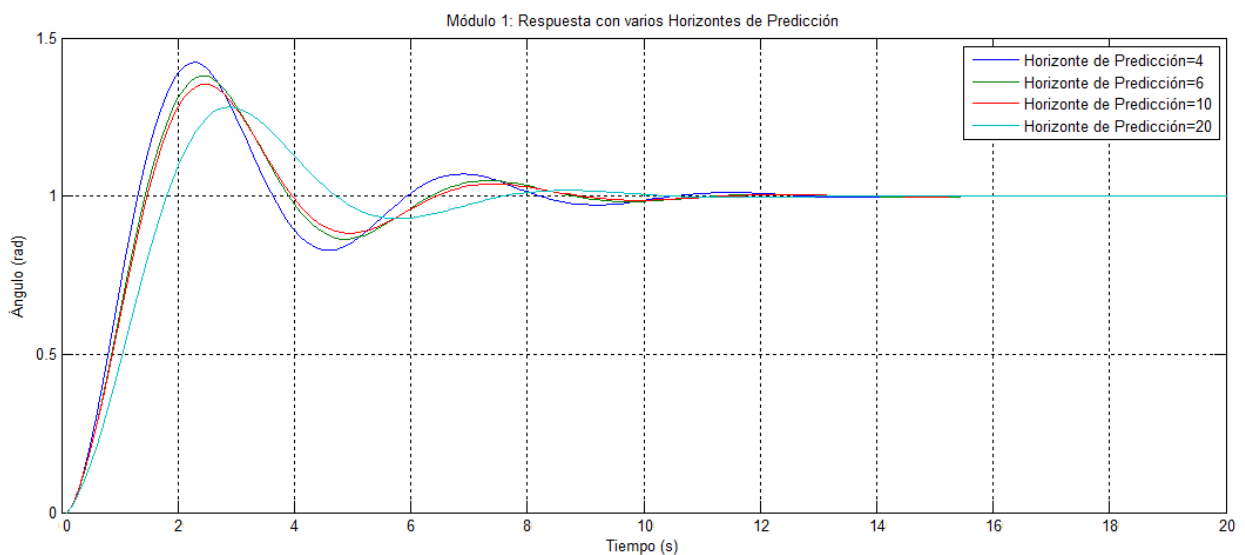


Figura 67. Salida de Posición Módulo 1 Control UMPC y diferentes horizontes de predicción

La velocidad para el módulo 1 se puede ver en la figura 68.

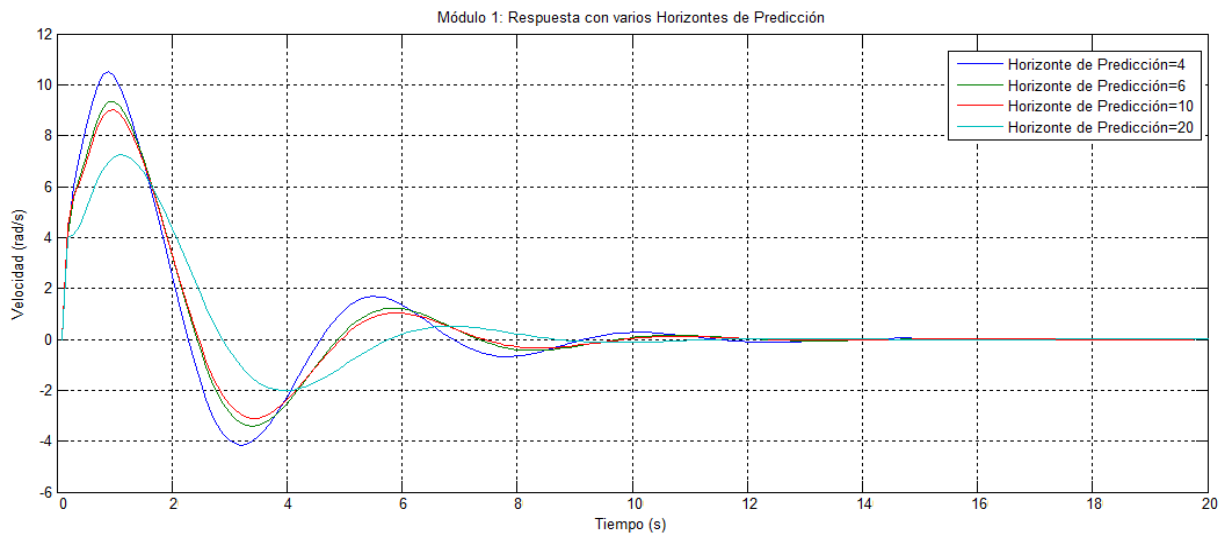


Figura 68. Salida de Velocidad Módulo 1 Control UMPC y diferentes horizontes de predicción

Como se puede ver en la Figura 67 y en la Figura 68 al aumentar el horizonte de predicción la respuesta se hace mucho más lenta y el sobrepico disminuye mientras que cuando el horizonte de predicción es muy pequeño la respuesta es mucho más rápida pero así mismo el sobrepico aumenta. Lo mismo pasa para el módulo 2 cuya respuesta se grafica en las figuras 69 y 70.

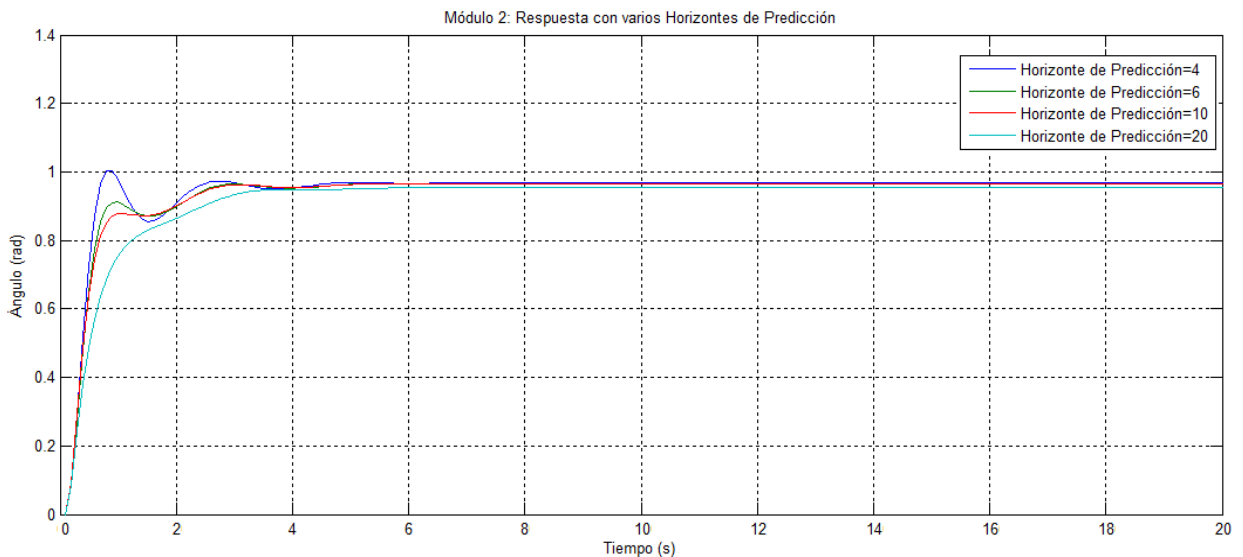


Figura 69. Salida de Posición Módulo 2 Control UMPC y diferentes horizontes de predicción

Y la gráfica de velocidad es

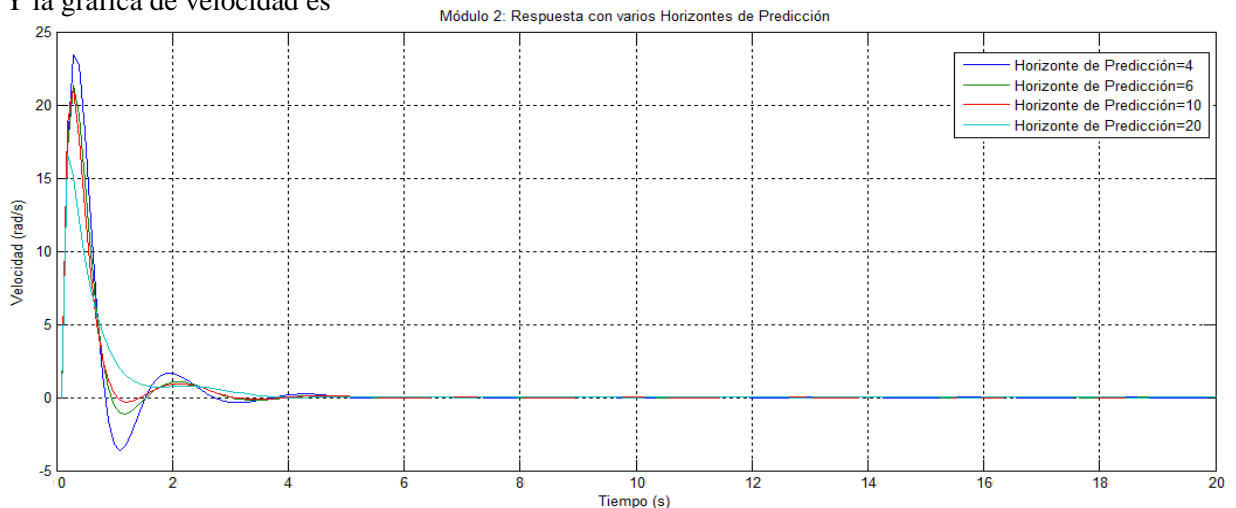


Figura 70. Salida de Velocidad Módulo 1 Control UMPC y diferentes horizontes de predicción

Por último, se grafican las señales de control y sus cambios a medida que el horizonte de predicción también lo hace.

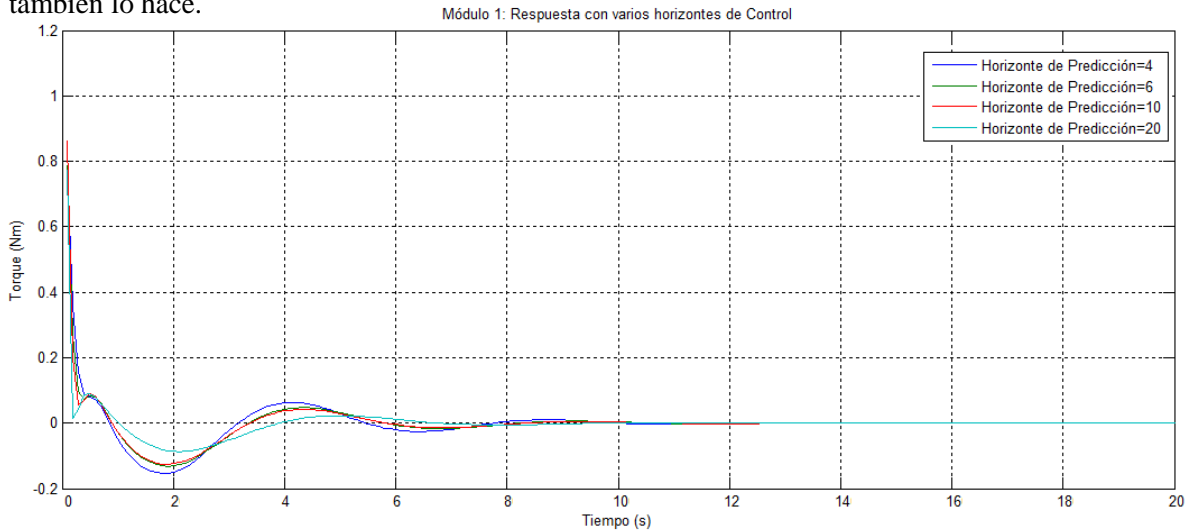


Figura 71. Módulo 1: Señal de control con varios horizontes de predicción

Y la señal de control del módulo 2 se puede observar en la figura 72.

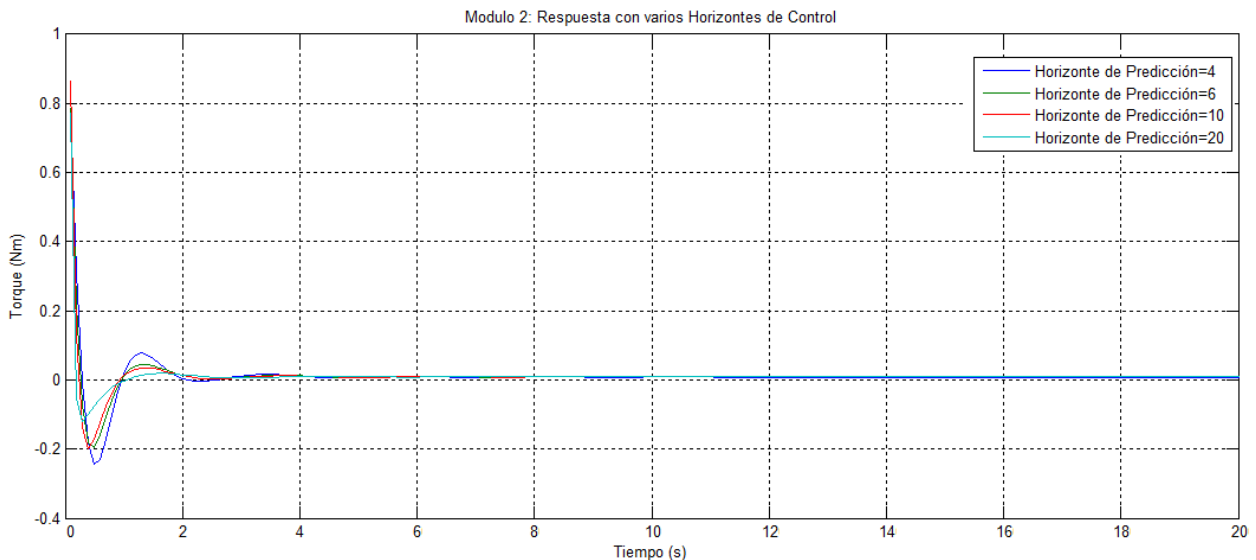


Figura 72. Módulo 1: Señal de control con varios horizontes de predicción

Es así, como la posición, la velocidad y la aceleración tiene una respuesta más rápida para los horizontes de predicción de un valor relativamente pequeño. De acá en adelante, en las siguientes pruebas que se realicen, se va a utilizar un horizonte de predicción de 10, el cual es un punto intermedio entre rapidez y sobrepico.

6.4 DESEMPEÑO DEL CONTROLADOR UMPC VARIANDO LOS PESOS DE SALIDA Y LOS PESOS DE ENTRADA

Tal y como se mencionó en el capítulo 4, otro de los parámetros que también pueden modificarse para el diseño del control UMPC son los pesos de entrada y los pesos de salida, por lo que en esta sección se muestran los resultados de variar dichos pesos.

6.4.1 Control UMPC con diferentes pesos de entrada

Para un solo módulo, una referencia $r = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ y unos pesos de entrada iguales a 1, 2, 5 y 10 se obtienen los resultados de las figuras 73 y 74.

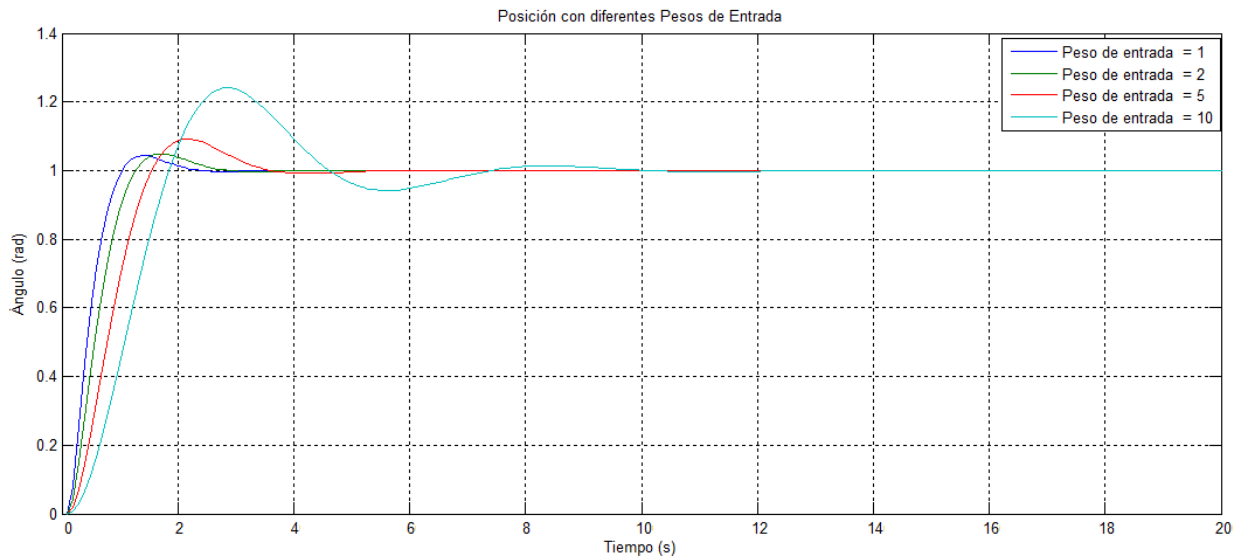


Figura 73. Salida de posición Control UMPC diferentes pesos de Entrada

Y donde la señal de control es:

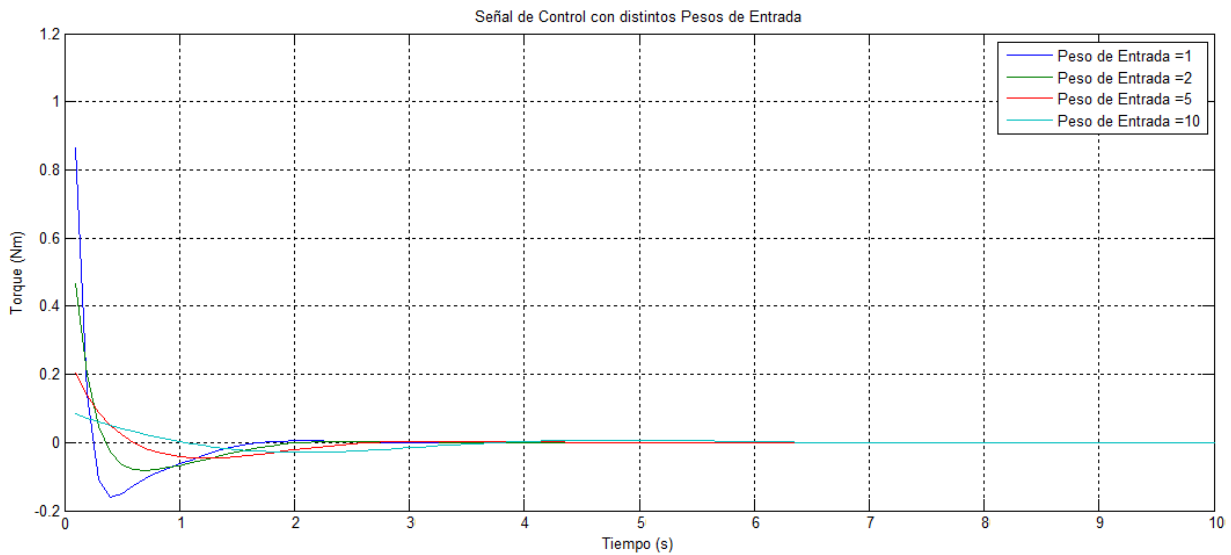


Figura 74. Señal de Control UMPC para diferentes pesos de Entrada

De la figura 74 se puede concluir tal y como se dijo en el capítulo 4 que, si el valor de peso de entrada es diferente de cero, la variable manipulada está más cerca a su valor nominal, eliminando así el error en estado estacionario de la salida de la planta. Los resultados obtenidos se ven en la tabla 13.

Peso de Salida	Peso de Entrada	Ts (s)	Mp	Tp (s)
1	1	0.25	4.5%	0.13
1	2	0.32	5.5%	0.17
1	5	0.4	9.1%	0.2
1	10	0.8	24%	0.3

Tabla 13. Desempeño del controlador UMPC variando los pesos de entrada.

6.4.2 Control UMPC con diferentes pesos de Salida

Para un solo módulo, una referencia $r = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ y unos pesos de salida iguales a 1, 2, 5 y 10 se obtienen los resultados de la figura 75.

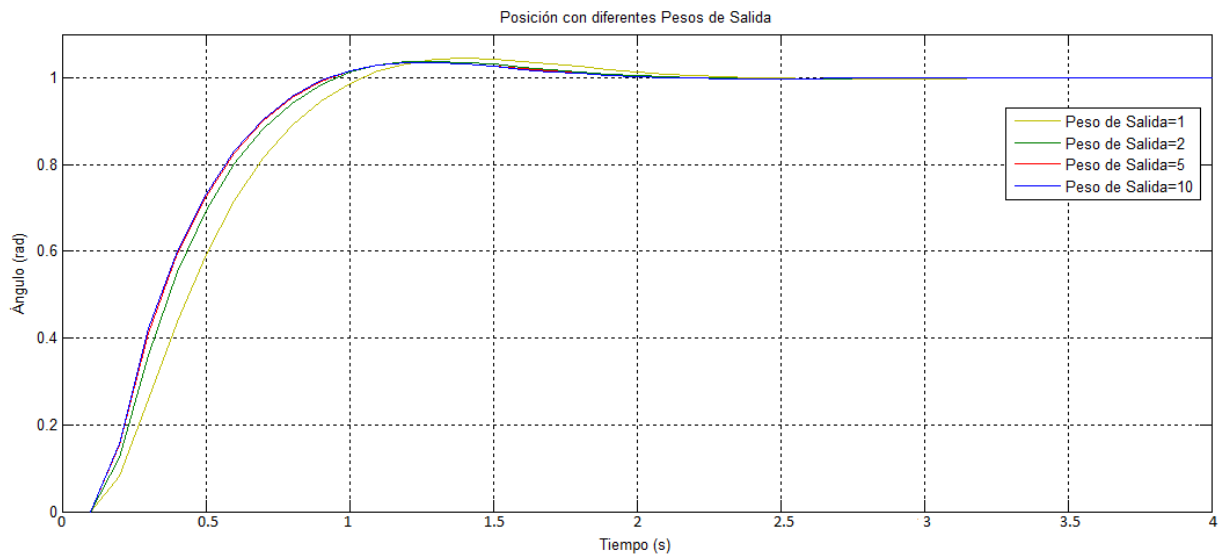


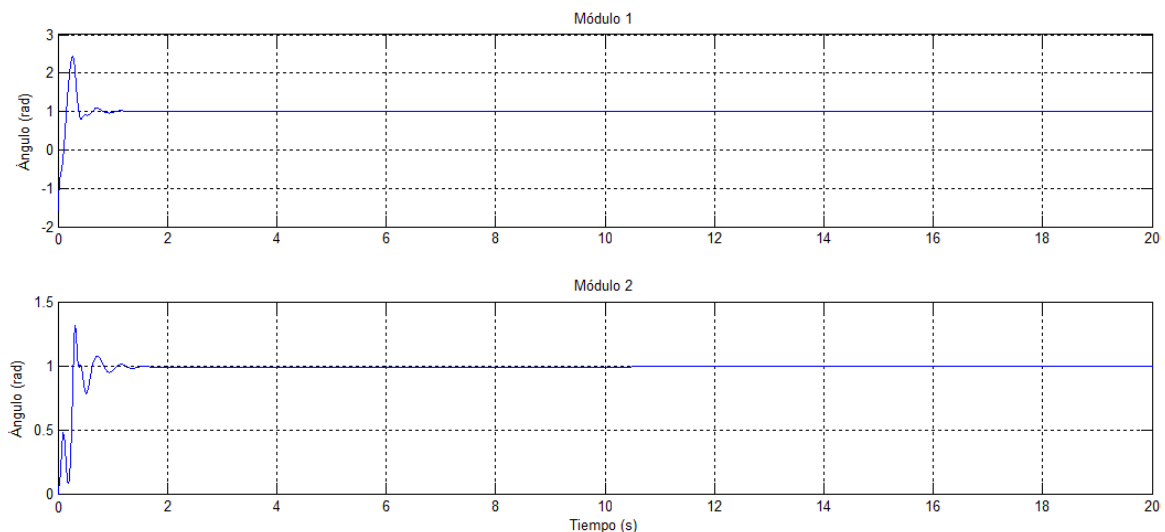
Figura 75. Salida de posición Control UMPC diferentes pesos de Salida

Para este caso, a medida que los pesos de salida aumentan la respuesta del controlador también lo hace por lo que sigue la referencia más rápidamente. Los resultados obtenidos se resumen en la tabla 14.

Peso de Salida	Peso de Entrada	Ts (s)	Mp	Tp (s)
1	1	0.25	4.5%	0.14
2	1	0.23	3.8%	0.12
5	1	0.2	3.5%	0.12
10	1	0.2	3.3%	0.11

Tabla 14. Desempeño del controlador UMPC variando los pesos de entrada.

Por último, se escoge el controlador que tiene el mejor desempeño con los siguientes parámetros de simulación: horizonte de predicción igual a 10, horizonte de control igual a 2, pesos de entrada igual 2 y peso de salida igual a 10. La respuesta del control distribuido para seis módulos diseñado con los parámetros anteriores se observa en la figura 76.



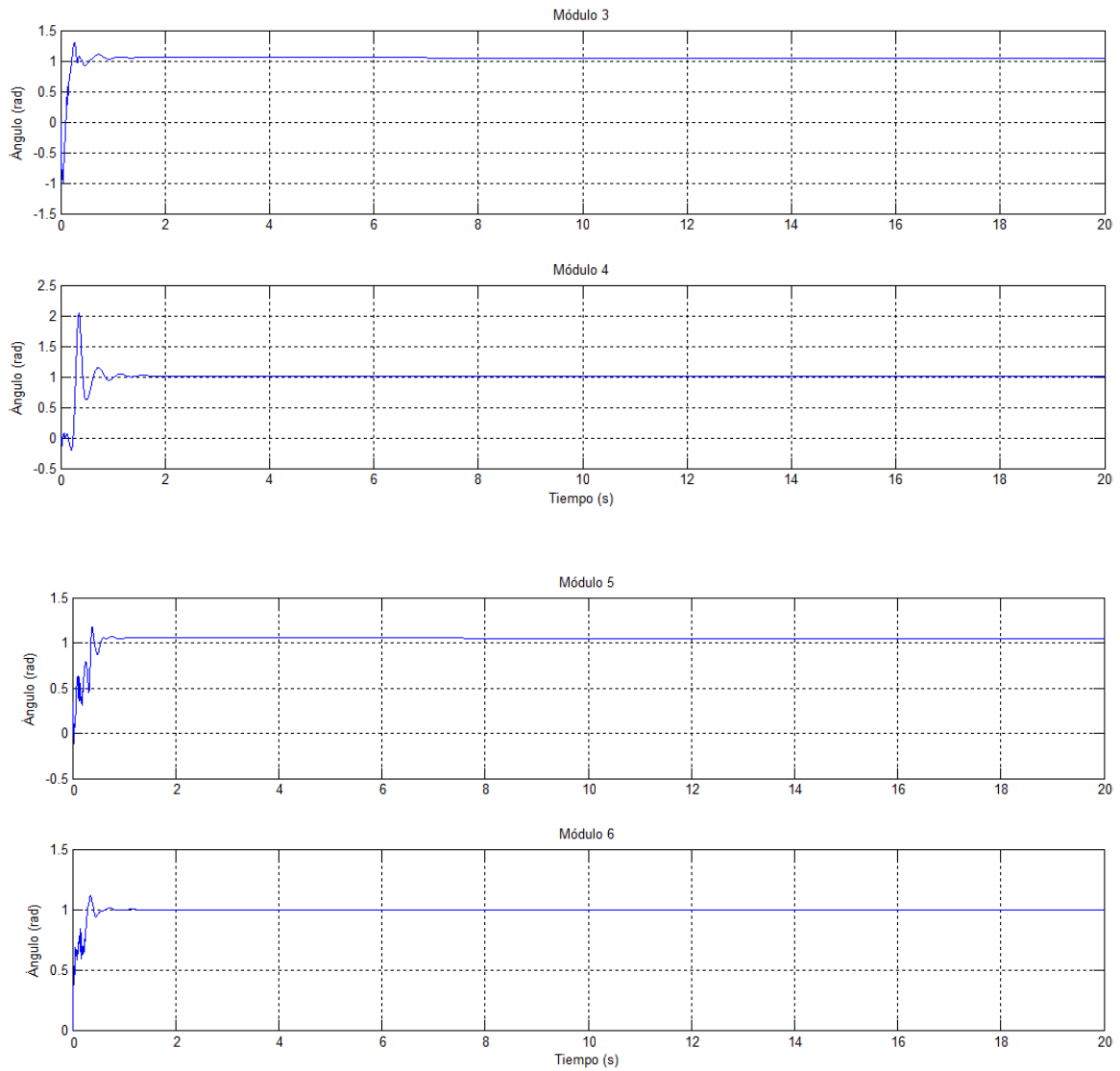


Figura 76. Control UMPC distribuido a seis módulos con los mejores parámetros de simulación

Como se observa en la figura 76, todos los módulos alcanzan la referencia deseada la cual es 1 para todos. Al comparar esta respuesta con la respuesta de la figura 49, se ve que el controlador de la figura 76 tiene un mejor desempeño el cual se encontró luego de realizar las distintas pruebas del capítulo 6.

7. CONCLUSIONES

El controlador predictivo distribuido diseñado en este trabajo tiene como objetivo controlar una cadena cinemática compuesta por seis módulos. De esta manera, el control desarrollado cumple con los objetivos planteados y muestra un buen desempeño de acuerdo con los resultados obtenidos en las pruebas que fueron realizadas.

Es así, como la obtención del modelo dinámico del sistema es uno de los pasos más importantes en el diseño de esta clase de controladores debido a que como se menciona a lo largo de este libro, de la complejidad del modelo dependerá la complejidad del controlador. Para sistemas mecánicos, el uso de software de simulación tal como SimMechanics® de Matlab permite la obtención de modelos que describen de forma correcta la dinámica del sistema.

La creación de modelos en SimMechanics® presenta la ventaja de poder simular sistemas con una gran cantidad de elementos mecánicos, sin tener que recurrir al modelo matemático de los mismos. Esto resulta de gran utilidad para la simulación de sistemas robóticos con múltiples grados de libertad donde generalmente las expresiones matemáticas resultantes cuentan con una gran cantidad de términos los cuales resultan poco prácticos al momento de ser simulados, tal y como pasa con el modelo teórico de la cadena robótica de seis módulos.

El diseño del controlador a partir de los modelos obtenidos en SimMechanics y por medio de la formulación de Euler-Lagrange, resulta fácil de implementar debido a que Matlab permite la interconexión de elementos del toolbox MPC con los elementos de los modelos. Una vez se crea el controlador, se puede observar que el control UMPC (Unconstrained Model Predictive Control) tiene un mejor desempeño en cuanto a tiempos de establecimiento (casi 80 veces menor) respecto al control GPC (Generalised Predictive Control) tal y como se ve en el capítulo 4 de este libro, aunque la respuesta con el control UMPC presenta un mayor sobrepico.

Para el control UMPC y el GPC a medida que el horizonte de predicción aumenta el tiempo de establecimiento también lo hace. En este punto, es importante resaltar que la implementación en simulaciones del control UMPC resulta mucho más práctica debido a que no importa cuál sea el horizonte de predicción el diseño del controlador toma el mismo tiempo, mientras que con el control GPC a medida que el horizonte de predicción aumenta, también aumenta el tamaño de las matrices de predicción las cuales deben ser calculadas manualmente, resultando poco práctico para horizontes de predicción relativamente grandes.

Los pesos de entrada y de salida también son importantes a la hora de diseñar el controlador, ya que si los pesos de salida aumentan se aumenta la cercanía de la salida de la planta con la referencia y si los pesos de entrada aumentan, la variable manipulada estará más cerca a su valor nominal, eliminando así el error en estado estacionario de la salida de la planta.

De esta manera, el control que muestra un mejor desempeño es el UMPC, sin embargo, el control GPC tiene una mejor respuesta cuando son aplicadas perturbaciones al sistema, corrigiendo la salida aproximadamente 10 veces más rápido respecto a como lo hace el control UMPC.

Una vez se distribuye el control predictivo, se comprueba la utilidad del mismo con esta clase de sistemas, debido a que no solo sigue las referencias indicadas sino que también lo hace en un menor tiempo. Esto se debe a que todos los controles trabajan de forma independiente tratando de lograr un objetivo común. Es importante aclarar, que a cada uno de los módulos le es asignada una posición en particular la cual debe seguir, esta posición generalmente es indicada por un generador de trayectorias, sin embargo, como esta cadena no se desplaza, las posiciones pueden ser indicadas manualmente evitando así un gasto innecesario como lo sería dicho generador de trayectorias.

Como trabajo futuro, se propone modificar el controlador de tal forma que pueda controlar posiciones que sean variantes en el tiempo, es decir, cuando la cadena se esté desplazando. Para esto, es necesario validar nuevamente el modelo aplicando señales a cada uno de los servomotores para así observar si tiene el mismo comportamiento del modelo que ha sido simulado. Por otro lado deben tenerse en

cuenta restricciones de torque ya que según las hojas de especificaciones del servomotor dynamixel AX-12¹⁶ solo puede tomar valores entre 12 kgf.cm y 16.5 kgf.cm o en el SI 1.18 Nm y 1.62 N.m, por lo que seguramente la entrada la señal de control deberá limitarse. Además, se propone probar este controlador para cadenas que cuenten con más de seis módulos para así poder plantear un sistema de control que sea capaz de manipular cadenas con un número indefinido de módulos. Una vez se realicen las propuestas anteriores, este sistema de control podría ser implementado físicamente para así ver el funcionamiento del mismo y las restricciones que se le presentarían en el mundo real.

¹⁶ Hoja de Especificaciones Dynamixel AX-12. Disponible en: [http://robosavvy.com/RoboSavvyPages/Support/Bioloid/AX-12\(english\).pdf](http://robosavvy.com/RoboSavvyPages/Support/Bioloid/AX-12(english).pdf)

8. BIBLIOGRAFÍA

- [1] Shigeo Hirose and Hiroya Yamada, “Snake-Like Robots, Machine Design of Biologically Inspired Robots”, *March 2009*.
- [2] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, JOHN WILEY & SONS, INC.
- [3] Gregory S. Chirikjian and Joel W. Burdick, “A Modal Approach to Hyper-Redundant Manipulator Kinematics”, *IEEE Transactions on Robotics and, VOL. 10, NO. 3, JUNE 1994*
- [4] ZAK S.H. *Systems and Control*. New York:Oxford University Press. 2003. Capítulo 1
- [5] Eduardo F. Camacho and Carlos Bordons, *Model Predictive Control*, Springer-Verlag Berlin Heidelberg New York 1998.
- [6] J.A. Rossiter, *Model Based Predictive Control*, CRC Press, 2003. Capítulos 1,3 y 4.
- [7] A. N. Venkat, J. B. Rawlings, and S. J. Wright. Stability and optimality of distributed model predictive control. In CDC and ECC, pages 6680–6685, Dec 2005. <http://www.che.wisc.edu/tech-reports/twmcc-2006-03.pdf>.
- [8] T. Nagano, T. Kondo and K. Ito, “Distributed Motor Control System With Transmission Time Delay”, en *Communications, Control and Signal Processing*, 2008. ISCCSP 2008. 3rd International Symposium. Pp 1236-1241. 2008.
- [9] BAY J.S. *Fundamentals of Linear State Space Systems*. New York: McGraw Hill International.1999. Capítulo 11.
- [10] R. Salcedo, R. Correa, “Simulator for Model Predictive Control”, Universidad Industrial de Santander. 2008
- [11] W. B. Dunbar. Distributed receding horizon control of dynamically coupled nonlinear systems. *TAC,52(7):1249–1263, Jul 2007*.
- [12] T. Keviczky, F. Borrelli, and G. J. Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42(12):2105–2115, Dec 2006.

