

**MODELADO, SIMULACION Y CONTROL DE UN MANIPULADOR ROBOTICO DE 2 GRADOS  
DE LIBERTAD EMPLEANDO ADAMS/MATLAB**

**CARLOS IVAN MENDOZA POVEDA**



**UNIVERSIDAD PONTIFICIA BOLIVARIANA**

**BUCARAMANGA**

**2012**

**MODELADO, SIMULACION Y CONTROL DE UN MANIPULADOR ROBOTICO DE 2 GRADOS  
DE LIBERTAD EMPLEANDO ADAMS/MATLAB**

**CARLOS IVAN MENDOZA POVEDA**

**Trabajo de grado para optar por el título de  
Especialista en Control e Instrumentación Industrial**

**Director**

**Luis Ángel Silva, PhD**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA**

**BUCARAMANGA**

**2012**

**Nota de aceptación:**

---

---

---

---

---

---

---

**Firma del presidente del jurado**

---

**Firma del jurado**

---

**Firma del jurado**

**Bucaramanga, Enero 2012**

# Contenido

Introducción.....	12
1. OBJETIVOS .....	13
1.1 OBJETIVO GENERAL.....	13
1.2 OBJETIVOS ESPECÍFICOS .....	13
2. Estado del arte .....	14
2.1 Control de posición .....	14
2.1.1 Control P con retroalimentación de velocidad y control PD .....	14
2.1.2 Control PD con compensación de gravedad .....	15
2.1.3 Control PD con compensación precalculada de gravedad .....	16
2.1.4 Control PID .....	16
2.2 Control de movimiento.....	17
2.2.1 Control por precompensación y control PD con precompensación.....	17
2.2.1.1 Control por precompensación .....	17
2.2.1.2 Control PD con precompensación .....	18
2.2.2 Control PD+ y control PD con compensación .....	19
2.2.2.1 Control PD+ .....	19
2.2.2.2 Control PD con compensación.....	19
2.2.3 Control Par-Calculado y Control Par-Calculado+ .....	20
2.2.3.1 Control Par-Calculado .....	20
2.2.3.2 Control Par-Calculado+ .....	21
2.3 MSC Adams .....	22
2.4 Modelo de Co-simulación Adams-Matlab .....	24
3. Modelo cinemático y dinámico del robot .....	25
3.1 Modelo del robot manipulador .....	25
3.1.1 Modelo cinemático .....	25
3.1.1.1 Modelo cinemático Directo .....	25
3.1.1.2 Modelo cinemático Inverso .....	27

3.1.2 Modelo Dinámico.....	29
3.1.2.1 Modelo Dinámico Inverso.....	30
3.1.2.2 Modelo Dinámico Directo.....	32
4. Modelo de simulación MSC Adams.....	33
4.1 Diseño del modelo del robot.....	33
4.2 Selección comercial de los motores.....	40
4.3 Modelo planificador.....	50
4.4 Construcción del modelo de co-simulación.....	51
5. Modelo de Co-simulación y diseño del controlador.....	52
5.1 Diseño del controlador PD.....	53
5.2 control PD con compensación en gravedad.....	58
5.3 Evaluación de la robustez del sistema.....	62
5.4 Validación del controlador para diferentes posiciones deseadas en el espacio de trabajo.....	65
6. Conclusiones y trabajos futuros.....	75
7. Bibliografía.....	76

## Índice de figuras

Figura 1 Esquema control de posición .....	14
Figura 2 Control P con retroalimentación de velocidad y control PD .....	15
Figura 3 Control PD con compensación de gravedad.....	15
Figura 4 Control PD con compensación precalculada de gravedad .....	16
Figura 5 Control PID.....	17
Figura 6 Control de movimiento .....	17
Figura 7 Control por precompensación .....	18
Figura 8 Control PD con precompensación .....	18
Figura 9 Control PD+.....	19
Figura 10 Control PD con compensación .....	20
Figura 11 Control Par-Calculado.....	21
Figura 12 Control Par-Calculado+ .....	22
Figura 13 MSC Adams/View .....	23
Figura 14 Main toolbox.....	23
Figura 15 Plugin Manager.....	24
Figura 16 Adams/control .....	24
Figura 17 Cinemática Directa .....	25
Figura 18 Manipulador de dos grados de libertad .....	26
Figura 19 Cinemática Inversa .....	27
Figura 20 Modelo Dinámico Inverso .....	30
Figura 21 Modelo dinámico directo .....	32
Figura 22 Modelo del manipulador .....	33
Figura 23 Units settings .....	33
Figura 24 Base del manipulador .....	34
Figura 25 Base fija del manipulador .....	34
Figura 26 Configuración eslabón .....	35
Figura 27 Primer Eslabón .....	35
Figura 28 Unión base primer eslabón .....	36
Figura 29 Unión primer y segundo eslabón .....	36
Figura 30 Manipulador con Motion .....	37
Figura 31 Ingreso de variables.....	38
Figura 32 Variable de entrada .....	38
Figura 33 Navegador .....	38
Figura 34 Define la variable de entrada .....	39
Figura 35 Variable de salida .....	39
Figura 36 Posición Inicial del manipulador .....	40

Figura 37 Manipulador a 45° .....	41
Figura 38 MSC Adams/View Posición angular 45° .....	41
Figura 39 Torque generado posición 45° .....	41
Figura 40 Manipulador a 90° .....	42
Figura 41 MSC Adams/View Posición 90° .....	42
Figura 42 Torque generado Posición 90° .....	42
Figura 43 Manipulador a 135° .....	43
Figura 44 MSC Adams/View Posición 135° .....	43
Figura 45 Torque generado Posición 135° .....	43
Figura 46 Manipulador a 180° .....	44
Figura 47 MSC Adams/View Posición 180° .....	44
Figura 48 Torque generado Posición 180° .....	44
Figura 49 Manipulador a 45° con inclinación .....	45
Figura 50 MSC Adams/View 45° .....	45
Figura 51 Torque generado 45° .....	45
Figura 52 Manipulador a 90° con inclinación .....	46
Figura 53 MSC Adams/View 90° .....	46
Figura 54 Torque generado 90° .....	46
Figura 55 Manipulador a 135° con inclinación .....	47
Figura 56 MSC Adams/View 135° .....	47
Figura 57 Torque generado 135° .....	47
Figura 58 Manipulador a 180° con inclinación .....	48
Figura 59 MSC Adams/View 180° .....	48
Figura 60 Torque generado 180° .....	48
Figura 61 Diagrama Bloques motor DC .....	49
Figura 62 Planificador .....	50
Figura 63 Trayectoria planificador .....	50
Figura 64 Adams_sys .....	51
Figura 65 Proceso Adams/Matlab .....	52
Figura 66 Controlador PD .....	53
Figura 67 Control PD .....	54
Figura 68 Trayectoria del manipulador con control PD .....	55
Figura 69 MSC Adams/PostProcessor MD .....	55
Figura 70 Importar resultados .....	56
Figura 71 Par articulación 1 .....	56
Figura 72 Par articulación 2 .....	56
Figura 73 Posición Actual vs deseada control PD .....	57
Figura 74 Control PD con compensación en gravedad .....	58

Figura 75 Diseño matlab del sistema .....	59
Figura 76 Recorrido del efector final .....	60
Figura 77 posición actual vs posición deseada .....	60
Figura 78 Par primera articulación .....	61
Figura 79 Par segunda articulación .....	61
Figura 80 Error posición angular .....	62
Figura 81 Perturbación del sistema .....	62
Figura 82 Trayectoria con perturbación .....	63
Figura 83 Perturbación .....	63
Figura 84 Posiciones angulares .....	64
Figura 85 Salida del control .....	64
Figura 86 Par articulación 1 con perturbación .....	65
Figura 87 Par articulación 2 con perturbación .....	65
Figura 88 Manipulador a 90° .....	66
Figura 89 Posición angular deseada vs actual de 90 .....	67
Figura 90 Señal del control PD con compensación de gravedad .....	67
Figura 91 Par generado .....	68
Figura 92 Manipulador a 135° .....	68
Figura 93 Posición angular actual vs deseada a 135° .....	69
Figura 94 Señal del control para 135° .....	69
Figura 95 Par generado a 135° .....	70
Figura 96 Manipulador a 45° no recto .....	70
Figura 97 Posición angular actual vs deseada a 45° brazo doblado .....	71
Figura 98 Señal de control a 45° brazo doblado .....	71
Figura 99 Par generado a 45° brazo doblado .....	72
Figura 100 Manipulador a 90° no recto .....	72
Figura 101 Posición angular actual vs deseada a 90° brazo doblado .....	73
Figura 102 Señal de control a 90° brazo doblado .....	73
Figura 103 Par generado a 90° brazo doblado .....	74



## Índice de tablas

Tabla 1 Especificaciones brazos del manipulador [1].....	40
Tabla 2 Par de motores.....	49
Tabla 3 Datos de los motores seleccionados para cada articulación [8][9] .....	49
Tabla 4 Posición de cada color [1] .....	66

## RESUMEN GENERAL DE TRABAJO DE GRADO

**TITULO:** MODELADO, SIMULACION Y CONTROL DE UN MANIPULADOR ROBOTICO DE 2 GRADOS DE LIBERTAD EMPLEANDO ADAMS/MATLAB

**AUTORES:** CARLOS IVÁN MENDOZA POVEDA

**FACULTAD:** INGENIERÍA ELECTRÓNICA

**DIRECTOR:** LUIS ANGEL SILVA

### RESUMEN

En este proyecto se desarrolló el modelamiento de un brazo robótico con dos grados de libertad. Se definen sus principales características cinemáticas y dinámicas, llegando a un modelo matemático el cual describe el comportamiento del manipulador en el espacio, a través de los parámetros de Denavit-Hastenberg y las ecuaciones lagrangianas. Se construye el modelo virtualmente por medio del software de simulación MSC Adams reemplazando el modelo dinámico directo. Se establecerá un controlador PD con compensación de gravedad desde el entorno matlab para mover la estructura del robot virtual. Para el análisis y simulaciones utiliza la herramienta Adams/Control la cual establece una co-simulación entre Matlab y MSC Adams, permitiendo analizar el comportamiento de la evolución de la posición espacial y articular, los pares generados por los motores de cada articulación y el los errores del sistema.

**PALABRAS CLAVE:** Co-simulación, Matlab, MSC Adams, modelo matemático, posición espacial, posición articular, pares.

**V° B° DIRECTOR DE TRABAJO DE GRADO**

## ABSTRACT OF THESIS PROJECT

**TITLE:** MODELING, SIMULATION AND CONTROL OF A ROBOT MANIPULATOR OF 2 FREEDOM DEGREES USING ADAMS / MATLAB

**AUTHORS:** CARLOS IVÁN MENDOZA POVEDA

**DEPARTMENT:** ELECTRONICS ENGINEERING

**DIRECTOR:** LUIS ANGEL SILVA

### ABSTRACT

In this project we developed the model of a robotic arm with two degrees of freedom. Its main characteristics are defined kinematics and dynamic, leading to a mathematical model which describes the behavior of the manipulator in space, through the parameters of Denavit-Hastenberg and Lagrangian equations. Virtual model is built using the simulation software MSC Adams replacing the direct dynamic model. There shall be a PD controller with gravity compensation from the MATLAB environment to move the structure of the virtual robot. For analysis and simulation tool uses Adams / Control which provides a co-simulation between Matla and Adams, allowing to analyze the behavior of the evolution of the spatial position and articulation, the pairs generated by the motors at each joint and errors the system.

**KEYWORDS** Co-simulation, Matlab, MSC Adams, mathematical model, spatial position, joint position, pairs

V° B° THESIS DIRECTOR

## Introducción

En el desarrollo industrial siempre se espera la mejora de sus procesos eficazmente con rangos de errores muy pequeños generando así diferentes optimizaciones en su maquinaria y sus lazos de control. Existen diferentes tipos de procesos en la manufactura, donde la precisión, velocidad, economía entre otras es muy importante, por lo cual se tiene en cuenta factores como tipos de control, complejidad del robot; ya que no se justifica implementar un manipulador donde puede desarrollar diferentes tipos de procesos y solo va a hacer uno como manejar un peso máximo de un material a mover y este sea capaz de mover 10 veces ese peso generando un costo mayor.

Por tal motivo es necesario un estudio del manipulador para obtener sus características de movimiento, torque y velocidad, por lo cual se diseñan modelos de estos donde se obtienen diferentes características tales como sus modelos dinámicos y cinemáticos, para el estudio de sus condiciones de trabajo.

En este documento se desarrolló el estudio de un robot manipulador de dos grados de libertad con el cual se obtendrán sus respectivos modelos que nos ayudara a encontrar diferentes condiciones o características de estos como son tipo de motores con sus pares necesarios para lograr un movimiento del manipulador, medidas y tipo de material por medio de simulaciones en el software MSC Adams, también así implementando un controlador PD con compensación en gravedad desde el software Matlab.

Teniendo ya el diseño del manipulador también es posible desarrollar diferentes tipos de comportamientos implementando variaciones en su tipo de control, debido a que en este caso se trabajó con un manipulador no lineal las técnicas de control cuentan con posiciones, velocidades y aceleraciones angulares, teniendo en cuenta esto para una mayor precisión y suavidad del movimiento puede convertirse algo complejo en cuanto a las constantes del controlador.

A continuación se muestra en el capítulo 2 algunos de los controles que se pueden implementar, además se obtendrá los modelos dinámicos y cinemáticos del manipulador en el capítulo 3 y finalmente se diseña la estructura del manipulador y mostrando el proceso de simulación para un brazo robótico con dos articulaciones donde se obtienen múltiples resultados facilitando la construcción virtual de la estructura y lazo de control en los dos últimos capítulos.

# 1. OBJETIVOS

## 1.1 OBJETIVO GENERAL

- Generar la simulación y control del modelo de un robot manipulador de dos grados de libertad con una trayectoria definida por medio de la interacción entre los software Generar la simulación y control de un robot manipulador de dos grados de libertad a través del proceso de simulación entre MSC ADAMS y MATLAB.

## 1.2 OBJETIVOS ESPECÍFICOS

- Desarrollar el modelo de un robot de dos grados de libertad en el software de simulación Adams.
- Encontrar las ecuaciones que definen los modelos cinemáticos y dinámicos del robot de dos grados de libertad.
- Implementar un controlador PD con compensación de gravedad en MATLAB.
- Generar el proceso de co-simulación entre MSC ADAMS y MATLAB para observar el comportamiento dinámico del manipulador en la ejecución de diferentes trayectorias.

## 2. Estado del arte

Para disminuir el margen de error del movimiento es importante la implementación de un controlador, pero en algunos casos se encuentran sistemas que no son lineales, por lo tanto la configuración de este cambian con cada posición del sistema. Una opción para optimizar el controlador es trabajar con la posición, velocidad y aceleración angulares de la estructura a controlar y/o robot manipulador.

Hay varios modelos de controladores, algunos de ellos son:

### 2.1 Control de posición

El control de posición es usado en los robots manipuladores cuando en su trayectoria, interactúan con el medio ambiente. En un controlador de posición se usa posiciones y velocidades angulares deseadas y actuales del planificador, como se muestra en la figura 1, el objetivo de este controlador es determinar  $\tau$  tal que:

$$\lim_{t \rightarrow \infty} q(t) = q_d$$

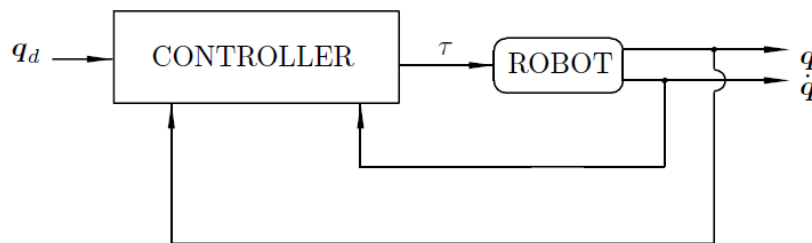


Figura 1 Esquema control de posición [5]

#### 2.1.1 Control P con retroalimentación de velocidad y control PD

Este es el controlador más sencillo que se puede emplear, la ecuación del controlador es

$$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}}$$

Siendo  $K_p$  y  $K_v$  matrices simétricas positivas. El vector  $\tilde{q}$  está dado por  $\tilde{q} = q_d - q$  error de posición, y  $\dot{\tilde{q}}$  la derivada del error de posición como el error de velocidad. El lazo de control de este controlador corresponde a la siguiente figura

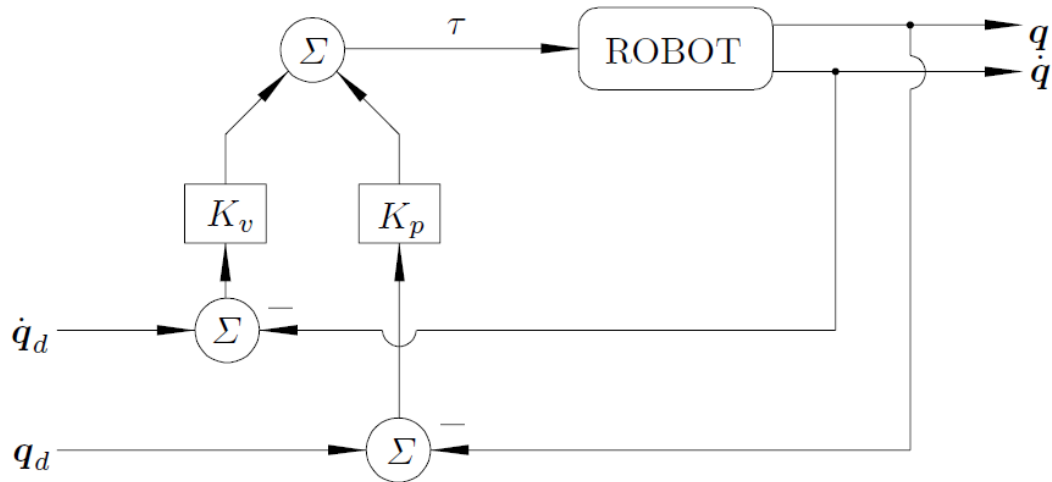


Figura 2 Control P con retroalimentación de velocidad y control PD[5]

### 2.1.2 Control PD con compensación de gravedad

En un controlador PD no se consigue eliminar el error en régimen permanente sin la acción integral ( $K_i$ ), el control PD con compensación de gravedad utilizando el vector de gravedad  $g(q)$  compensa el  $\tau$  en tiempo real con la posición actual ( $q$ ). Para este tipo de controlador es necesario el conocimiento parcial del modelo del manipulador, es decir  $g(q)$ , la ley de control PD con compensación en gravedad viene dada por:

$$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} + g(q)$$

Por lo tanto, una compensación calculada a partir de los términos de gravedad, permitirá utilizar un controlador PD asegurando que el error permanente es nulo.

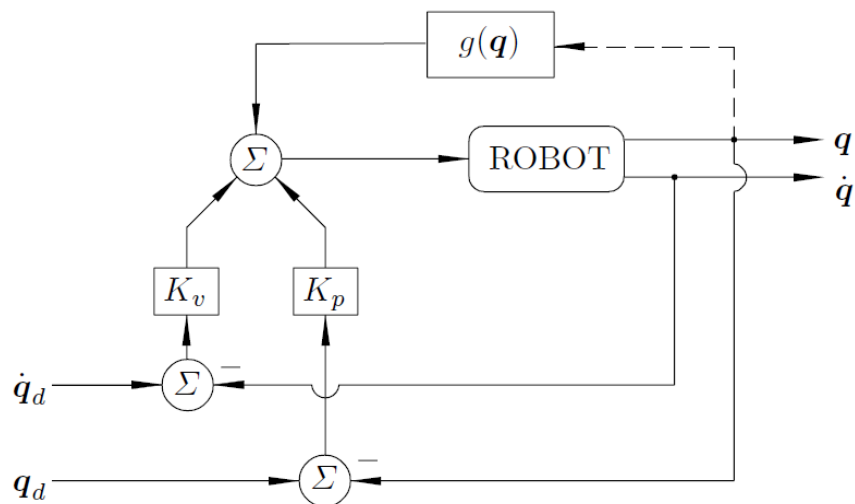


Figura 3 Control PD con compensación de gravedad[5]

### 2.1.3 Control PD con compensación precalculada de gravedad

La única diferencia con el Anterior controlador radica en que el termino  $g(q_d)$  reemplaza a  $g(q)$ , la ley de control PD con compensación precalculada de gravedad viene dada por:

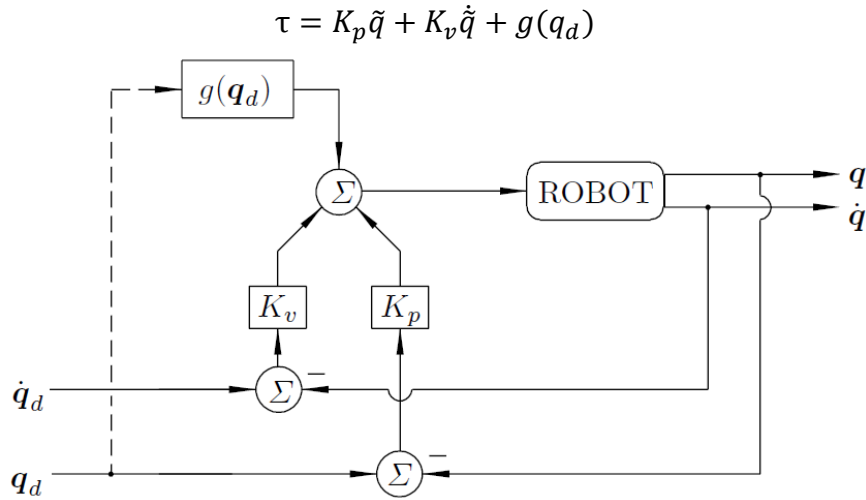


Figura 4 Control PD con compensación precalculada de gravedad[5]

### 2.1.4 Control PID

Al implementar uno de los controladores anteriores no va a satisfacer el objetivo de control de que el error en posición debido a que va a tender a un vector constante pero siempre diferente a  $\tilde{q} = 0$ . La alternativa es la introducción del componente integral para llevar a cero el error de posición. La ley de control PID viene dada por:

$$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} + K_i \int_0^t \tilde{q}(\sigma) d\sigma$$

Donde  $K_p, K_v, K_i$  son matrices de diseño simétricas y definidas positivas.



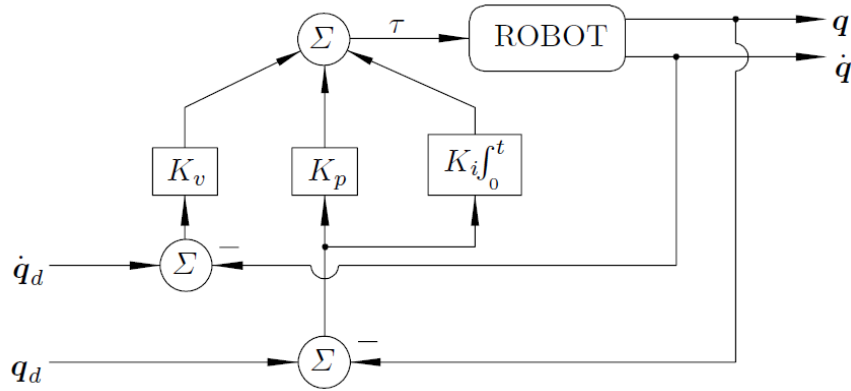


Figura 5 Control PID[5]

## 2.2 Control de movimiento

Uno de los temas importantes del control de robots manipuladores es el control de movimiento o control de trayectoria. Este tipo de controlador es usado en un espacio de trabajo libre siguiendo una trayectoria de varios puntos sin interactuar con el medio ambiente.

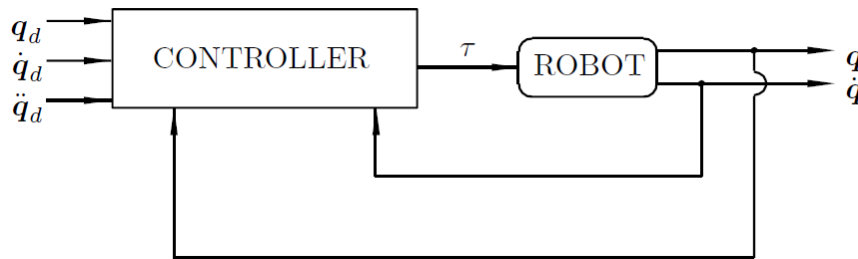


Figura 6 Control de movimiento [5]

### 2.2.1 Control por precompensación y control PD con precompensación

#### 2.2.1.1 Control por precompensación

En las estrategias de control más elementales se encuentra el control en malla abierta donde el controlador es el modelo dinámico inverso, por lo tanto es incapaz de satisfacer el objetivo de control de movimiento. La ley de control viene dada por:

$$\tau = M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + g(q_d)$$

Esta no depende de  $q$  ni de  $\dot{q}$ , este tipo de controlador solo se basa en el modelo del robot

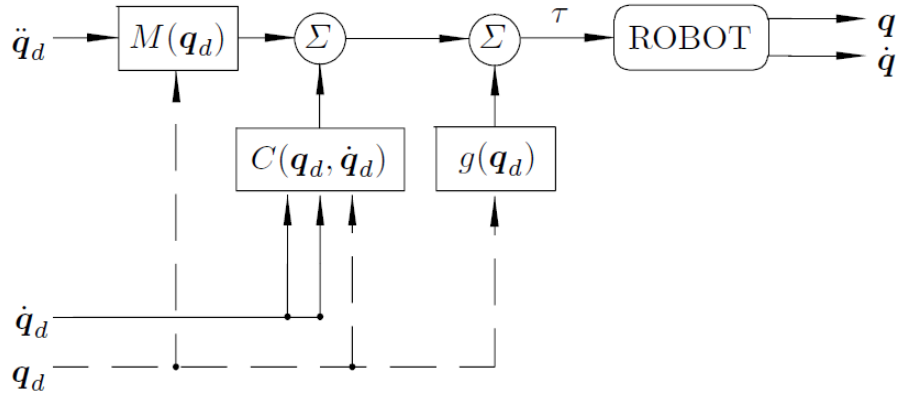


Figura 7 Control por precompensación [5]

### 2.2.1.2 Control PD con precompensación

Este tipo de controlador tiene los fundamentos del PD, puesto que se basa en el modelo de energía, fue introducido debido a la necesidad de un controlador que siga las trayectorias variantes en el tiempo. La ley de control viene dada por:

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

La siguiente figura muestra el bloque correspondiente al control PD con precompensación de robots.

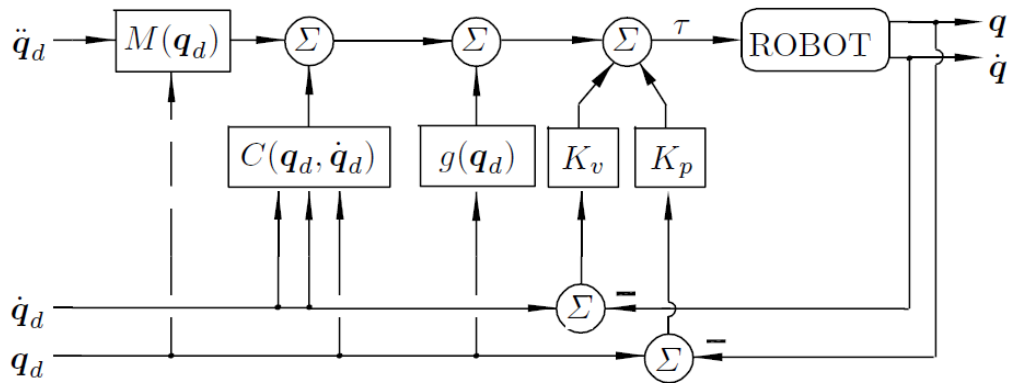


Figura 8 Control PD con precompensación[5]

El beneficio de este controlador sobre el control de movimiento para robots ha mostrado un excelente desempeño, comparado con el empleo del controlador del tipo par calculado. Para obtener excelentes resultados es necesaria una adecuada selección de la matriz de ganancia proporcional  $K_p$ .

### 2.2.2 Control PD+ y control PD con compensación

En el desarrollo de control de movimiento para robots manipuladores como los nombrados anteriormente, el proceso de sintonía de ganancia es laborioso. Los tipos de controladores que se van a presentar donde se garantice el cumplimiento del objetivo de control de movimiento y la sintonía sea triviales.

#### 2.2.2.1 Control PD+

Para este controlador es necesario el conocimiento exacto del modelo, también las trayectorias angulares deseadas. La ley de control viene dada por:

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

La siguiente figura muestra el bloque correspondiente al control PD+

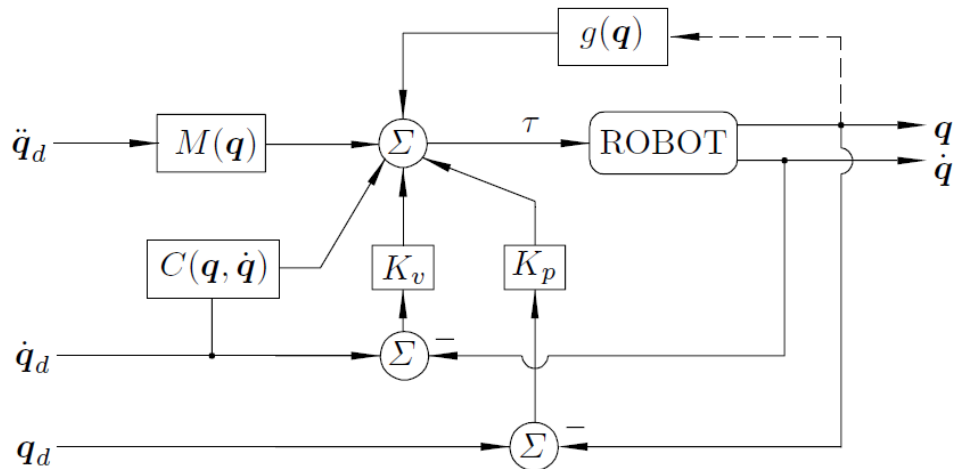


Figura 9 Control PD+[5]

#### 2.2.2.2 Control PD con compensación

Este tipo de control es una generalización del control PD+, siendo este capaz de garantizar el objetivo de control de sus movimientos. La ley de control viene dada por:

$$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} + M(q)[\ddot{q}_d + A\ddot{\tilde{q}}] + C(q, \dot{q})[\dot{q}_d + A\dot{\tilde{q}}] + g(q)$$

La siguiente figura muestra el bloque correspondiente al control PD con compensación.

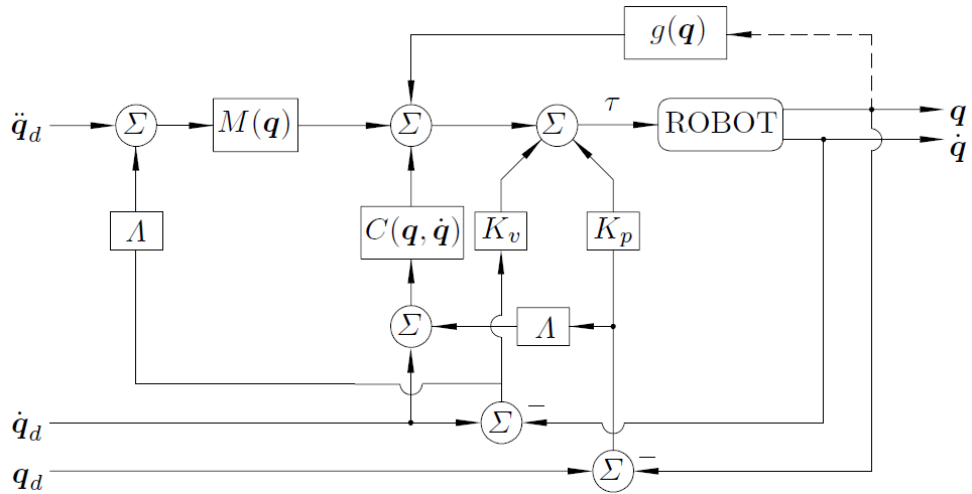


Figura 10 Control PD con compensación [5]

### 2.2.3 Control Par-Calculado y Control Par-Calculado+

En los controladores anteriores tenían en común el control PD. Estos dos controladores presentados no presentan este término lineal PD permitiendo obtener una ecuación de malla cerrada lineal.

#### 2.2.3.1 Control Par-Calculado

El control Par-Calculado hace uso explícito del conocimiento de las matrices  $M(q)$ ,  $C(q, \dot{q})$  y del vector  $g(q)$ . La ley de control viene dada por:

$$\tau = M(q)[\ddot{q}_d + K_p \tilde{q} + K_v \dot{\tilde{q}}] + C(q, \dot{q})\dot{q} + g(q)$$

La siguiente figura muestra el bloque correspondiente al control Par-Calculado.

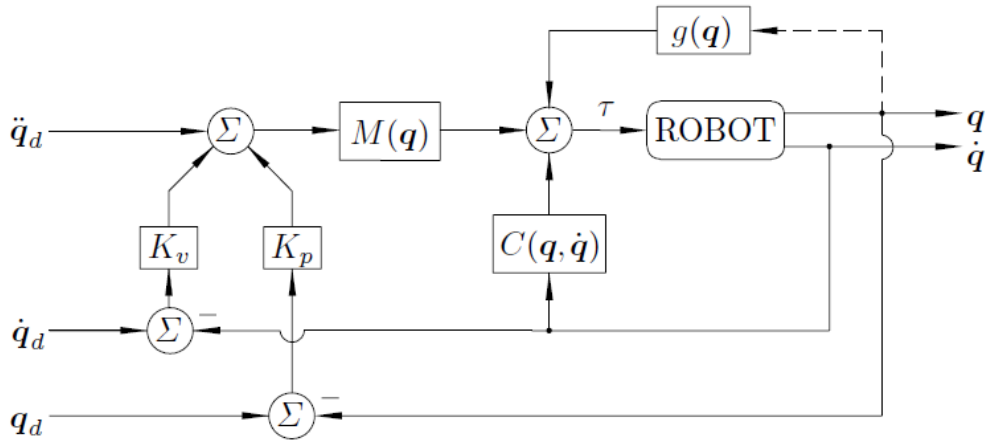


Figura 11 Control Par-Calculado[5]

### 2.2.3.2 Control Par-Calculado+

La ley de control viene dada por:

$$\tau = M(q)[\ddot{q}_d + K_p \tilde{q} + K_v \dot{\tilde{q}}] + C(q, \dot{q})\dot{q} + g(q) - C(q, \dot{q})v$$

El vector  $v$  se obtiene de filtrar los errores de posición  $\tilde{q}$  y de velocidad  $\dot{\tilde{q}}$  de la siguiente manera:

$$v = -\frac{bp}{p + \lambda} \dot{\tilde{q}} - \frac{b}{p + \lambda} [K_v \dot{\tilde{q}} + K_p \tilde{q}]$$

Donde  $p$  es el operador diferencial ( $p = \frac{d}{dt}$ ) y  $\lambda, b$  son constantes positivas de diseño.

La siguiente figura muestra el bloque correspondiente al control Par-Calculado+.



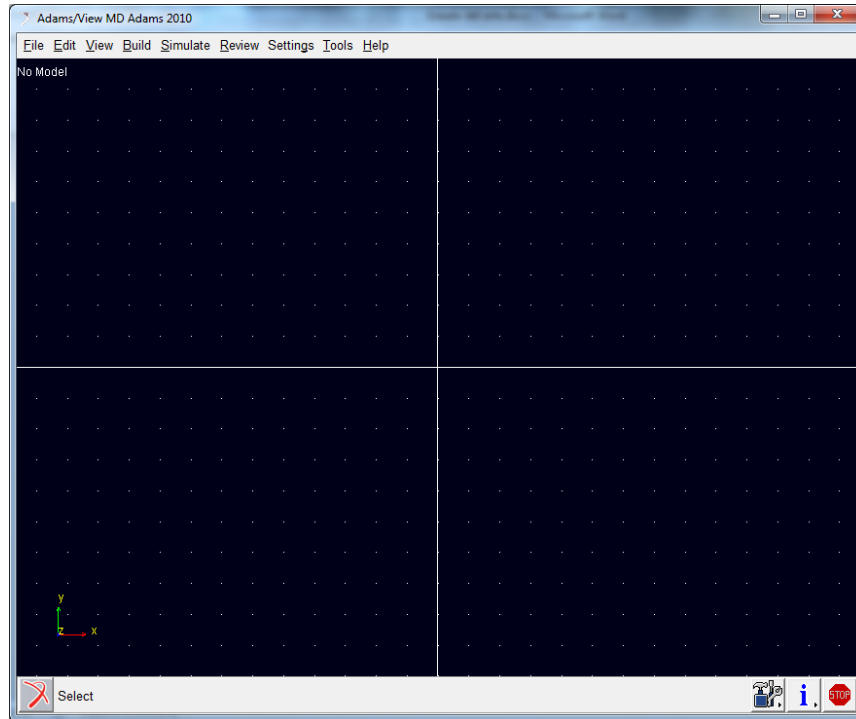


Figura 13 MSC Adams/View[1]

También se tiene la ventana de la barra de herramientas principal (*figura 14*). Aparece por defecto a la izquierda de la pantalla, y en ella se disponen los iconos de las principales herramientas y elementos que se pueden utilizar. Cuenta con múltiples herramientas para el diseño de un modelo virtual.

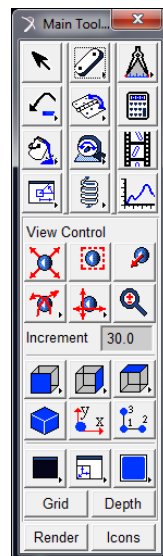


Figura 14 Main toolbox[1]

## 2.4 Modelo de Co-simulación Adams-Matlab

El software MSC Adams cuenta con una librería llamada Adams/Control, la cual nos permite la comunicación con otros programas como Matlab para la simulación conjunta de estos definiendo variables de entradas y salidas. Con esta herramienta es posible ingresar un control para el modelo desarrollado en MSC Adams generando el movimiento por las señales enviadas por el controlador desde Matlab.

Esta librería se activa desde el menú Tools/Plugging manager

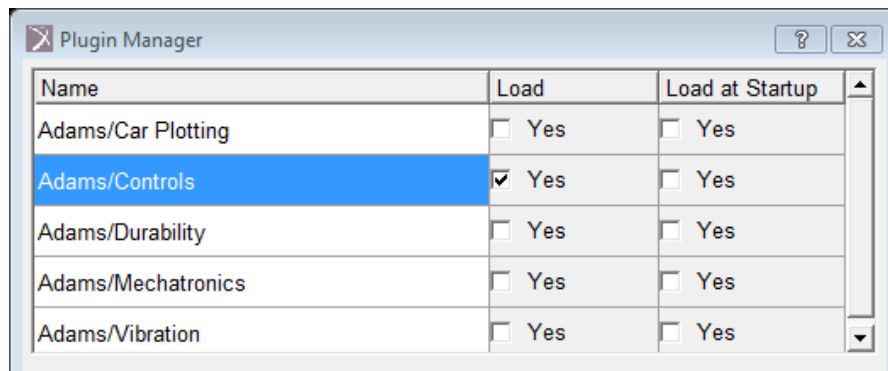


Figura 15 Plugin Manager[1]

En la figura 16 se muestra el entorno de Adams/control para la configuración de las variables del modelo.

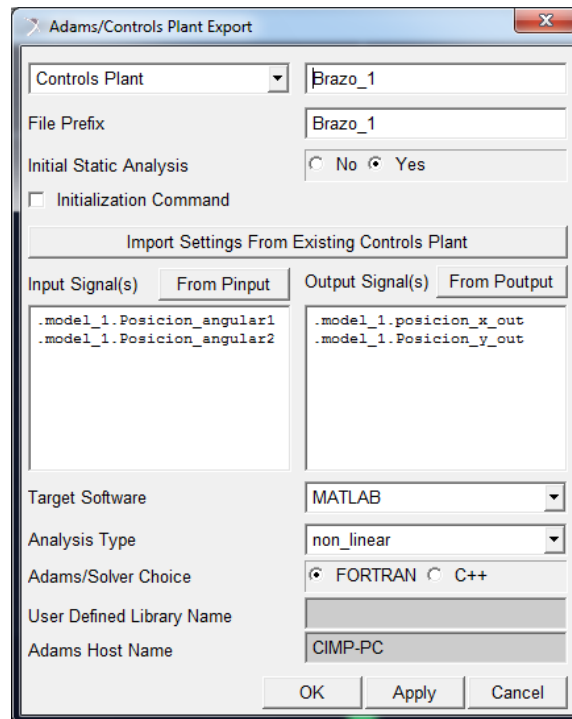


Figura 16 Adams/control [1]



### 3. Modelo cinemático y dinámico del robot

#### 3.1 Modelo del robot manipulador

Se opta para el desarrollo de esta tesis por un robot manipulador de dos grados de libertad no lineal. Para el análisis del robot es necesario encontrar los modelos dinámicos y cinemáticos para el análisis y sintonización del controlador.

##### 3.1.1 Modelo cinemático

En la aplicación de los robots manipuladores se debe entender que se moverá en el espacio, por lo cual se debe representar posiciones y orientación. Para esto se asignan en sistemas coordenados en el plano, con esto se logra obtener la posición inicial del efector y la posición final, junto con la trayectoria del manipulador.

##### 3.1.1.1 Modelo cinemático Directo

La función de este modelo es poder obtener las posiciones espaciales por medio de sus posiciones angulares,



Figura 17 Cinemática Directa [1]

Para el estudio cinemático directo se aplica el algoritmo de Denavit-Hartenberg, que es un método matricial que permite establecer de manera sistemática un sistema de coordenadas ligado a cada eslabón de una cadena articulada, pudiéndose determinar a continuación las ecuaciones cinemáticas de la cadena completa.

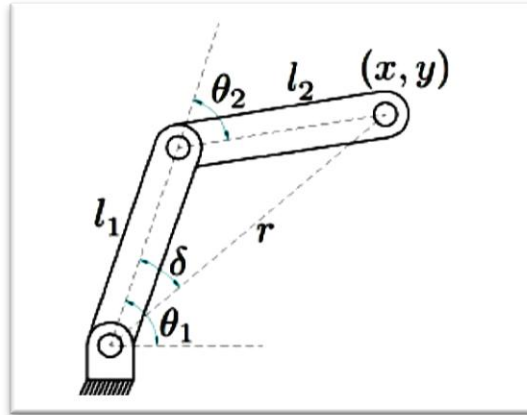
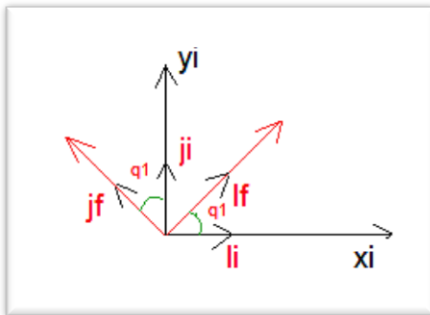


Figura 18 Manipulador de dos grados de libertad [1]

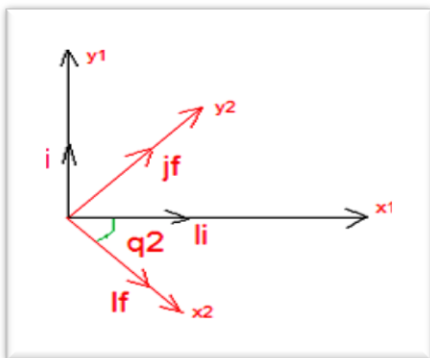
Se obtiene las ecuaciones desde tres puntos del manipulador,  $T_1^R$  equivalente a la base del manipulador,  $T_2^1$  al codo del brazo y  $T_3^2$  al efector final del manipulador.

Base del manipulador



$$T_1^R = \begin{bmatrix} \cos(q_1) & -\text{sen}(q_1) & 0 & 0 \\ \text{sen}(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Codo del manipulador



$$T_2^1 = \begin{bmatrix} \cos(q_2) & \text{sen}(q_2) & 0 & l_1 \\ -\text{sen}(q_2) & \cos(q_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Efector final

$$T_3^2 = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Con el producto de las tres matrices anteriores obtenemos el sistema del brazo

$$T_3^R = T_1^R T_2^1 T_3^2$$

$$T_3^R = \begin{bmatrix} \cos(q_1) & -\text{sen}(q_1) & 0 & 0 \\ \text{sen}(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(q_2) & \text{sen}(q_2) & 0 & l_1 \\ -\text{sen}(q_2) & \cos(q_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^R = \begin{bmatrix} \cos(q_1 - q_2) & \text{sen}(q_1 - q_2) & 0 & l_1 \cos(q_1) + l_2 \cos(q_1 - q_2) \\ \text{sen}(q_1 - q_2) & \cos(q_1 - q_2) & 0 & l_1 \text{sen}(q_1) + l_2 \text{sen}(q_1 - q_2) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$p^R = T_3^R p^3, \text{ siendo } p^3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$p^R = \begin{bmatrix} l_1 \cos(q_1) + l_2 \cos(q_1 - q_2) \\ l_1 \text{sen}(q_1) + l_2 \text{sen}(q_1 - q_2) \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{aligned} x &= l_1 \cos(q_1) + l_2 \cos(q_1 - q_2) \\ y &= l_1 \text{sen}(q_1) + l_2 \text{sen}(q_1 - q_2) \\ z &= 0 \end{aligned}$$

### 3.1.1.2 Modelo cinemático Inverso

Con la cinemática inversa obtenemos por medio de las posiciones cardinales las posiciones angulares,



Figura 19 Cinemática Inversa [1]

Para el movimiento del manipulador se cuenta con una trayectoria en un sistema de coordenados X y Y, pero en la ubicación de los eslabones del manipulador, se trabaja en posiciones angulares o ángulos, para esto es necesario la conversión de la trayectoria a estos. Con el modelo cinemático inverso logramos obtener las posiciones angulares para cada punto en el espacio. Por lo tanto es necesario encontrar un par de ecuaciones que satisfagan la relación de los sistemas coordenados a posiciones angulares  $q_1$  y  $q_2$ .

Como se mencionó anteriormente, las ecuaciones para un sistema coordenado basado en ángulos son,

$$\begin{aligned}x &= l_1 \cos(q_1) + l_2 \cos(q_1 - q_2) \rightarrow x = l_1 C_1 + l_2 C_{12} \\y &= l_1 \sen(q_1) + l_2 \sen(q_1 - q_2) \rightarrow y = l_1 S_1 + l_2 S_{12}\end{aligned}$$

Siendo,

$$\begin{aligned}C_{12} &= C_1 C_2 - S_1 S_2 \\S_{12} &= C_1 S_2 - S_1 C_2\end{aligned}$$

Elevando al cuadrado estos términos obtenemos que,

$$x^2 + y^2 = l_1^2 + l_2^2 + 2l_1 l_2 C_2$$

Despejando  $C_2$ ,

$$C_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2}$$

Para que pueda existir una solución el lado derecho debe tener un valor entre -1 y 1. Suponiendo que el destino está en el espacio de trabajo escribimos una expresión para  $s_2$

$$s_2 = \pm \sqrt{1 - c_2^2}$$

Para calcular  $Q_2$  utilizamos la rutina de arco tangente de dos argumentos

$$Q_2 = \text{Atan2}(s_2, c_2)$$

Habiendo encontrado  $Q_2$  resolvemos las siguientes ecuaciones para encontrar  $Q_1$

$$x = l_1 C_1 + l_2 C_{12} \quad (1)$$

$$y = l_1 S_1 + l_2 S_{12} \quad (2)$$

Usando la ecuación de solución y realizando un cambio de variables tenemos que

$$r = + \sqrt{l_1^2 + l_2^2}$$

Y

$$\gamma = \text{Atan2}(l_2, l_1)$$

Entonces

$$l_1 = r \cos \gamma$$

$$l_2 = r \sen \gamma$$

Reemplazando en la ecuación (1) y (2)

$$\frac{x}{r} = \cos\gamma\cos\theta_1 - \text{sen}\gamma\text{sen}\theta_1$$
$$\frac{y}{r} = \cos\gamma\text{sen}\theta_1 - \text{sen}\gamma\cos\theta_1$$

Por lo tanto

$$\cos(\gamma + \theta_1) = \frac{x}{r}$$
$$\text{sen}(\gamma + \theta_1) = \frac{y}{r}$$

Usando el arreglo de dos argumentos obtenemos

$$\gamma + \theta_1 = \text{Atan2}\left(\frac{y}{r}, \frac{x}{r}\right) = \text{Atan2}(y, x)$$

Despejando  $\theta_1$

$$\theta_1 = \text{Atan2}(x, y) - \text{Atan2}(l_2, l_1)$$

### 3.1.2 Modelo Dinámico

En la simulación de un manipulador es necesario un modelo para ver cómo se mueve cuando se emplea un conjunto de pares de fuerza. Es necesario calcular las ecuaciones de movimiento para cada articulación. El modelo dinámico general de un manipulador de  $n$  grados de libertad está regido por:

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

Siendo,

$\tau$ : Torque ejercido por los actuadores de cada articulación

$q$ : Vector de variables articulares

$M(q)$ : Matriz de inercia

$C(q, \dot{q})$ : Matriz centrífuga y de coriolis

$g(q)$ : Vector de gravedad

$f(\dot{q})$ : Propiedades de la dinámica residual

Para encontrar estos modelos existe:

- Formulación lagrange que se basa en el balance energético
- Formulación de Newton-Euler que se basa en la segunda ley de Newton y la ecuación de inercia de Euler

Este trabajo de grado se laboró en base a la formulación lagrange

### 3.1.2.1 Modelo Dinámico Inverso

Este modelo expresa las fuerzas y pares que intervienen en función de la evolución de las coordenadas articulares y sus derivadas.



Figura 20 Modelo Dinámico Inverso [1]

Expresa las fuerzas y pares que intervienen en función de la evolución de las coordenadas articulares y sus derivadas, para obtener las ecuaciones es necesario obtener la energía cinética y potencial de cada articulación, cuya función está dada por,

- Energía Cinética

$$K_1(q, \dot{q}) = \frac{1}{2} m_1 l_1^2 \dot{q}_1^2$$

$$K_2(q, \dot{q}) = \frac{1}{2} m_2 l_1^2 \dot{q}_1^2 + \frac{1}{2} m_2 l_2^2 [\dot{q}_1^2 - 2\dot{q}_1\dot{q}_2 + \dot{q}_2^2] + m_2 l_1 l_2 [\dot{q}_1^2 - \dot{q}_1\dot{q}_2] \cos(q_2)$$

- Energía Potencial

$$U_1(q) = m_1 g l_1 \text{sen}(q_1)$$

$$U_2(q) = m_2 g l_1 \text{sen}(q_1) + m_2 g l_2 \text{sen}(q_1 - q_2)$$

Aplicando el lagrangiano a ambas articulaciones obtenemos,

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

$$L(q, \dot{q}) = K_1(q, \dot{q}) + K_2(q, \dot{q}) - U_1(q) - U_2(q)$$

$$\begin{aligned} L(q, \dot{q}) = & \frac{1}{2} m_1 l_1^2 \dot{q}_1^2 + \frac{1}{2} m_2 l_1^2 \dot{q}_1^2 + \frac{1}{2} m_2 l_2^2 [\dot{q}_1^2 - 2\dot{q}_1\dot{q}_2 + \dot{q}_2^2] \\ & + m_2 l_1 l_2 [\dot{q}_1^2 - \dot{q}_1\dot{q}_2] \cos(q_2) - m_1 g l_1 \text{sen}(q_1) - m_2 g l_1 \text{sen}(q_1) \\ & - m_2 g l_2 \text{sen}(q_1 - q_2) \end{aligned}$$

Para la primera articulación,

$$\frac{\partial L}{\partial \dot{q}_1} = [m_1 l_1^2 + m_2 l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(q_2)] \dot{q}_1 - [m_2 l_1 l_2 \cos(q_2) + m_2 l_2^2] \dot{q}_2$$

$$\begin{aligned} \frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{q}_1} \right] &= [m_1 l_1^2 + m_2 l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(q_2)] \ddot{q}_1 \\ &\quad - [m_2 l_1 l_2 \cos(q_2) + m_2 l_2^2] \ddot{q}_2 - 2m_2 l_1 l_2 \sin(q_2) \dot{q}_1 \dot{q}_2 \\ &\quad + m_2 l_1 l_2 \sin(q_2) \dot{q}_2^2 \end{aligned}$$

$$\frac{\partial L}{\partial q_1} = -m_1 g l_1 \cos(q_1) - m_2 g l_1 \cos(q_1) - m_2 g l_2 \cos(q_1 - q_2)$$

Segunda articulación,

$$\begin{aligned} \frac{\partial L}{\partial \dot{q}_2} &= -m_2 l_1 l_2 \cos(q_2) \dot{q}_1 - m_2 l_2^2 \dot{q}_1 + m_2 l_2^2 \dot{q}_2 \\ \frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{q}_2} \right] &= [-m_2 l_1 l_2 \cos(q_2) - m_2 l_2^2] \ddot{q}_1 + m_2 l_2^2 \ddot{q}_2 + m_2 l_1 l_2 \sin(q_2) \dot{q}_1 \dot{q}_2 \end{aligned}$$

$$\frac{\partial L}{\partial q_2} = -m_2 l_1 l_2 [\dot{q}_1^2 - \dot{q}_1 \dot{q}_2] \sin(q_2) - m_2 g l_2 \cos(q_1 - q_2)$$

Donde la ecuación de movimiento para el manipulador está dada por,

$$\tau_i = \frac{d}{dx} \left[ \frac{\partial L}{\partial \dot{q}_i} \right] - \frac{\partial L}{\partial q_i}, i = 1, 2, 3 \dots$$

Donde  $\tau$  es un vector  $n \times 1$  de momentos de torsión de los actuadores, sustituyendo en la anterior ecuación obtenemos la dinámica inversa,

$$\begin{aligned} \tau_1 &= [m_1 l_1^2 + m_2 l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(q_2)] \ddot{q}_1 - [m_2 l_1 l_2 \cos(q_2) + m_2 l_2^2] \ddot{q}_2 \\ &\quad + 2m_2 l_1 l_2 \sin(q_2) \dot{q}_1 \dot{q}_2 - m_2 l_1 l_2 \sin(q_2) \dot{q}_2^2 + [m_1 l_1 \\ &\quad + m_2 l_1] g \cos(q_1) + m_2 g l_2 \cos(q_1 - q_2) \end{aligned}$$

$$\tau_2 = [-m_2 l_1 l_2 \cos(q_2) - m_2 l_2^2] \ddot{q}_1 + m_2 l_2^2 \ddot{q}_2 + m_2 l_1 l_2 \sin(q_2) \dot{q}_1^2 + m_2 g l_2 \cos(q_1 - q_2)$$

De las dos ecuaciones anteriores podemos obtener las siguientes matrices,

Matriz de inercia

$$M(q) = \begin{bmatrix} [m_1 l_1^2 + m_2 l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(q_2)] & [-m_2 l_1 l_2 \cos(q_2) - m_2 l_2^2] \\ [-m_2 l_1 l_2 \cos(q_2) - m_2 l_2^2] & m_2 l_2^2 \end{bmatrix}$$

Matriz centrífuga y de coriolis

$$C(q, \dot{q}) = \begin{bmatrix} -2m_2 l_1 l_2 \sin(q_2) \dot{q}_2 & m_2 l_1 l_2 \sin(q_2) (2\dot{q}_1 - \dot{q}_2) \\ m_2 l_1 l_2 \sin(q_2) \dot{q}_1 & 0 \end{bmatrix}$$

Vector de gravedad

$$g(q) = \begin{bmatrix} [m_1 l_1 + m_2 l_1] g \cos(q_1) + m_2 g l_2 \cos(q_1 - q_2) \\ m_2 g l_2 \cos(q_1 - q_2) \end{bmatrix}$$

### 3.1.2.2 Modelo Dinámico Directo

Este modelo expresa la evolución temporal de las coordenadas articulares del robot en función de las fuerzas y pares que intervienen,



Figura 21 Modelo dinámico directo [1]

Para obtener el modelo dinámico directo se emplea las ecuaciones obtenidas en el modelo dinámico inverso y empleando la regla de Cramer,

$$\begin{aligned} \tau_1 &= A\ddot{\theta}_1 + J + v_1\dot{\theta}_1 \\ \tau_2 &= E\ddot{\theta}_2 + F\dot{\theta}_3 + K + N + v_2\dot{\theta}_2 \\ \tau_3 &= H\ddot{\theta}_2 + I\dot{\theta}_3 + P + O + v_3\dot{\theta}_3 \end{aligned}$$

Debido a la complejidad matemática de este modelo se opta por la construcción de un modelo virtual del manipulador en el software de simulación MSC Adams donde se construye cada brazo de dos grados de libertad con su respectivo peso, material y longitud.



## 4. Modelo de simulación MSC Adams

### 4.1 Diseño del modelo del robot

Se optó por la construcción de un modelo de dos grados de libertad para la simulación del controlador en el software MSC Adams/View

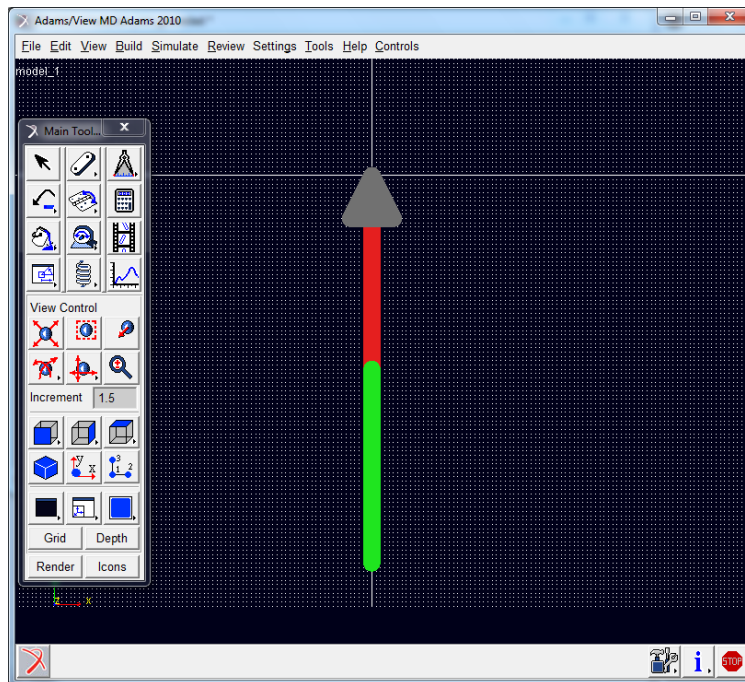


Figura 22 Modelo del manipulador [1]

Para la construcción del modelo es necesario establecer parámetros de medidas en el menú settings/Units, especificando pesos, longitudes entre otros (figura 23),

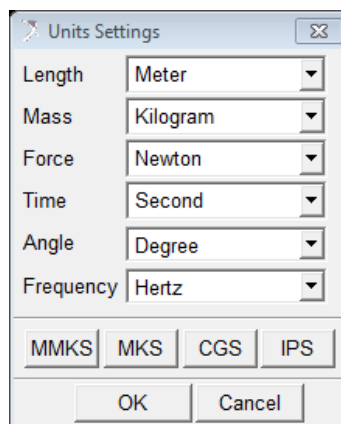



Figura 23 Units settings [1]

Luego se trabaja con el Main Toolbox; Inicialmente se coloca una base fija donde nuestro robot se va a sostener, se selecciona con click derecho Rigid body y se escoge plate , nos pedirá que se coloque una serie de puntos, se coloca el primero en el origen el cual servirá para unir con el primer eslabón (figura 24),

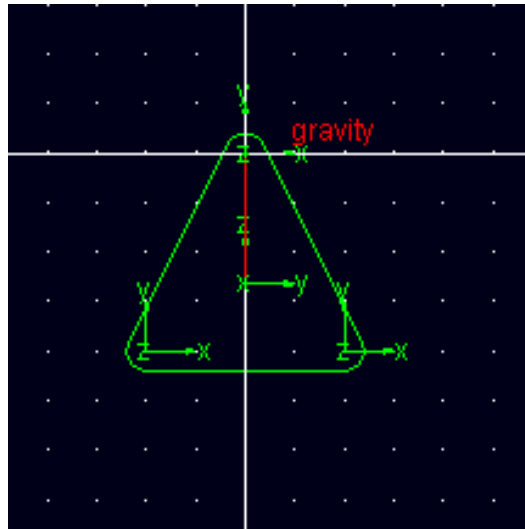



Figura 24 Base del manipulador [1]

Ahora hay que fijarlo para que sea una base fija y no caiga con la gravedad, para esto se selecciona Joint/fixed , en construcción se coloca "1 location" y luego se ubica en el punto del centro de nuestra base

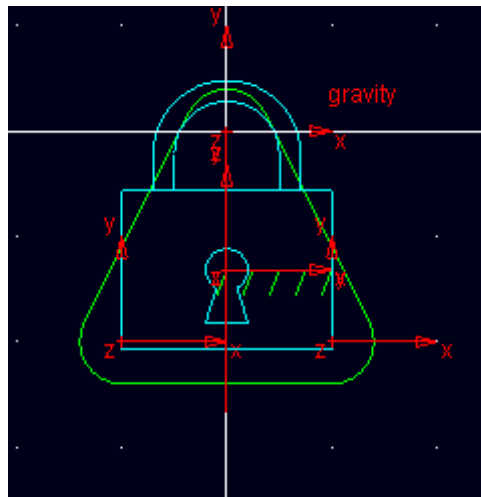



Figura 25 Base fija del manipulador [1]

Ya obtenido nuestra base fija (figura 25), se comienza a armar el robot con Rigid body y link  (figura 26), donde nos pedirá que se especifique ancho, largo y alto de nuestro primer eslabón

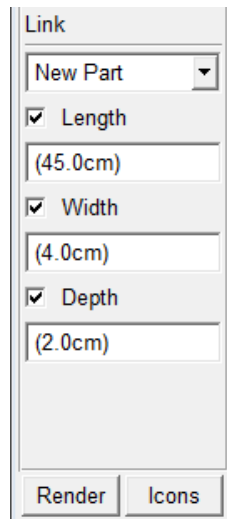


Figura 26 Configuración eslabón [1]

Luego pedirá el punto donde se pondrá el primer eslabón y su orientación, se seleccionara el origen del margen con orientación con la gravedad, como se muestra en la figura 27

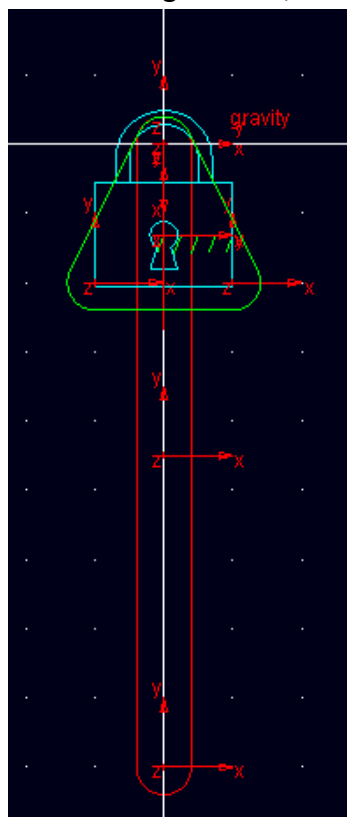



Figura 27 Primer Eslabón [1]

Ahora se une físicamente estos dos elementos por medio de Joint/Revolute ; en construcción seleccionándolo entre dos cuerpos una localización, nuestro primer cuerpo es el plate y el segundo nuestro eslabón con localización en el origen

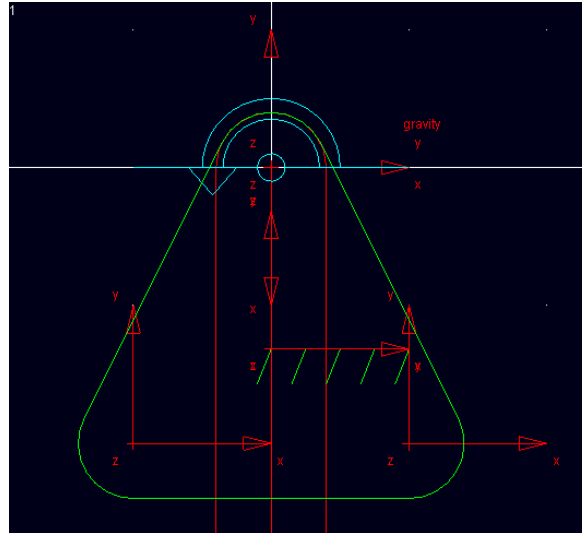


Figura 28 Unión base primer eslabón [1]

Nos ha de aparecer una flecha en sentido contrario a las manecillas del reloj (*Figura 28*), significando que ambas piezas están unidas en ese punto. Para el segundo eslabón repetiremos los dos pasos anteriores pero nuestros puntos de unión es el extremo del primer eslabón (*figura 29*).

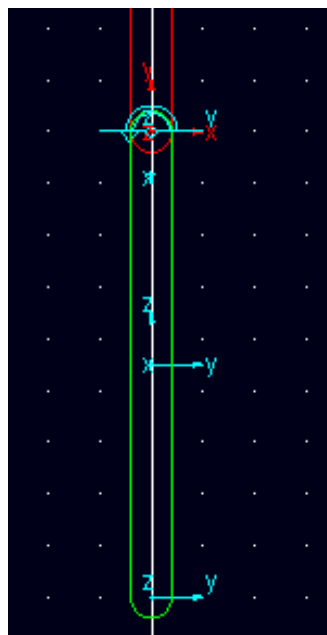



Figura 29 Unión primer y segundo eslabón [1]

Ya construido nuestro modelo de simulación, se procede a definir las variables de entradas y salidas como son torque y posición angular; para esto es necesario aplicar un movimiento en las articulaciones con el icono Motion , y se aplica en cada articulación

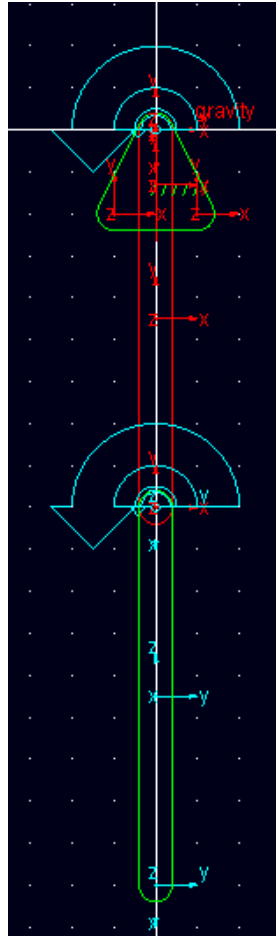


Figura 30 Manipulador con Motion [1]

Aparecerán flechas más grandes en sentido contrario a las manecillas del reloj como se muestra en la figura 30.

- VARIABLES DE ENTRADA

Para definir las variables de entradas que en nuestro caso son los pares de cada articulación, se selecciona en el menú Build /System Elements/State Variable/New (figura 31).

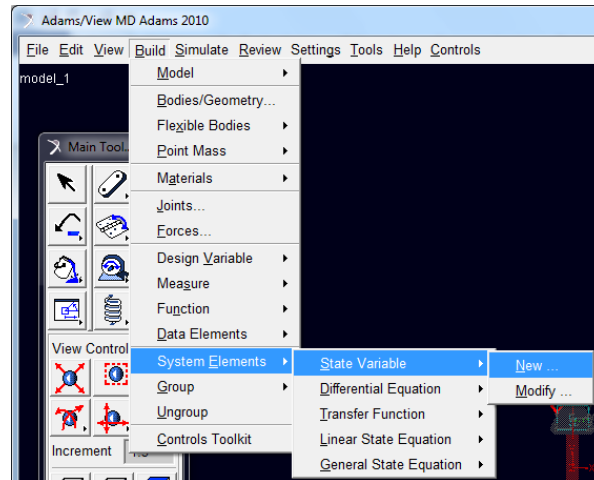


Figura 31 Ingreso de variables [1]

Aparece una nueva ventana donde se ingresa el nombre a nuestra variable de entrada (figura 32).

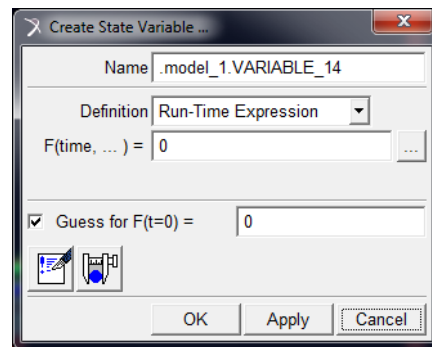


Figura 32 Variable de entrada [1]

En el menú Edit se selecciona Modify, abriendo una nueva ventana donde nos muestra todas las variables del modelo (figura 33).

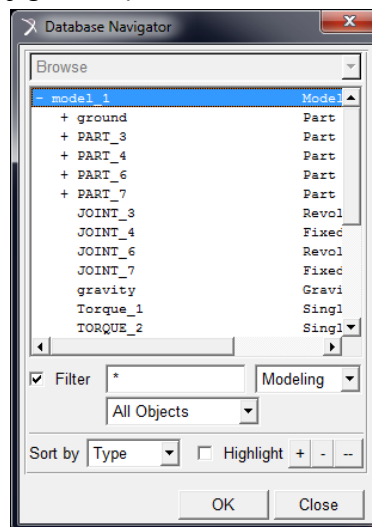


Figura 33 Navegador [1]

Seleccionando la variable donde queremos que ingrese la variable de entrada, en este caso los Motion creados anteriormente y en Function ingresamos VARVAL(Nombre de la variable de entrada(figura 34)).

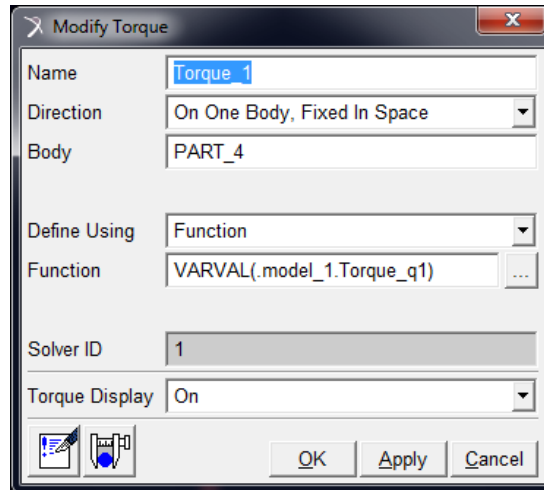


Figura 34 Define la variable de entrada [1]

- Variables de salida

Para obtener señales de salida, se crea igualmente las variables en el menú Build /System Elements/State Variable/New, luego en F(time,...) se ingresa el nombre de la variable de nuestro modelo que se desea como salida, sea un ángulo que se mide, distancia, fuerza, velocidad, etc...

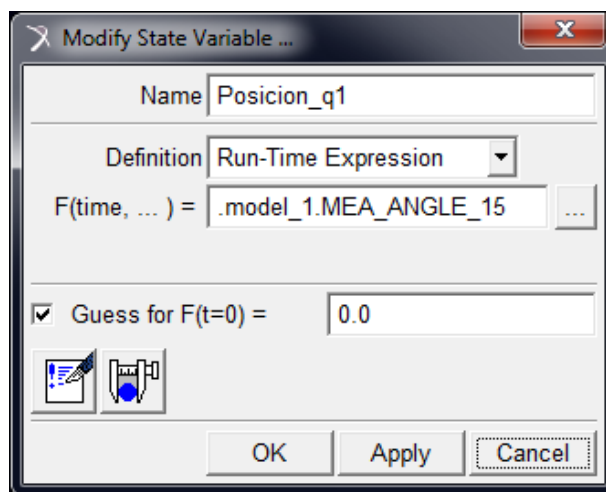


Figura 35 Variable de salida [1]

Con esto se finaliza la construcción del modelo virtual de nuestro manipulador, listo para lograr la conexión entre Matlab y MSC Adams.

Ambos brazos tienen las mismas características de peso, longitud y material.

Eslabón	Peso (Kg)	Largo (cm)	Ancho(cm)	Profundidad(cm)	Material
1	1.0541365252	45	4	2	Aluminio
2	1.0541365252	45	4	2	Aluminio

Tabla 1 Especificaciones brazos del manipulador [1]

#### 4.2 Selección comercial de los motores

Para la selección de los motores, se desarrolla una simulación en Adams View donde se mira el par generado en diferentes posiciones dadas. Esto se logra mediante un Spline la cual crea una función continua de una serie de datos, ingresando las posiciones angulares de cada brazo con su respectivo tiempo de respuesta. Se toman posiciones con el brazo completamente recto creando una circunferencia de  $0^\circ$  a  $180^\circ$ , también se toman posiciones a medio brazo para generar diferentes pares con el brazo recogido. Se tomaran muestras de señales desde MSC Adams mostrando la trayectoria obtenida por el manipulador junto con los pares generados en cada articulación.

Como posición inicial del manipulador como se muestra en la figura 37, se considerara con grado  $0^\circ$ .

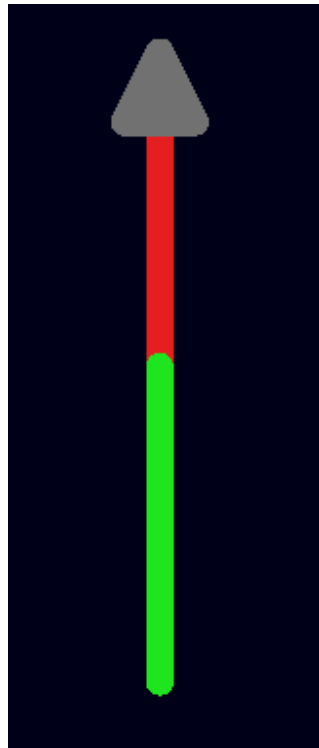


Figura 36 Posición Inicial del manipulador [1]



- Posición a 45° equivalente a X=636, Y= -636

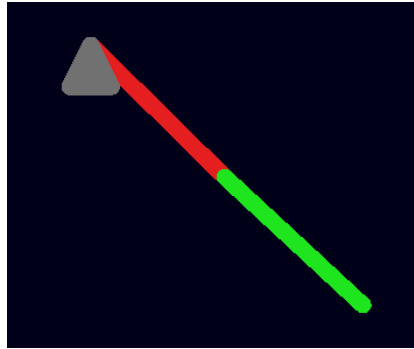


Figura 37 Manipulador a 45° [1]

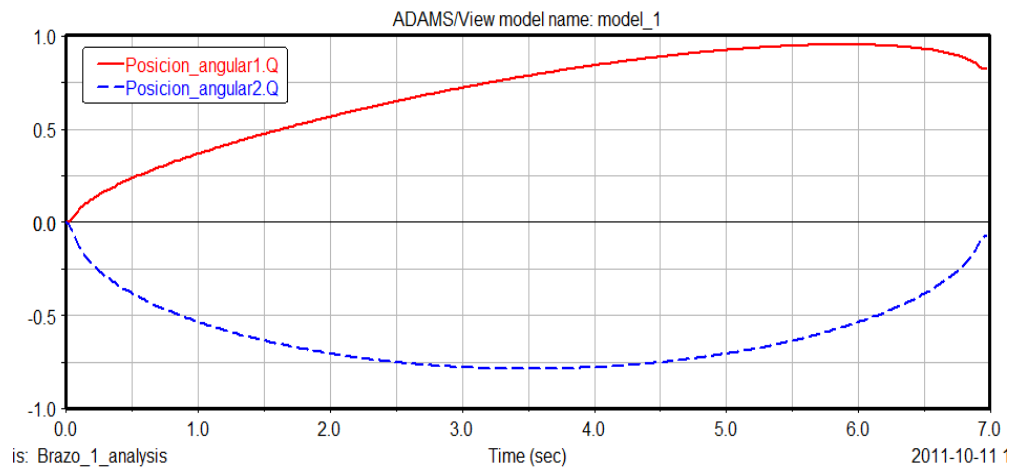


Figura 38 MSC Adams/View Posición angular 45° [1]

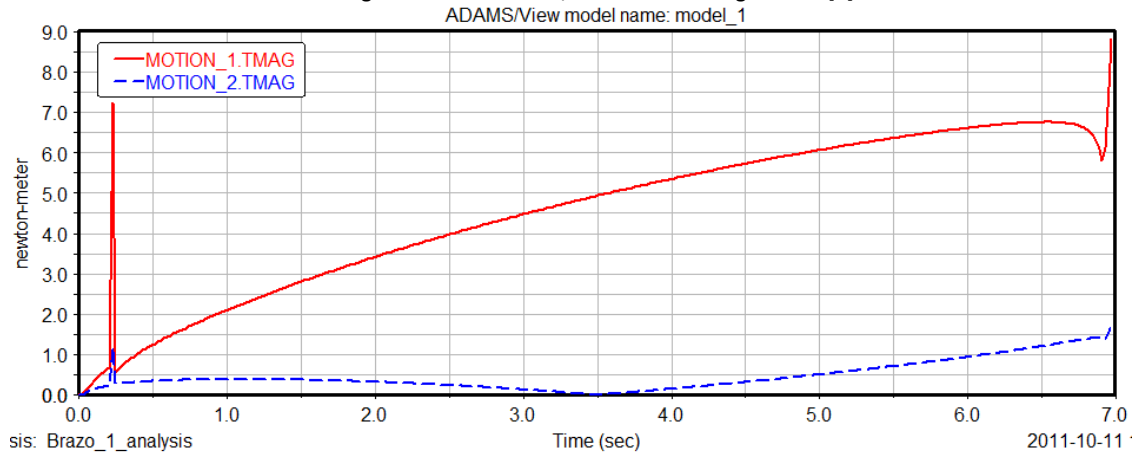


Figura 39 Torque generado posición 45° [1]

La posición angular actual se observa en las figuras 38, confirmando el correcto posicionamiento del manipulador. Se obtiene un par máximo de 10Nm para la primera articulación y 1.7Nm para la segunda articulación con una posición de 45° del manipulador como se muestra en la figura 39

- Posición a 90° equivalente a X=900, Y= 0

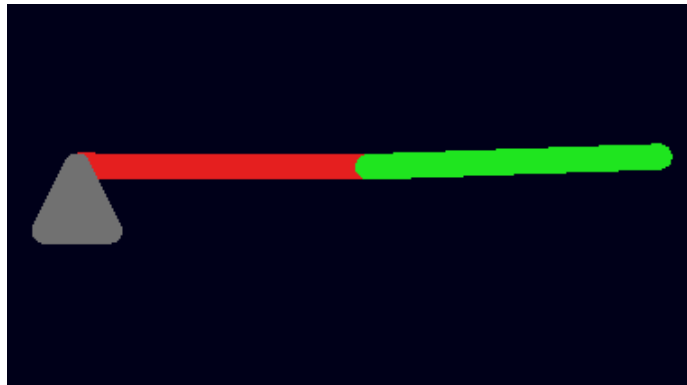


Figura 40 Manipulador a 90° [1]

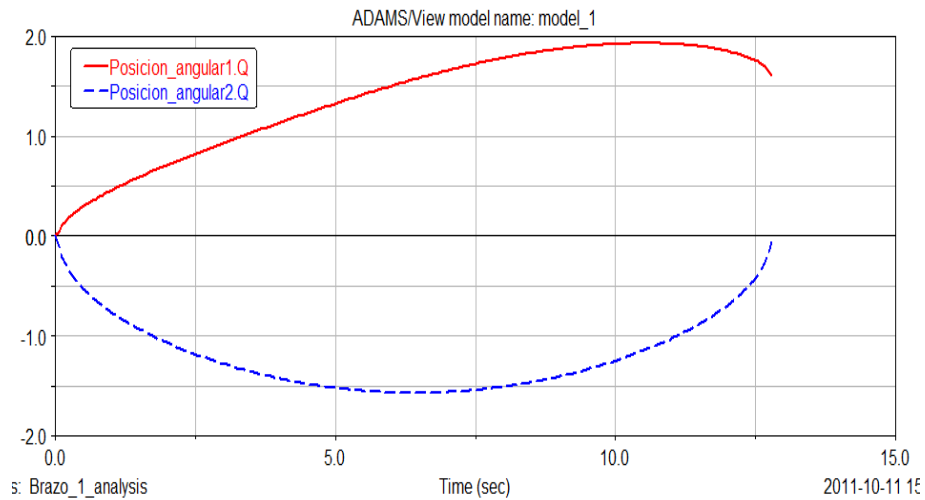


Figura 41 MSC Adams/View Posición 90° [1]

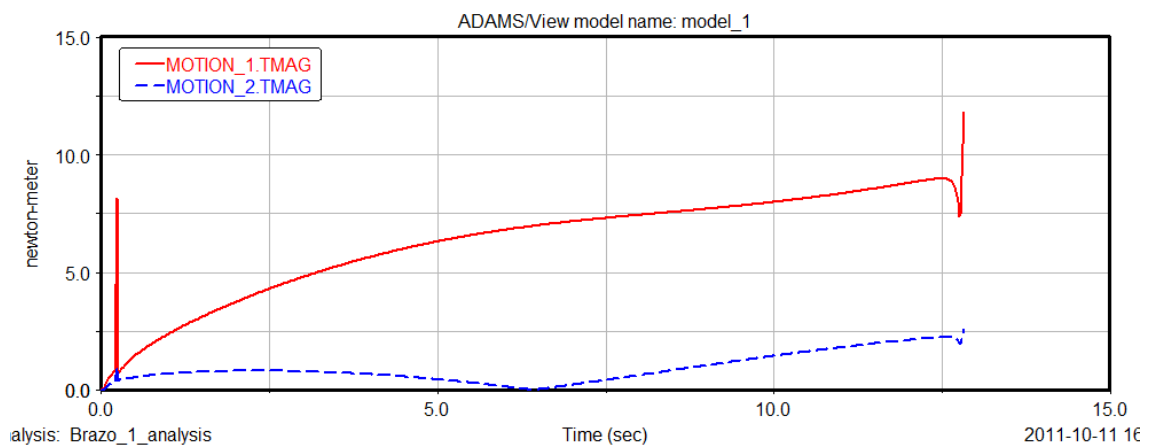


Figura 42 Torque generado Posición 90° [1]

Con una posición de 90° se obtiene un par para la primera articulación de 12Nm y para la segunda articulación un par de 2.5Nm.

- Posición a 135° equivalente a X=636, Y= 636

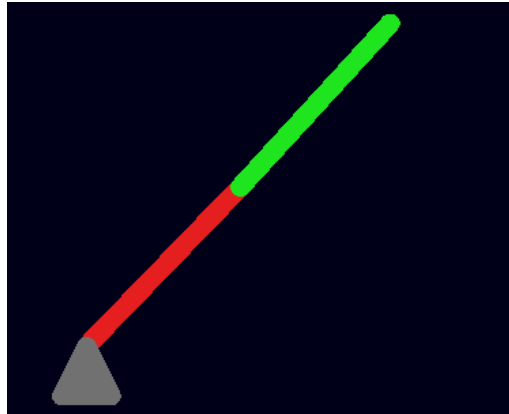


Figura 43 Manipulador a 135° [1]

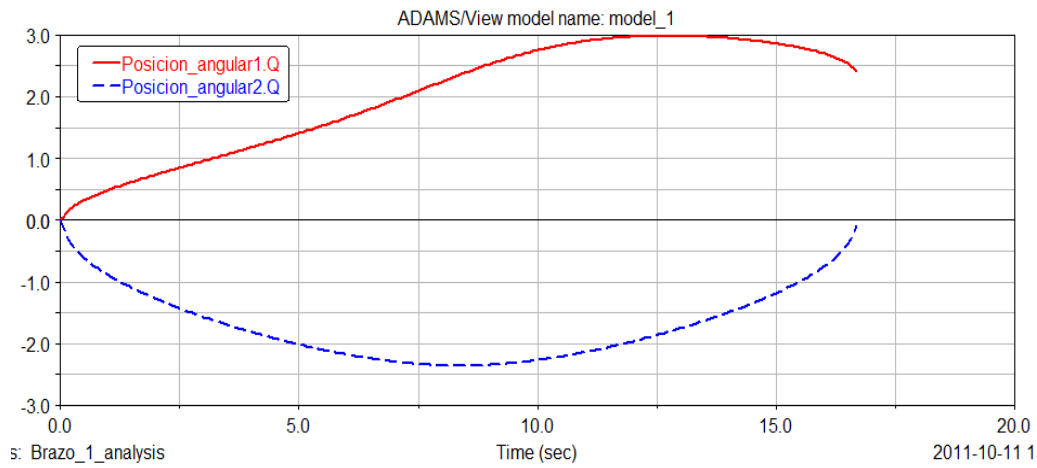


Figura 44 MSC Adams/View Posición 135° [1]

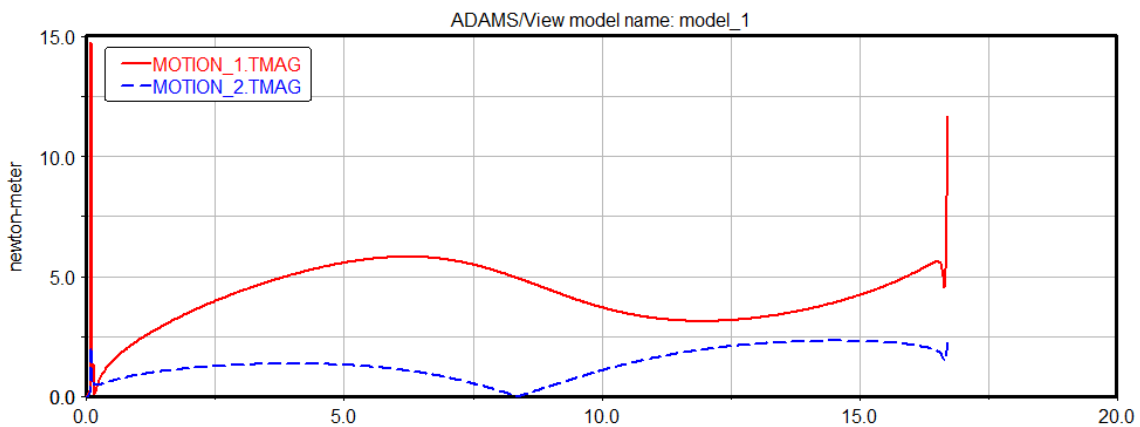


Figura 45 Torque generado Posición 135° [1]

Con una posición final de 135° se obtiene un par en la primera articulación de 14.8Nm y en la segunda articulación de 2.5Nm

- Posición a 180° equivalente a X=0, Y= 900

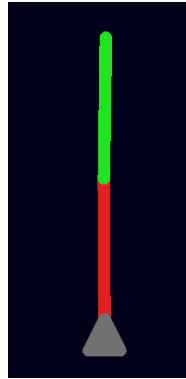


Figura 46 Manipulador a 180° [1]

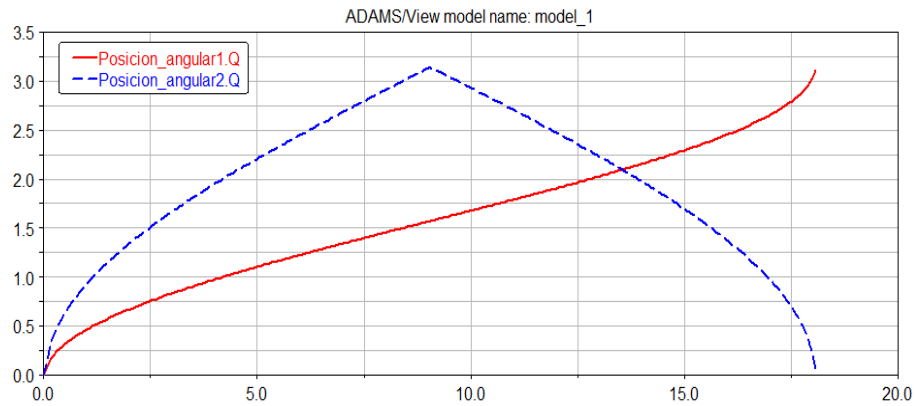


Figura 47 MSC Adams/View Posición 180° [1]

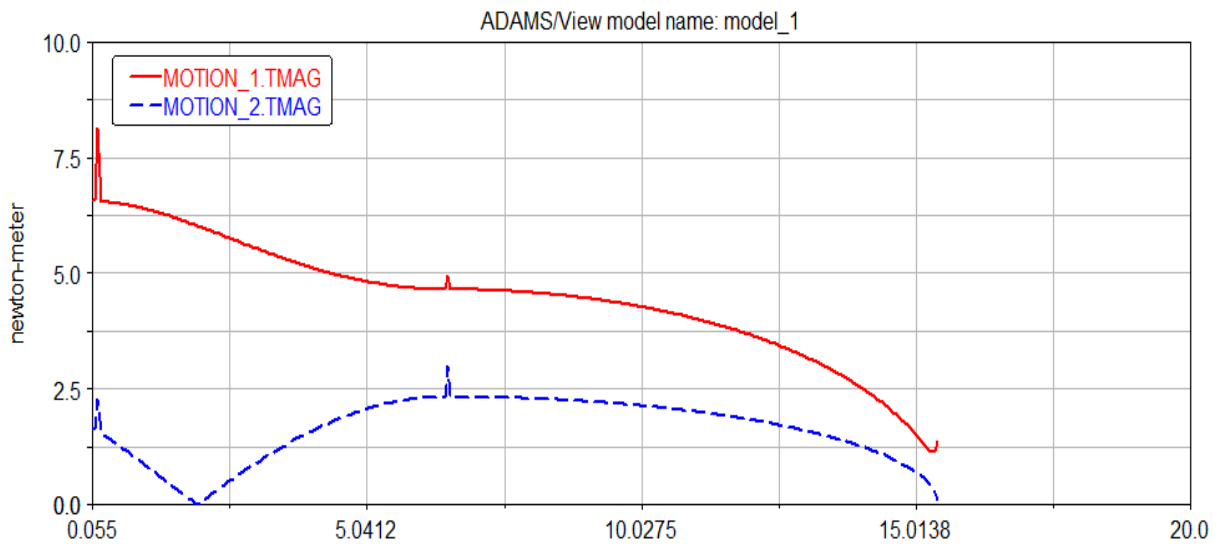


Figura 48 Torque generado Posición 180° [1]

Con una posición de 180° se obtiene un par de 8Nm para la primera articulación y 2.7Nm en la segunda articulación.

Para tener un mayor rango de posibles pares para diferentes posiciones del manipular se hace una simulación con el brazo recogido

- Posición a 45° equivalente a X=318, Y= -318

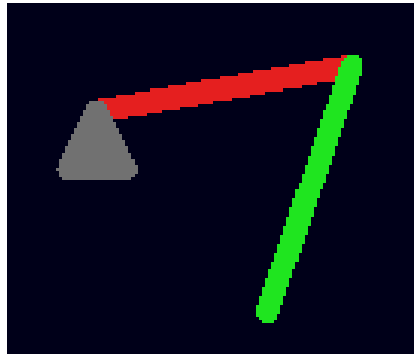


Figura 49 Manipulador a 45° con inclinación [1]

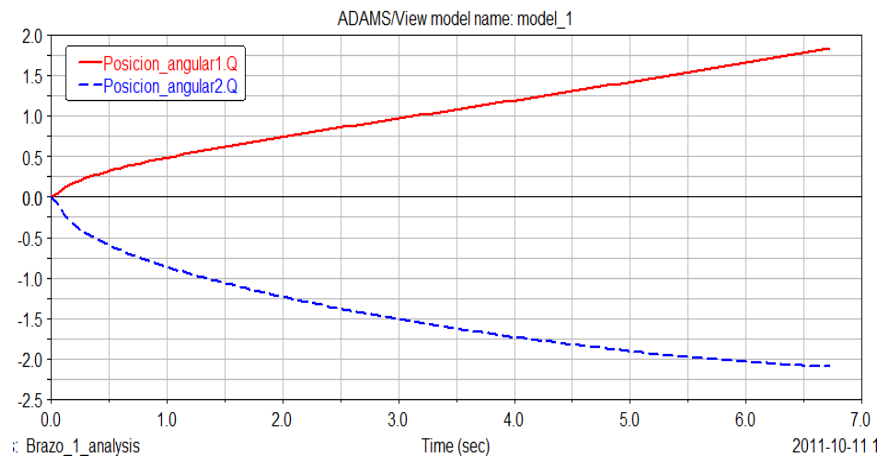


Figura 50 MSC Adams/View 45° [1]

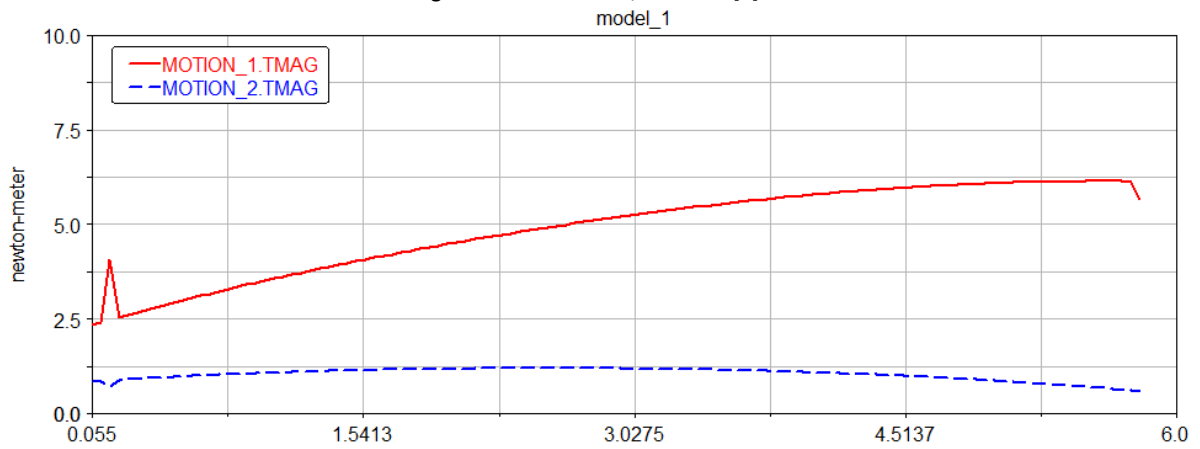


Figura 51 Torque generado 45° [1]

Con una posición de 45° con el manipulador recogido se obtiene un par de 7Nm para la primera articulación y 2Nm en la segunda articulación.

- Posición a 90° equivalente a X=450, Y= 0

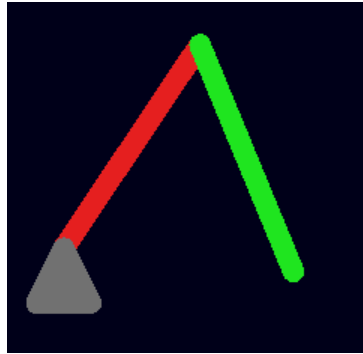


Figura 52 Manipulador a 90° con inclinación [1]

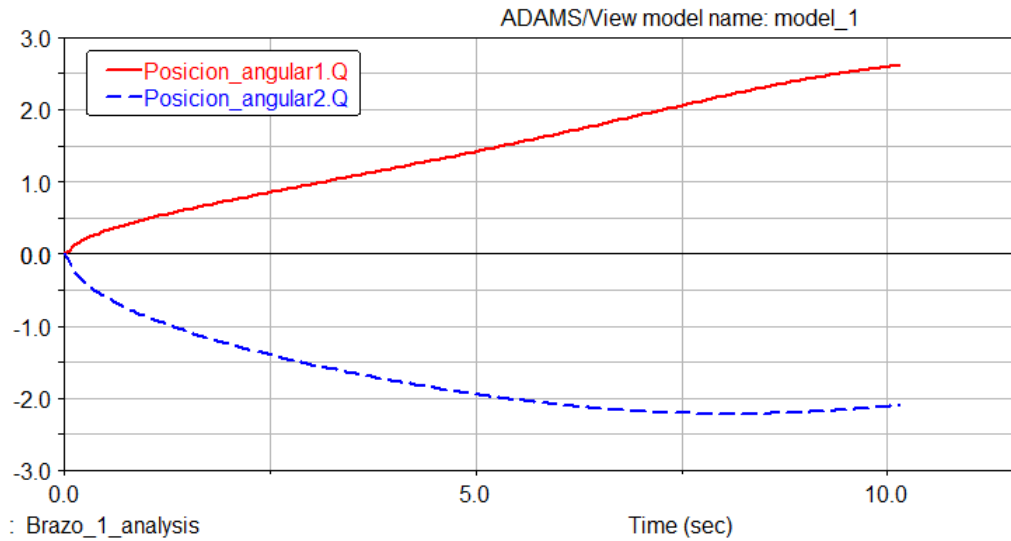


Figura 53 MSC Adams/View 90° [1]

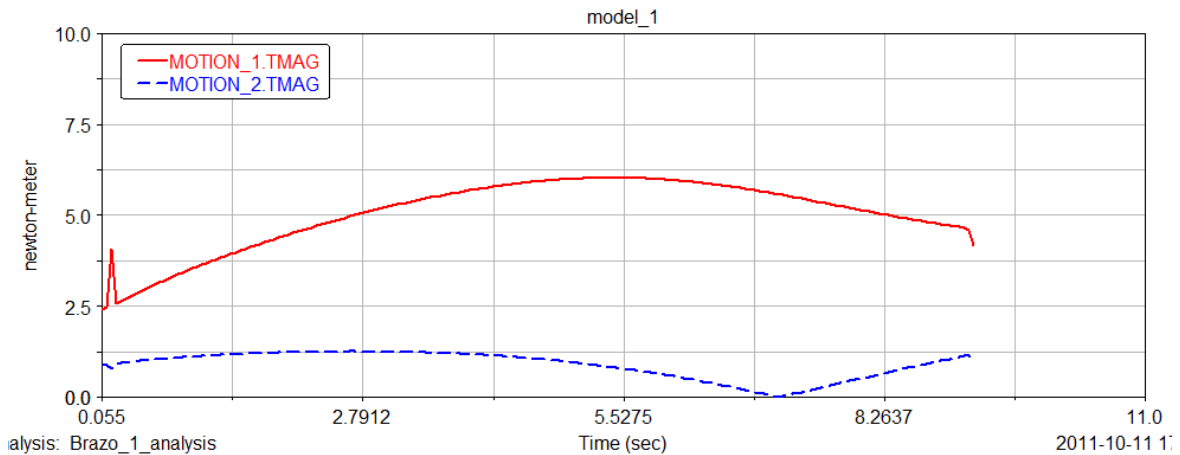


Figura 54 Torque generado 90° [1]

Con un ángulo de  $90^\circ$  y el brazo inclinado se obtiene un par de 6Nm para la primera articulación y un par de 2Nm para la segunda articulación.

- Posición a  $135^\circ$  equivalente a  $X=318, Y= -318$

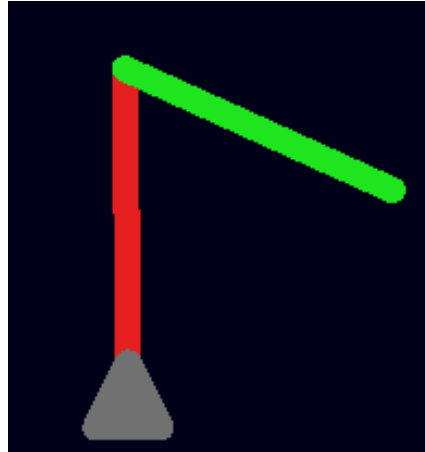


Figura 55 Manipulador a  $135^\circ$  con inclinación [1]

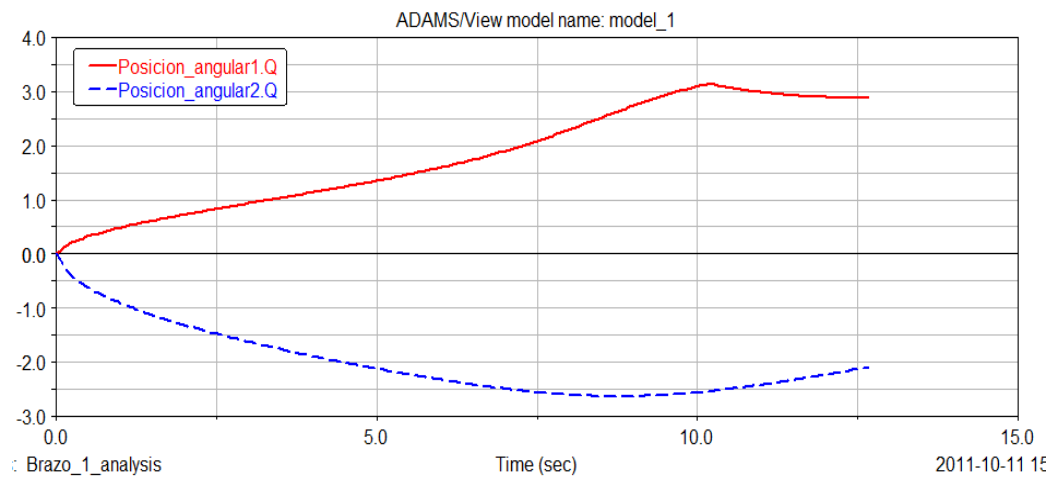


Figura 56 MSC Adams/View  $135^\circ$  [1]

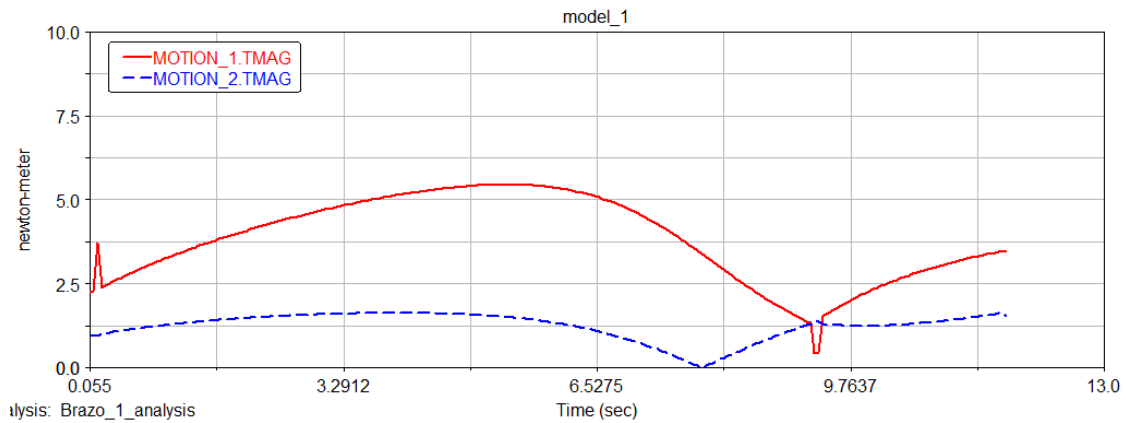


Figura 57 Torque generado  $135^\circ$  [1]

Para la primera articulación se genera un par de 5.5Nm máximo y para la segunda articulación un par de 2Nm.

- Posición a 180° equivalente a X=0, Y= 450

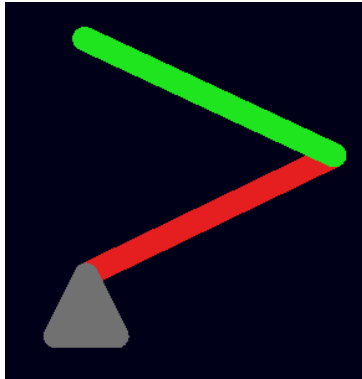


Figura 58 Manipulador a 180° con inclinación [1]

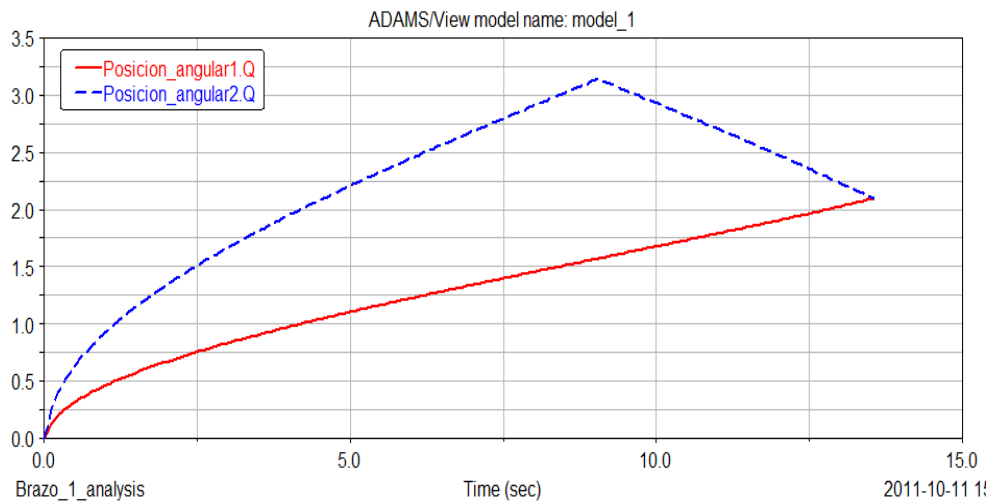


Figura 59 MSC Adams/View 180° [1]

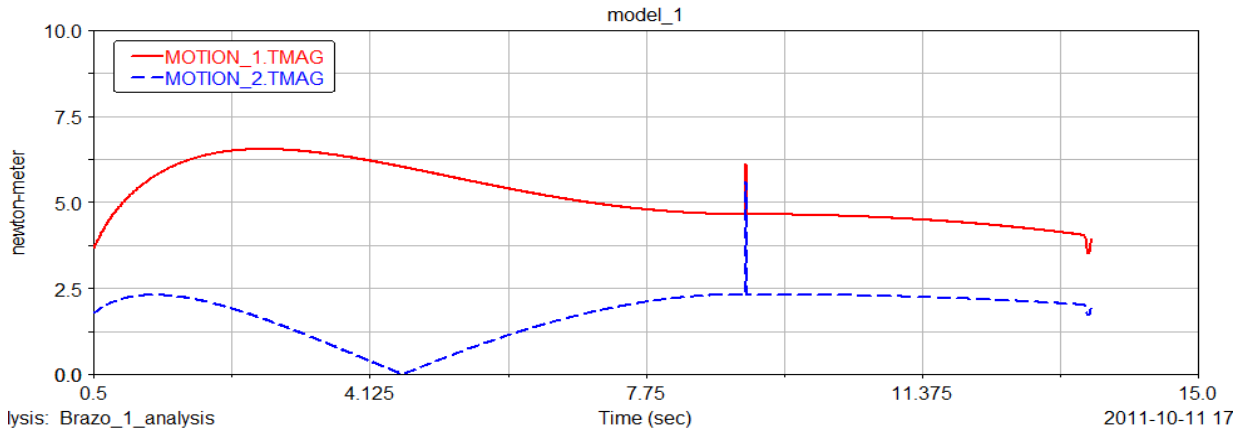


Figura 60 Torque generado 180° [1]



Para esta última posición el par de la primera articulación fue de 7Nm y de la segunda articulación de 2.5Nm.

Par generado	Simulación 1	Simulación 2	Simulación 3	Simulación 4	Simulación 5	Simulación 6	Simulación 7	Simulación 8
Par 1	10	12	14.8	8	7	6	5	7
Par 2	1.7	2.5	2.5	2.7	2	2	5.2	2.5

Tabla 2 Par de motores

Con estos resultados es posible establecer el par máximo a manejar por cada articulación, para la articulación uno se obtiene un par máximo de 14.8Nm y para la segunda articulación de 5.2Nm, con estos parámetros se escogen los motores los cuales se han escogido a través de la página web de specamotor donde nos da la opción de ingresar especificaciones de los motores así mismo ofreciéndonos una variedad de estos.

Los motores escogidos son de marca Kollmorgen con las siguientes especificaciones,

Motor	$T_{max}$ [Nm]	$R_{2ph}$ [ $\Omega$ ]	$L_{2ph}$ [H]	$k$ [ $\frac{Vs}{rad}$ or $\frac{Nm}{A}$ ]	Referencia
1	14.979	9.460	1.330e-1	1.722	B-404-A
2	5.836	5.3	2.810e-2	5.373e-1	B(E)-104-A

Tabla 3 Datos de los motores seleccionados para cada articulación [8][9]

El modelo diagrama de bloques de un motor DC esta dado por:

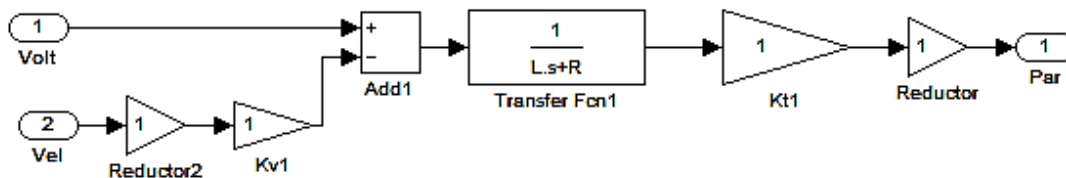


Figura 61 Diagrama Bloques motor DC [1]

### 4.3 Modelo planificador

Para poder obtener las posiciones angulares a partir de una serie de valores de coordenadas espaciales generadas por el planificador, como se muestra en el siguiente esquema,

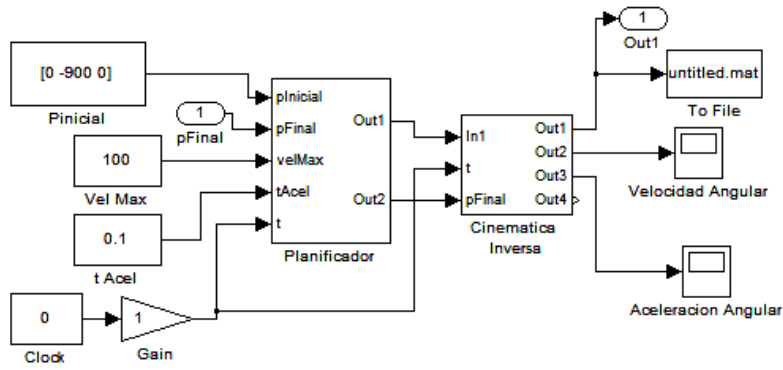


Figura 62 Planificador [1]

Se utiliza la cinemática inversa, donde el planificador nos genera un número de puntos desde el punto de origen hasta el punto final en forma de una recta con un muestreo definido (figura 73), por ejemplo, si estamos es el punto origen Pinicial[0 900] y queremos llegar a Pfinal[636 -636] obtenemos la siguiente trayectoria,

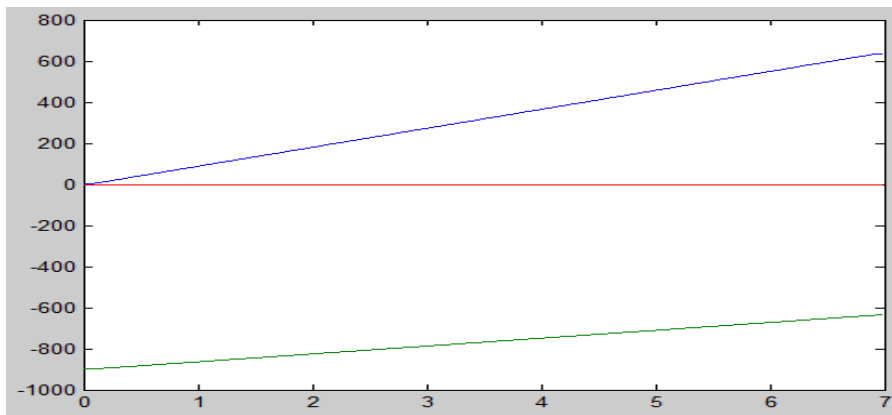


Figura 63 Trayectoria planificador [1]

Como el manipulador es de dos grados de libertad se obtiene dos rectas, una para cada articulación. La recta azul muestra la trayectoria de la primera articulación y la recta verde la de la segunda articulación.

Para más información del planificador remitirse al libro presentado por Luis Ángel Silva Control visual de robots paralelos. Análisis, desarrollo y aplicación a la plataforma robotenis.[4]

#### 4.4 Construcción del modelo de co-simulación

Para poder asociar MSC Adams con matlab, este cuenta con una librería llamada Adams/control, donde se definen entradas y salidas del robot manipulador virtual junto con el programa a asociar, que en nuestro caso es Matlab. Con esto es posible generar un subsistema en matlab lográndose controlar desde este. En MSC Adams/View en el menú Controls/Plant Export, aparecerá una ventana como la figura 15, donde se define las variables de entrada en Input signal(s) y las variables de salida en Output Signal(s), en Target software seleccionaremos Matlab y le damos OK.

Ya con esto se ha creado la conexión entre los dos programas. Al ubicarse en el origen de la carpeta en matlab donde se generó la planta Adams/control se obtiene el bloque del robot manipulador cargándolo con el nombre donde nos mostrara las variables ya establecidas, por ejemplo,

```
>> Controls_par
%%% INFO : ADAMS plant actuators names :
1 Torque_q1
2 Torque_q2
%%% INFO : ADAMS plant sensors names :
1 Posicion_q1
2 Posicion_q2
```

Y por medio del comando *Adams\_sys* en el command window, se obtien,

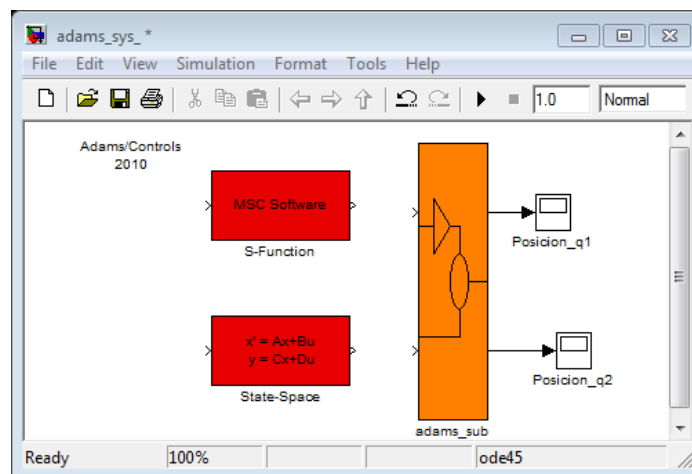


Figura 64 Adams\_sys [1]

Donde el icono naranja Adams\_sub representa al modelo virtual del robot en Adams/view con sus entradas y salidas ya definidas.

## 5. Modelo de Co-simulación y diseño del controlador

En el desarrollo de un nuevo manipulador es necesario el estudio del controlador y del robot como tal, para esto se implementa un modelo virtual en MSC Adams, reemplazando toda la estructura mecánica y la dinámica, logrando recrear en movimientos y fuerzas que puede ejercer físicamente. Debido a esto se optó por el desarrollo de una simulación conjunta entre MSC Adams y Matlab, este proceso se puede observar en la siguiente gráfica mostrando la intervención de cada software en el proceso

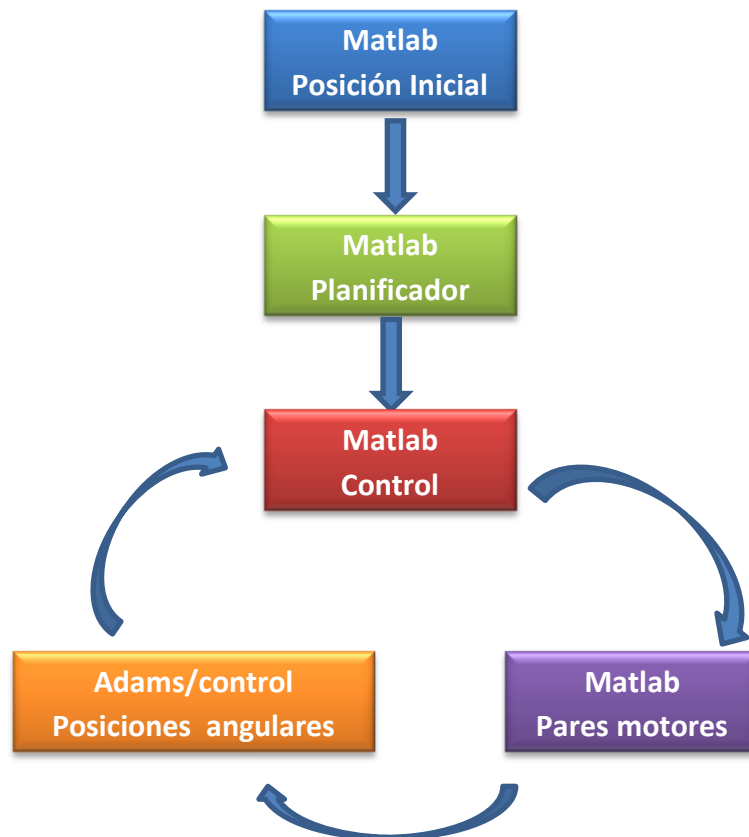


Figura 65 Proceso Adams/Matlab [1]

El software Matlab es el encargado de toda la parte del planificador, control y generación de pares los cuales son enviados a MSC Adams/control que genera el movimiento del manipulador virtualmente, así mismo entregando las coordenadas angulares de cada articulación para el control en lazo cerrado del sistema. Se comprara un controlador PD con un controlador PD con compensación en gravedad, analizando su estabilidad y comportamiento, adicional se mirará la robustez del controlador con un perturbación en el sistema.

## 5.1 Diseño del controlador PD

El controlador PD tiene el siguiente esquema,

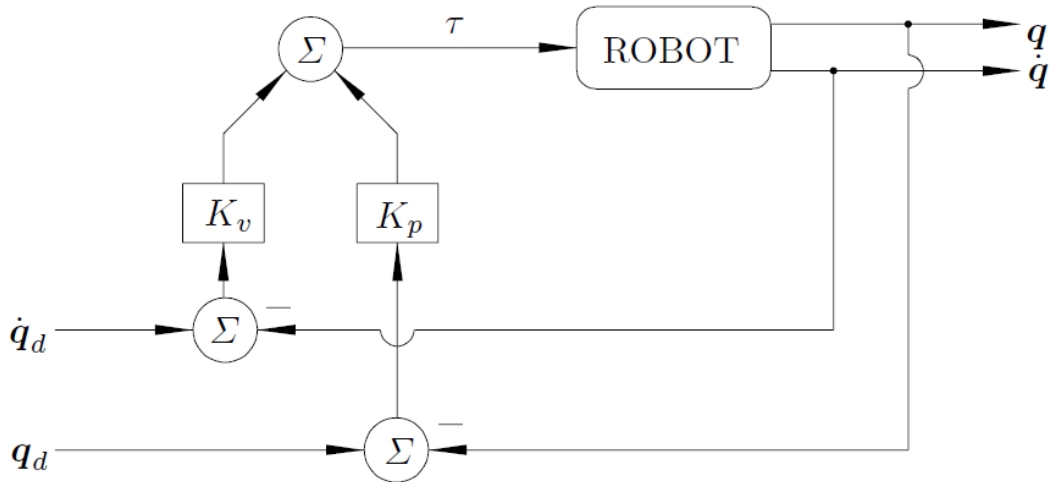


Figura 66 Controlador PD [5]

Donde  $K_v$  y  $K_p$  son matrices de diseño de tamaño  $2 \times 2$ , llamadas ganancias proporcional y derivativa estas son matrices simétricas y positivas. Llevando este esquema al entorno de matlab se obtiene el sistema en la figura 67.

Para los valores de  $K_p$  y  $K_v$  se parten de valores iguales a 0, incrementándolos hasta lograr una adecuada sintonización. En este caso los valores encontrados para las matrices son,

$$K_p = \begin{bmatrix} 450 & 0 \\ 0 & 150 \end{bmatrix}$$

$$K_v = \begin{bmatrix} 18 & 0 \\ 0 & 7 \end{bmatrix}$$

Con estos valores de sintonización se realiza la simulación del manipulador de dos grados de libertad obteniendo un resultado para la una posición de  $45^\circ$  con el brazo totalmente recto generando una trayectoria como se muestra en la siguiente figura 68,

Como se observa al inicio de la trayectoria del manipulador hay una leve perturbación del sistema, esto es generado debido a que el par generado por los motores es bastante elevado para poder romper la inercia de cada eslabón.

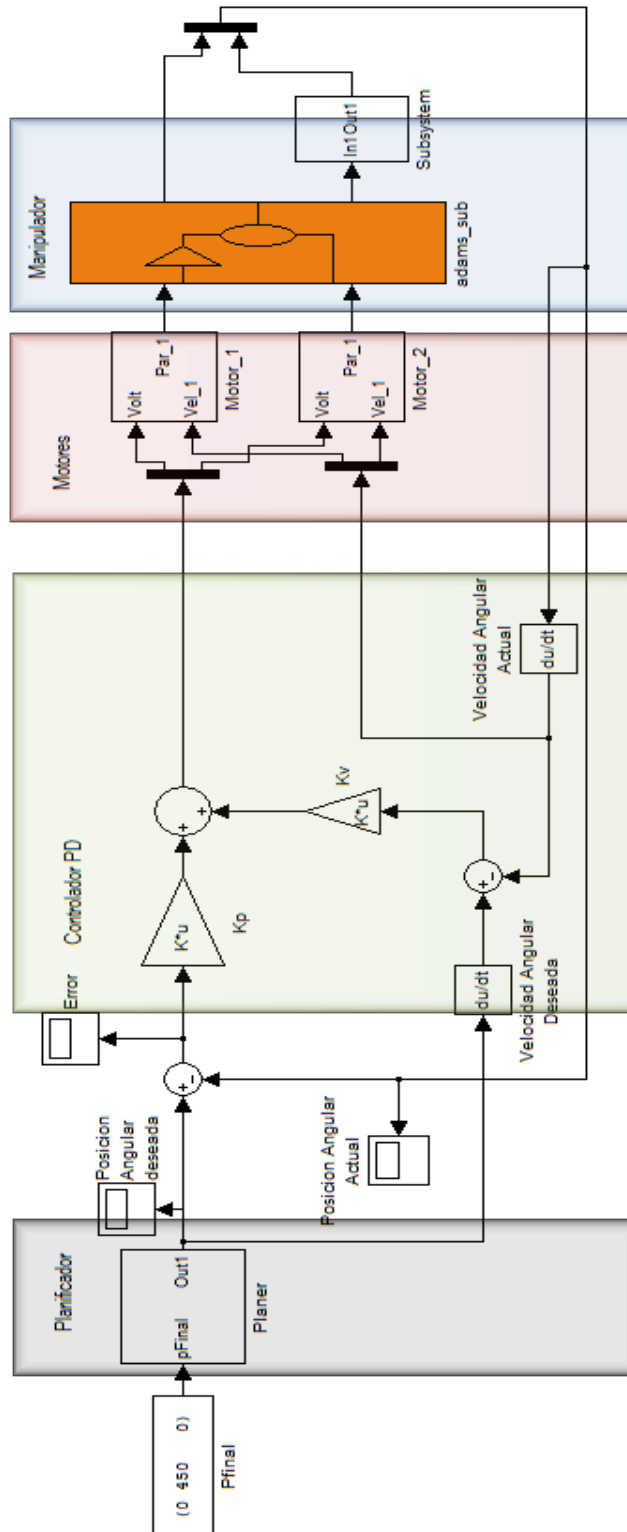


Figura 67 Control PD [1]

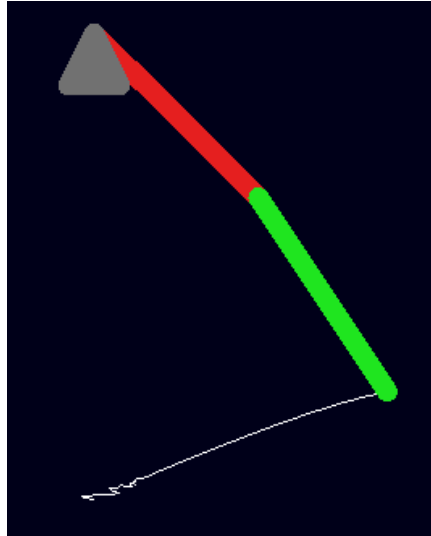



Figura 68 Trayectoria del manipulador con control PD [1]

La toma de muestras se logra por medio del software MSC Adams/PostProcessor MD, el cual está conectado con Adams/View para una lectura fácil de parámetros de simulación.

En el Main Toolbox seleccionamos Plotting o F8, , nos aparecerá una nueva ventana figura 69

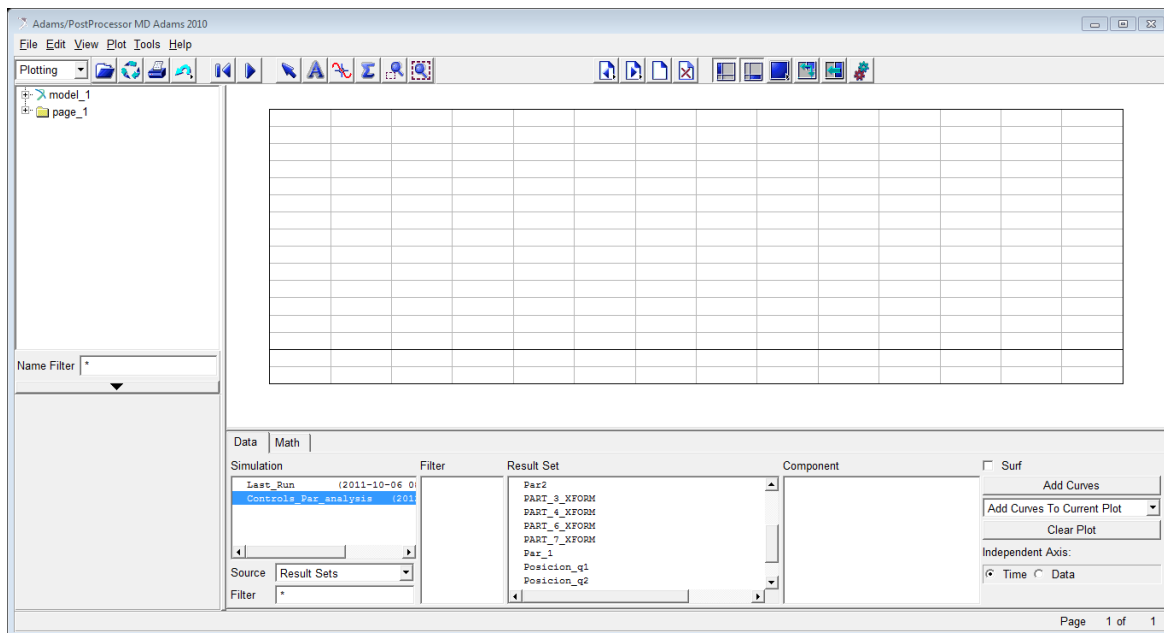


Figura 69 MSC Adams/PostProcessor MD [1]

En donde podemos verificar cada señal generada de nuestro manipulador, en este caso para verificar los pares generados e ingresados por el software Matlab y la trayectoria de nuestro brazo, para esto es necesario cargar la última simulación realizada, se selecciona en el menú File/Import/Results file, donde aparecerá una nueva ventana en la cual

indicaremos el nombre del archivo que se encuentra en la carpeta original del modelo de Adams y el nombre del modelo de este como se observa en la figura 70.

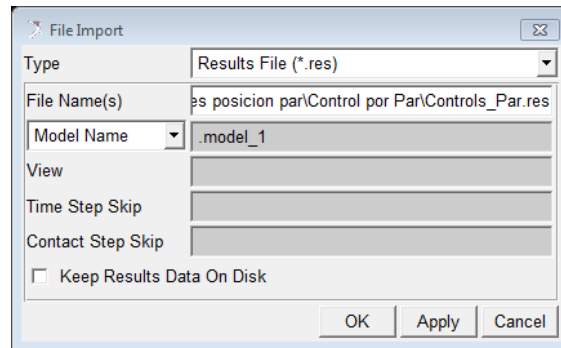


Figura 70 Importar resultados [1]

Con esto se logra visualizar tanto los movimientos realizados como los pares de cada motor, coordenadas angulares entre otras. En las siguientes gráficas se muestra el par generado para cada articulación con control PD.

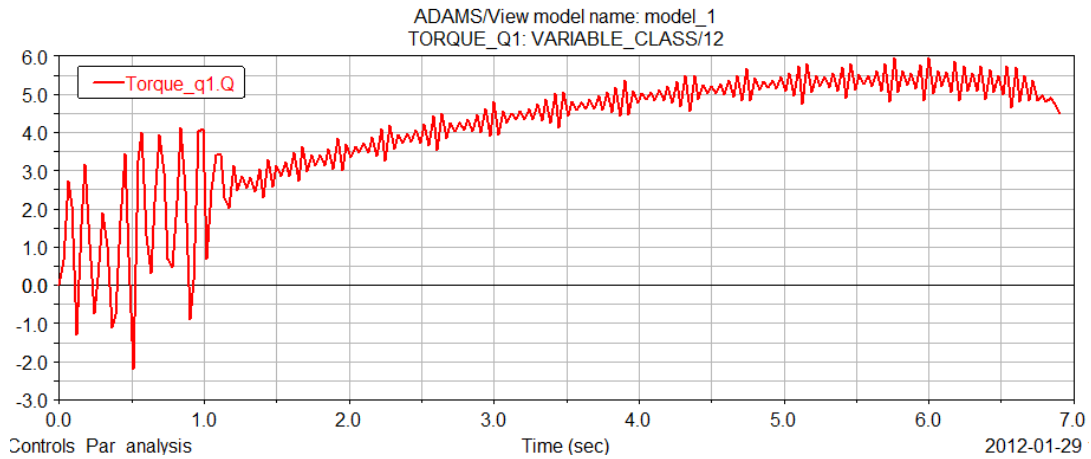


Figura 71 Par articulación 1 [1]

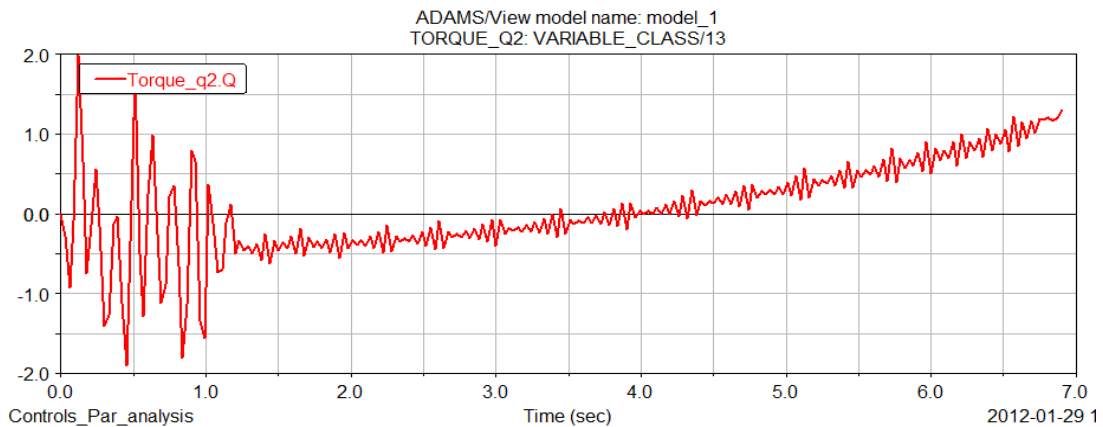


Figura 72 Par articulación 2 [1]



Se aprecia con claridad el alto torque generado para lograr mover el manipulador, generando una perturbación hasta lograr estabilizarse. Durante toda la trayectoria el par generado es algo inestable, presenta una leve oscilación debido a que el manipulador en todo momento intenta regresar a su posición inicial debido a la falta de compensación de gravedad. Como es sabido sin la presencia de un integrador el error no es igual a cero, pero se puede observar que el comportamiento del sistema es continuo y estable con un error bajo, en la siguiente figura la posición actual es equivalente a la deseada. La curva amarilla y fucsia pertenece a la posición deseada y la cian y roja a la posición actual.

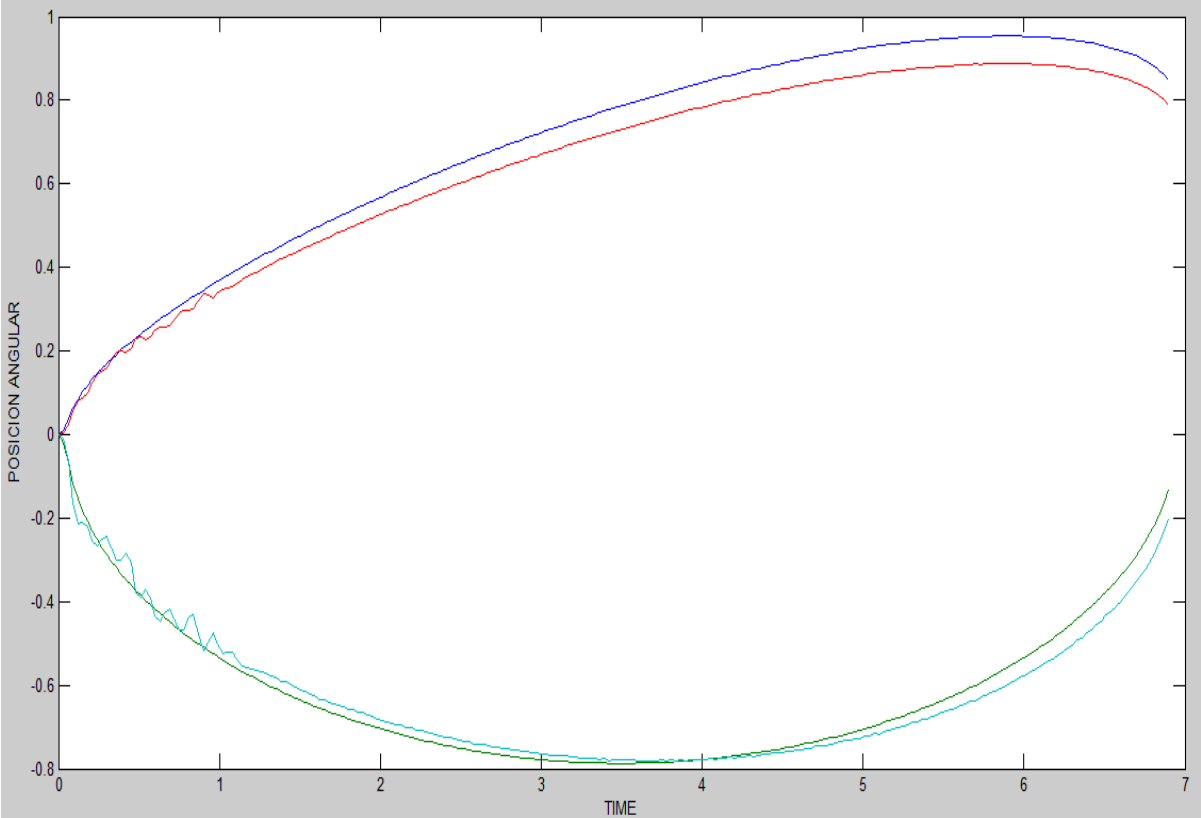


Figura 73 Posición Actual vs deseada control PD [1]

## 5.2 control PD con compensación en gravedad

El sistema está dado por el siguiente esquema

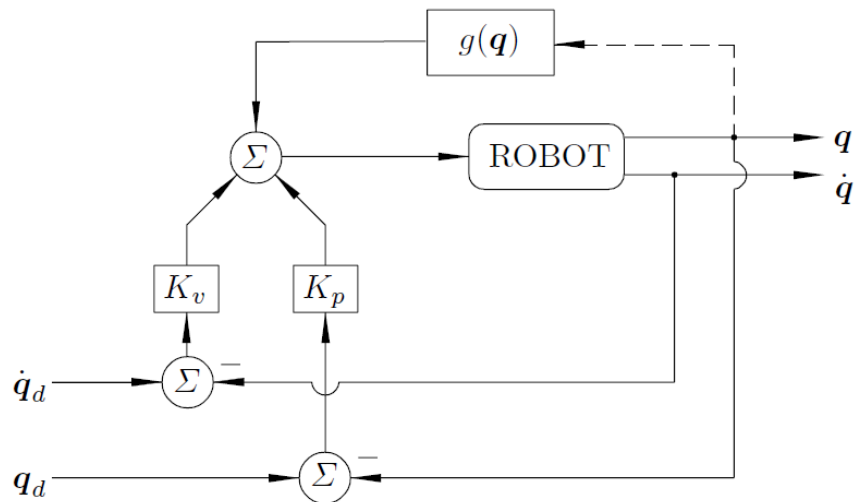


Figura 74 Control PD con compensación en gravedad [5]

Donde el  $g(q)$  es el vector de gravedad hallado anteriormente,  $K_v$  y  $K_p$  son matrices de diseño de tamaño  $2 \times 2$ , llamadas ganancias proporcional y derivativa estas son matrices simétricas y positivas. Llevando este esquema al entorno de matlab se obtiene el sistema en la figura 75.

Con los valores obtenidos en el anterior controlador y para lograr analizar el sistema, se toman los mismos valores para  $K_v$  y  $K_p$ ,

$$K_p = \begin{bmatrix} 450 & 0 \\ 0 & 150 \end{bmatrix}$$

$$K_v = \begin{bmatrix} 18 & 0 \\ 0 & 7 \end{bmatrix}$$

Con estos valores de sintonización se realiza la simulación del manipulador de dos grados de libertad obteniendo un resultado para la una posición de  $45^\circ$  con el brazo totalmente recto. En la figura 76 se muestra el recorrido del efector final,

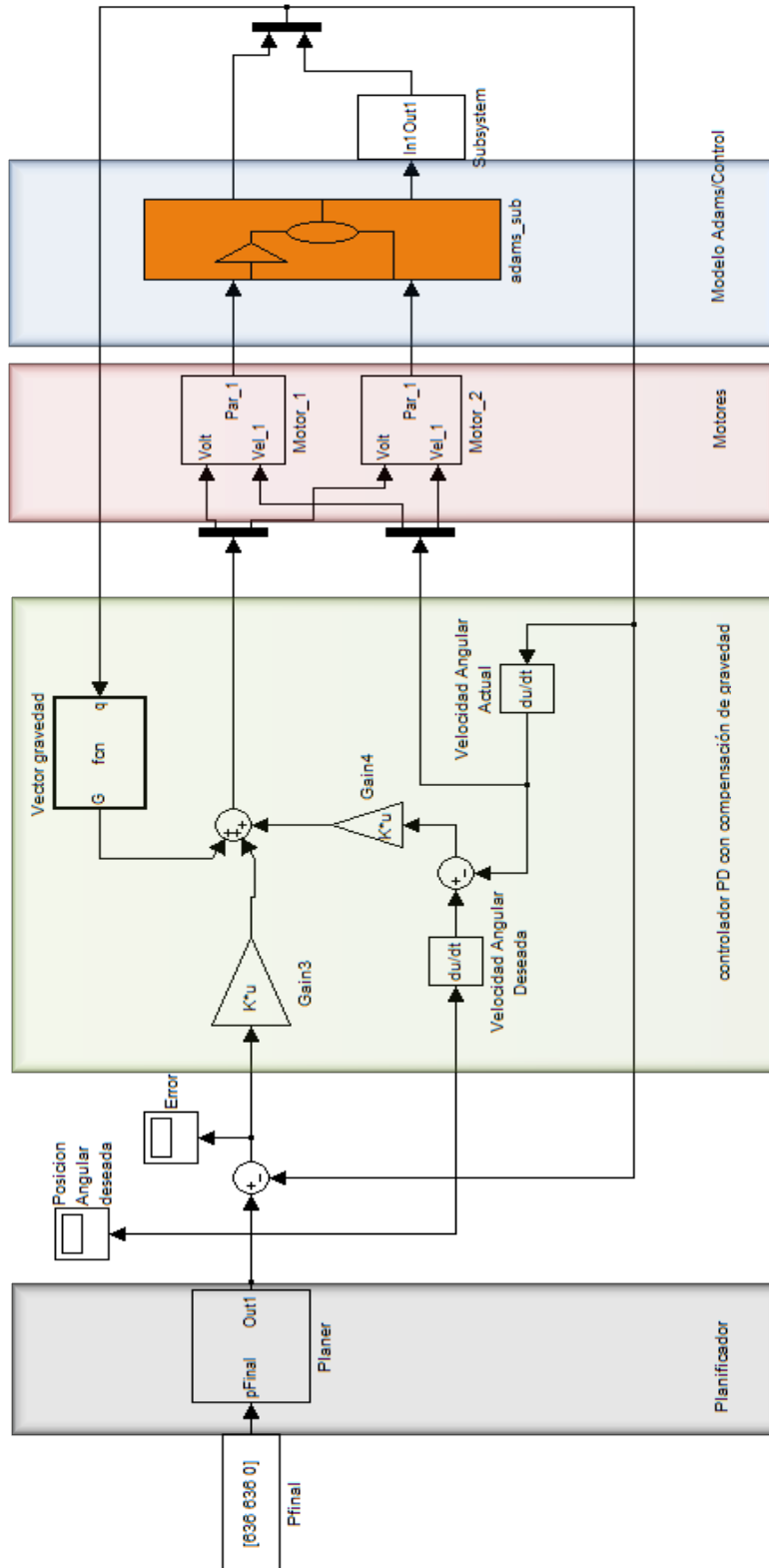


Figura 75 Diseño matlab del sistema [1]

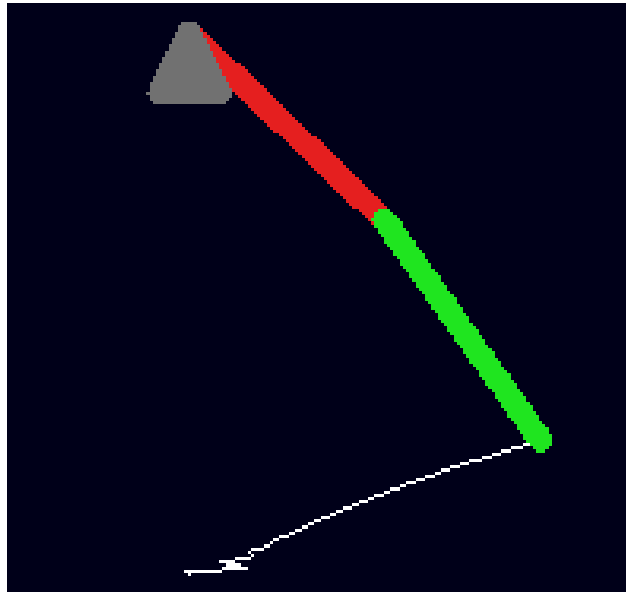


Figura 76 Recorrido del efector final [1]

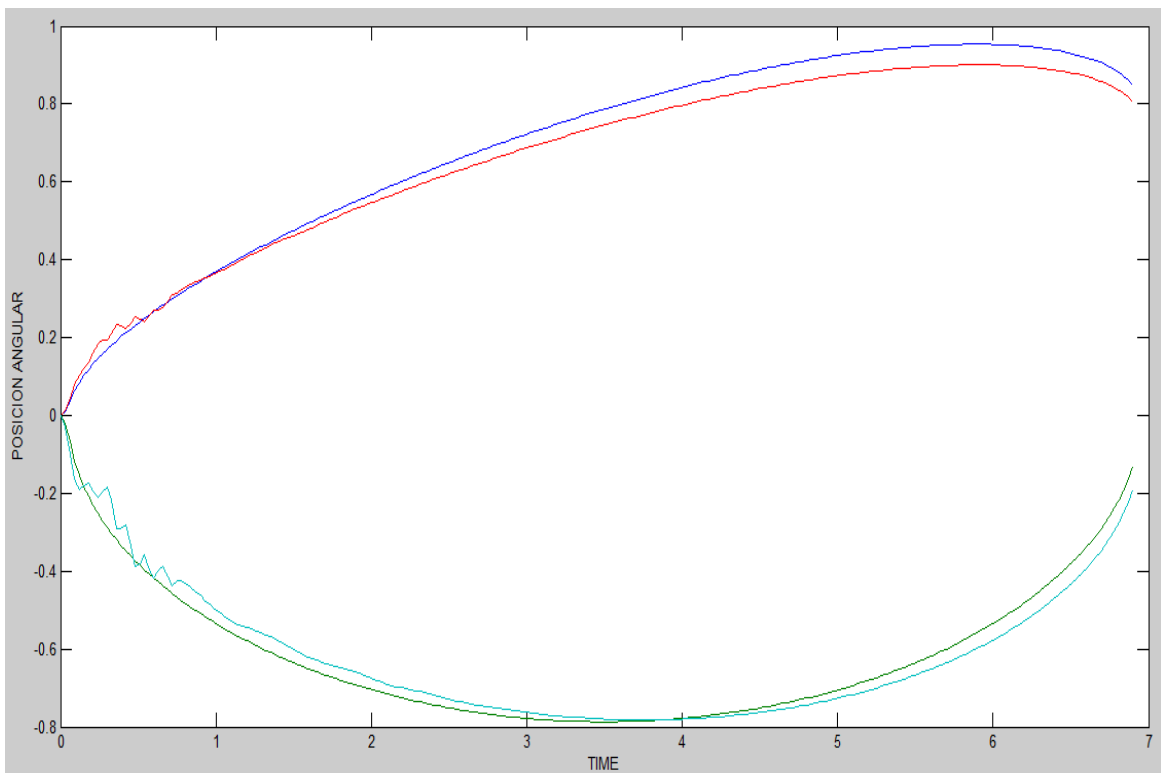


Figura 77 posición actual vs posición deseada [1]

El error obtenido con el uso de un controlador PD con compensación en gravedad, se aprecia bastante reducida, que no influye mayormente en el correcto funcionamiento del manipulador.

Los pares ejercidos por las articulaciones son,

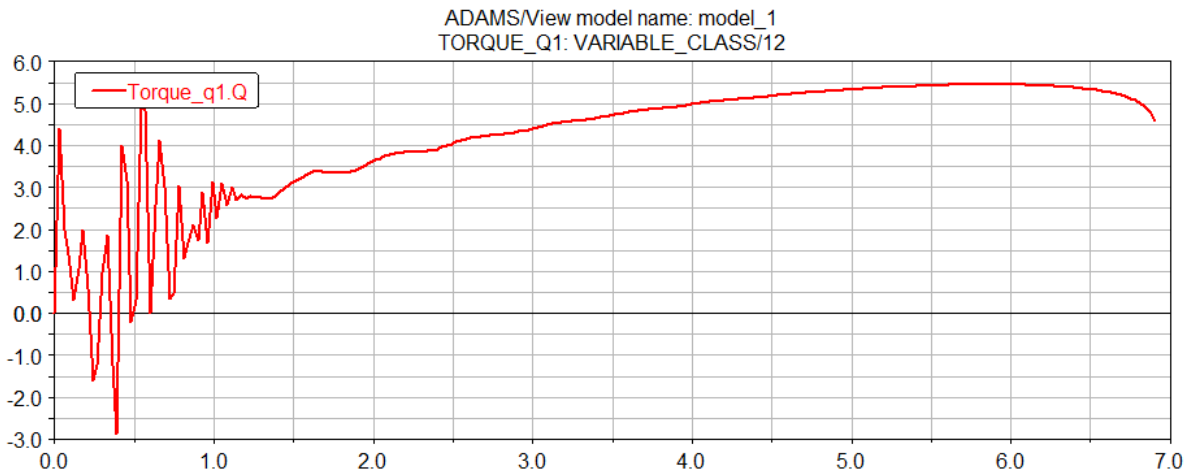


Figura 78 Par primera articulación [1]

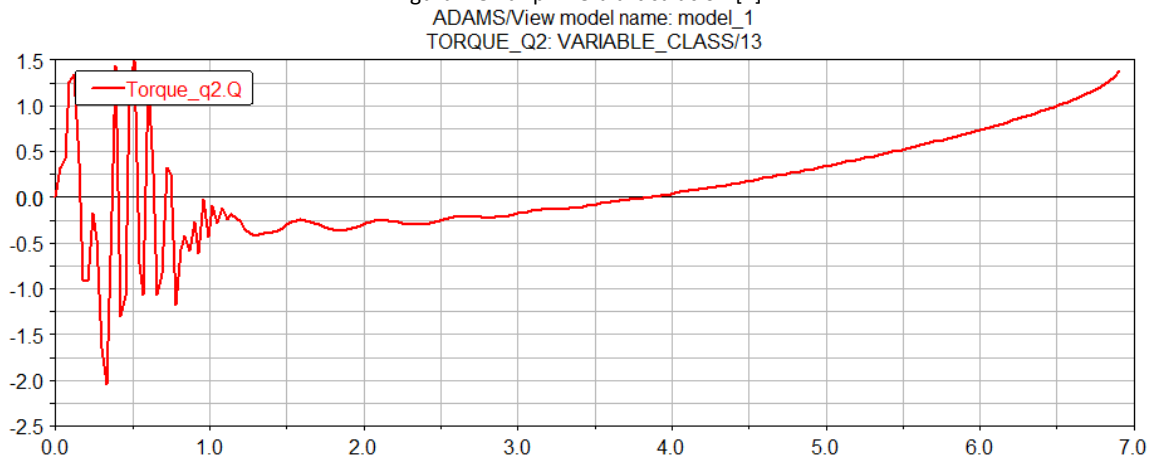


Figura 79 Par segunda articulación [1]

Como se logra observar aún se presenta una perturbación al inicio de la trayectoria debido a los motores, luego la compensación en gravedad del sistema logra obtener una mejor estabilidad en los pares generados, presentando una respuesta más limpia sin variantes. El error angular obtenido es menor al 6% como se observa en la siguiente figura.

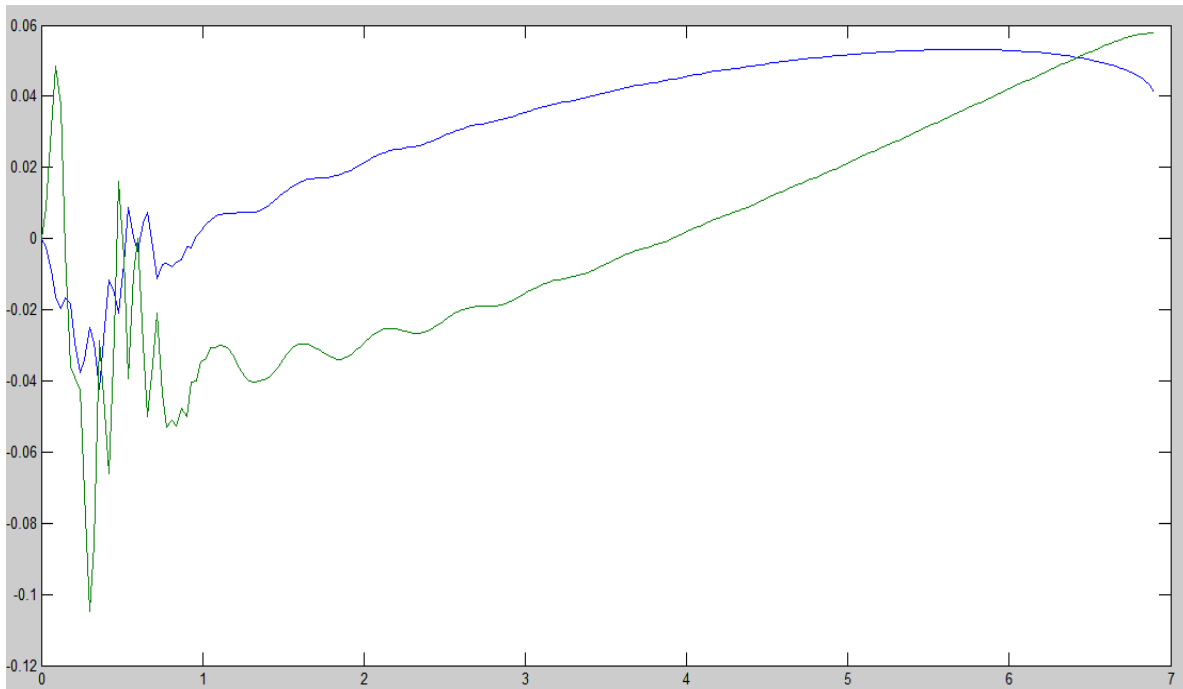


Figura 80 Error posición angular [1]

### 5.3 Evaluación de la robustez del sistema

Para mirar que tan eficiente es el sistema ante alguna perturbación se adiciona un ruido al sistema como se muestra en el siguiente figura 81.

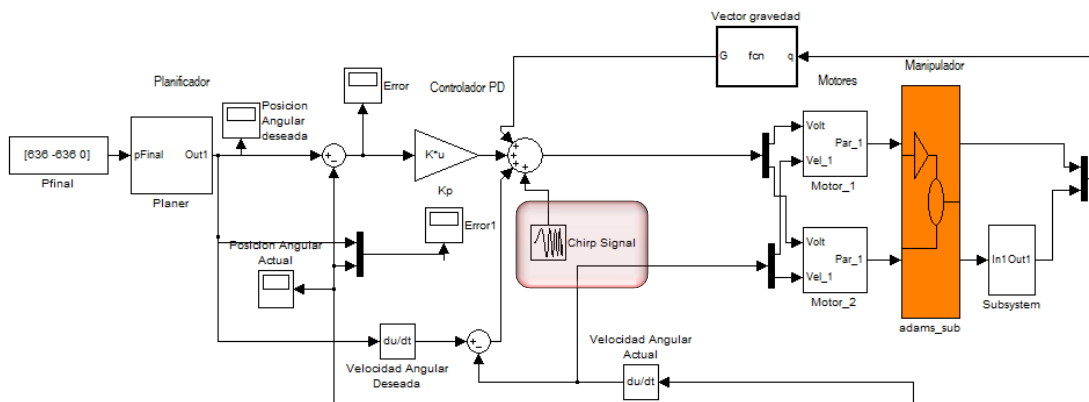


Figura 81 Perturbación del sistema [1]

La trayectoria del manipulador se alcanza a alterar con esta gran perturbación debido a q ya no presenta una linealidad como se muestra en la siguiente figura

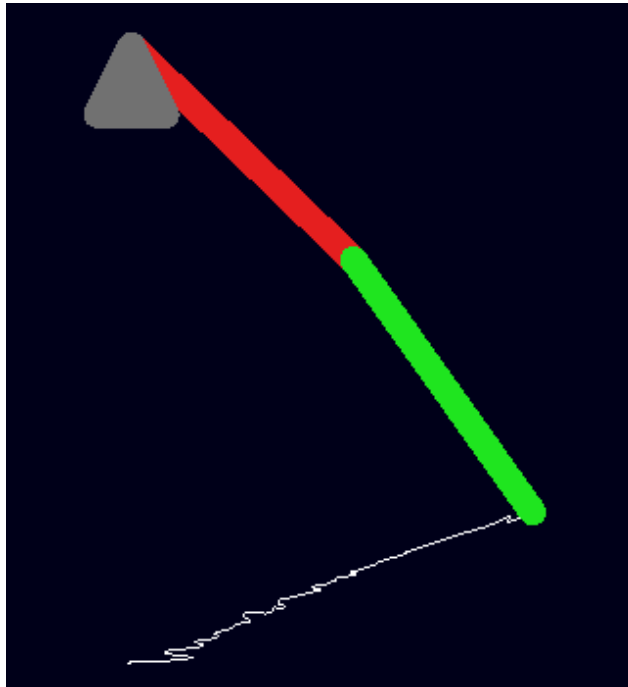


Figura 82 Trayectoria con perturbación [1]

Aun así, la respuesta del sistema funciona con este controlador, debido a que llega a su posición final como es esperado. La señal de perturbación que se inyecta al sistema se muestra en la siguiente figura

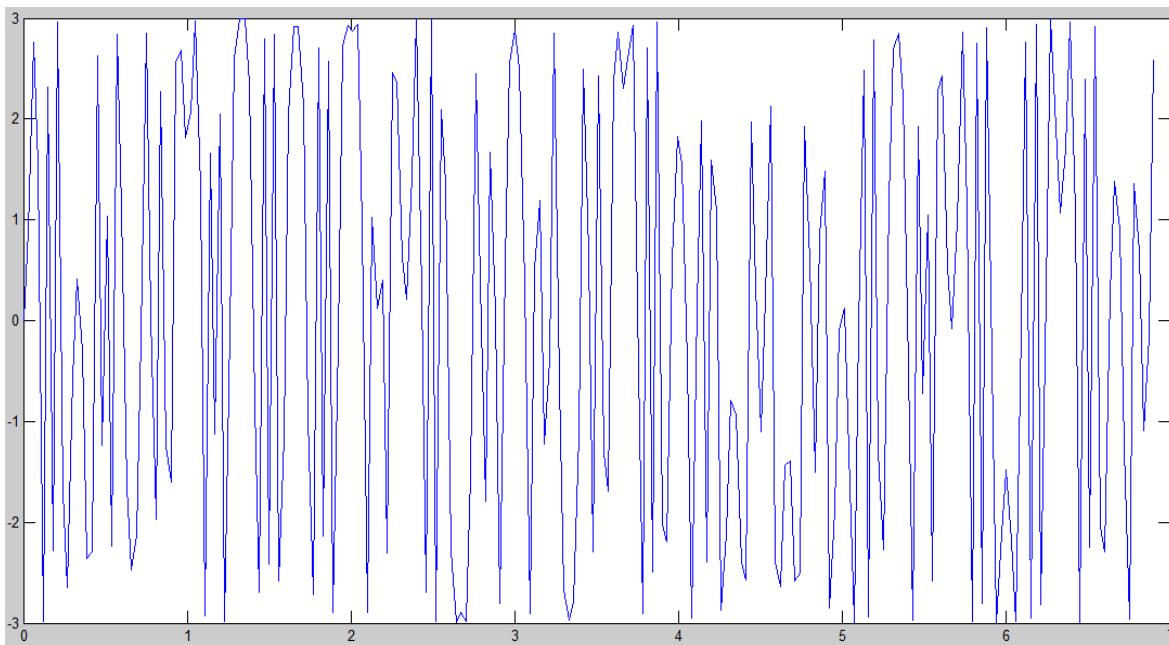


Figura 83 Perturbación [1]

El seguimiento de la trayectoria en posiciones angulares es el siguiente

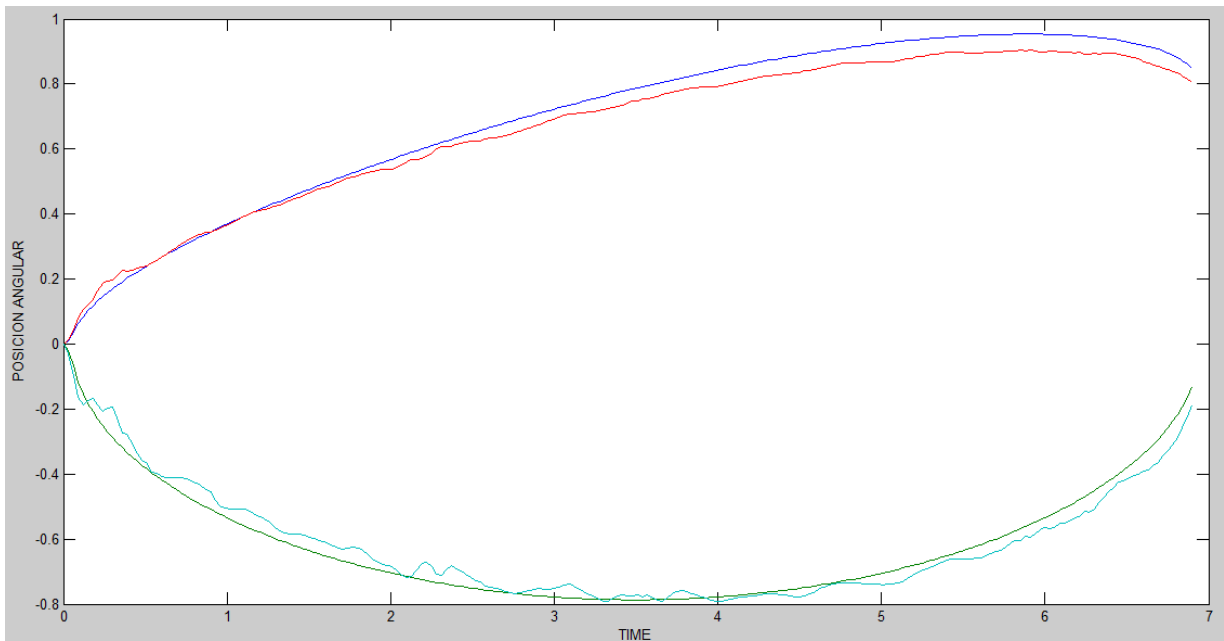


Figura 84 Posiciones angulares [1]

Se aprecia que la trayectoria actual se alcanza a afectar, pero aún es equivalente a la deseada conservando el mismo error.

En la siguiente figura se muestra la salida del control,

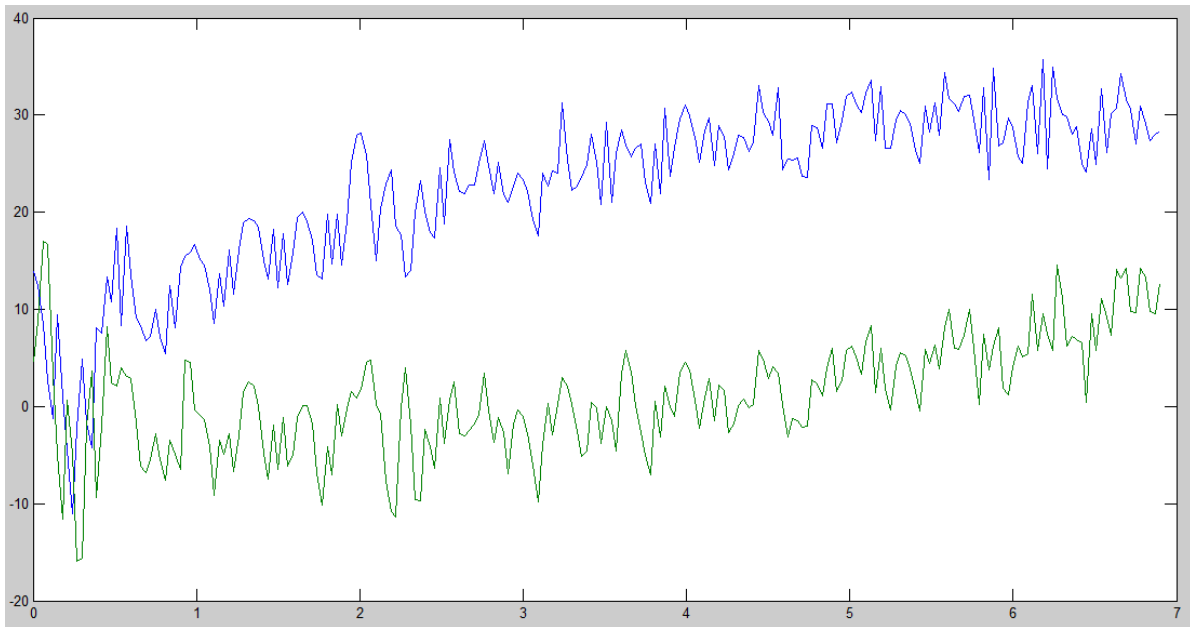
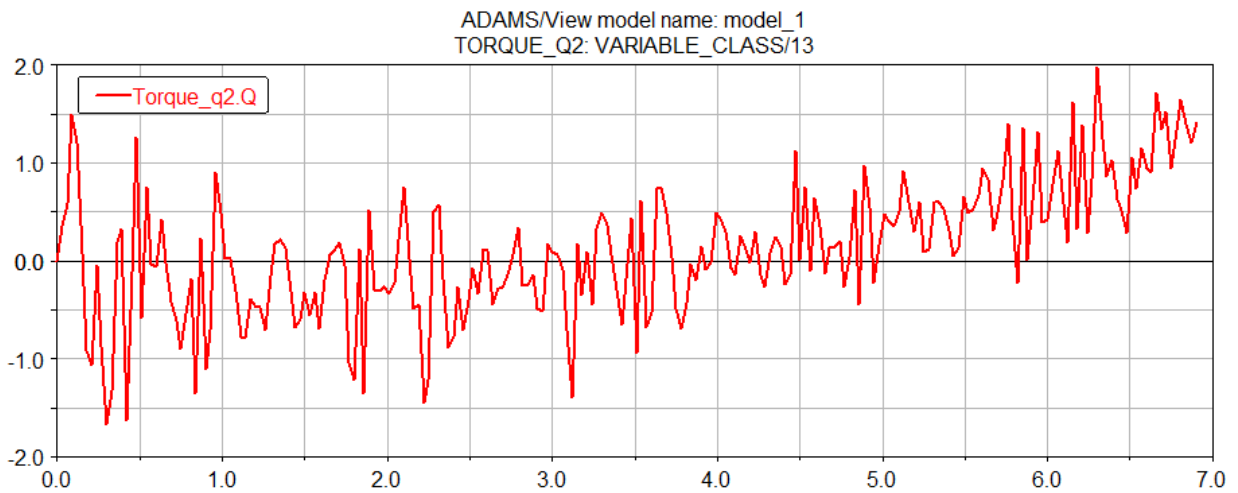
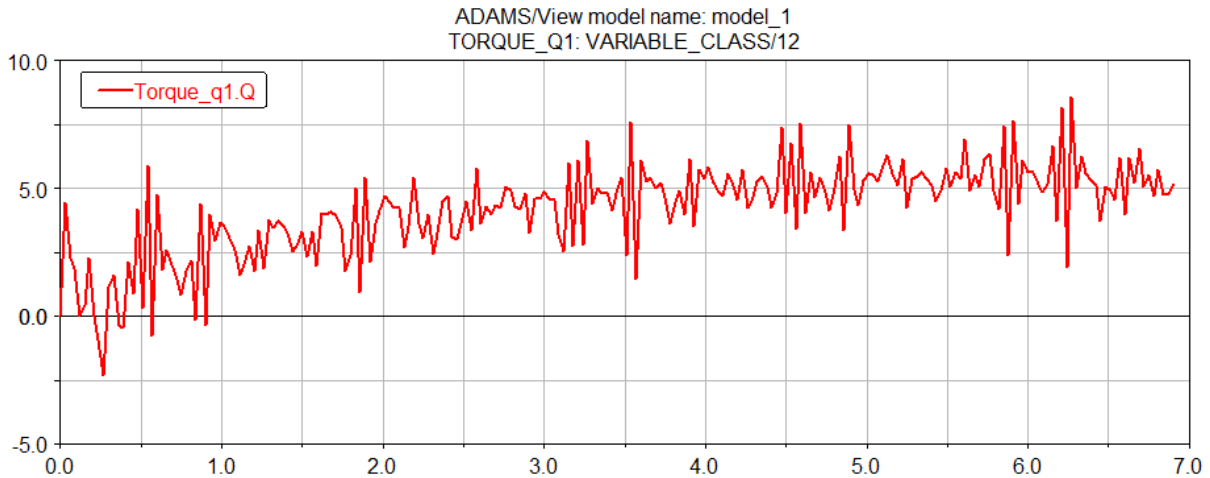


Figura 85 Salida del control [1]



Los pares generados en la perturbación son los siguientes



Si se compara con los pares generados sin la perturbación (*figuras 90-91*), presentan la misma tendencia.

#### 5.4 Validación del controlador para diferentes posiciones deseadas en el espacio de trabajo

El objetivo de este controlador es que sea único para cualquier posición debido a su no linealidad. Dependiendo de la sintonización se logra un movimiento suave y muy parejo con la trayectoria deseada. A continuación se muestra diferentes posiciones de trabajo para el manipulador de dos grados de libertad teniendo en cuenta el color de cada curva de la siguiente tabla,

Color	Posición
Azul	Deseada
Verde	Deseada
Cian	Actual
Roja	Actual

Tabla 4 Posición de cada color [1]

Inicialmente se prueba una posición de 90° del manipulador, la siguiente gráfica del robot muestra el recorrido total de la trayectoria, indicando con una línea blanca cual fue las posiciones que el planificador generó.

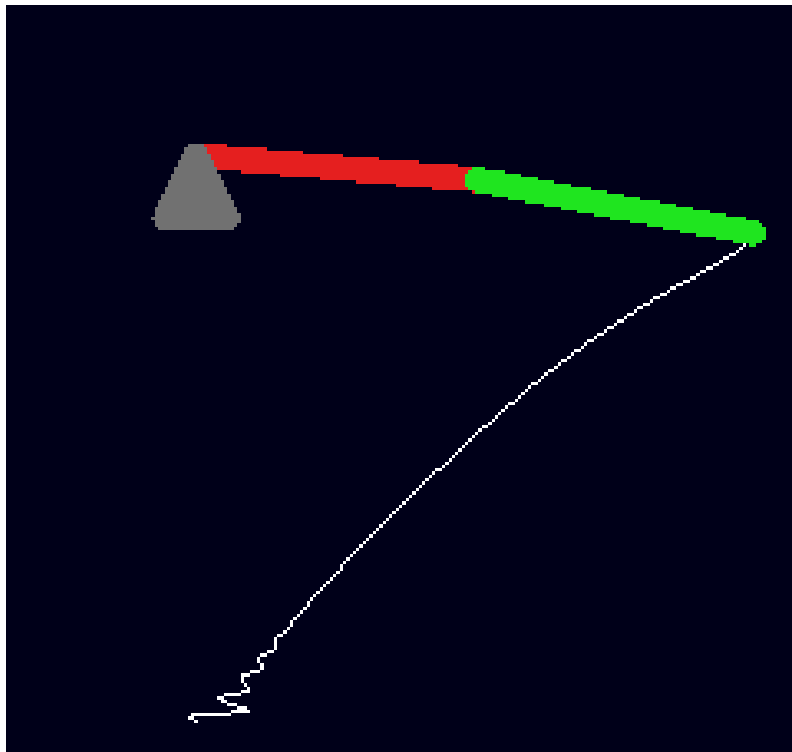


Figura 88 Manipulador a 90° [1]

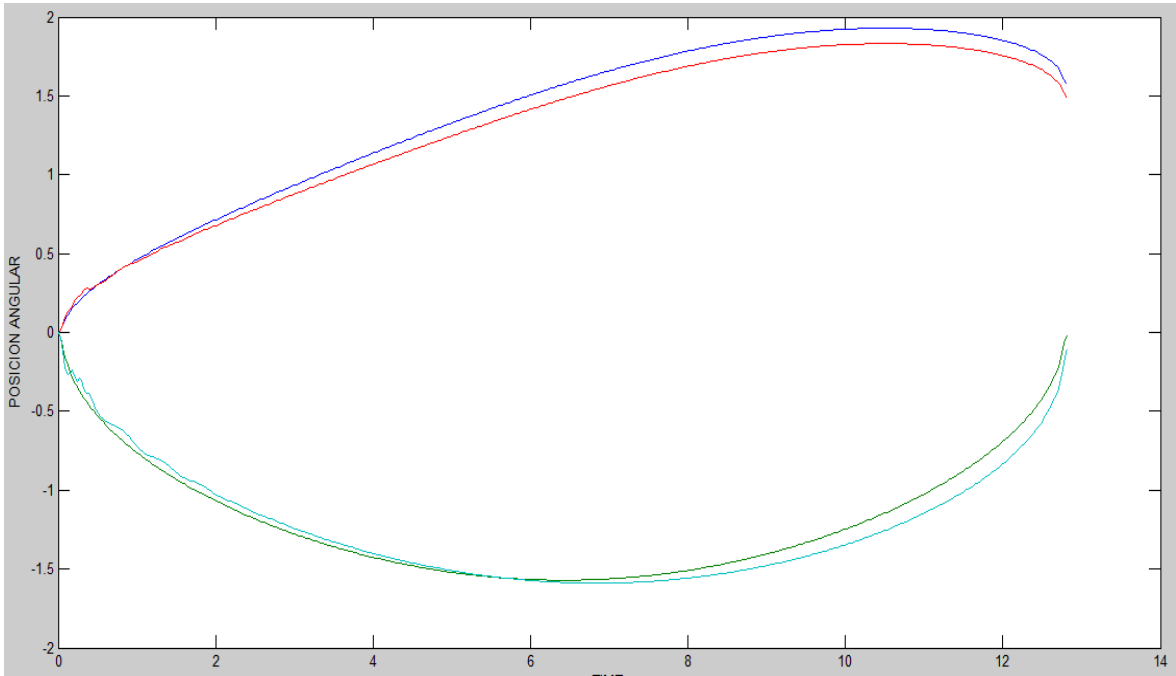


Figura 89 Posición angular deseada vs actual de 90 [1]

Señal del control PD con compensación de gravedad

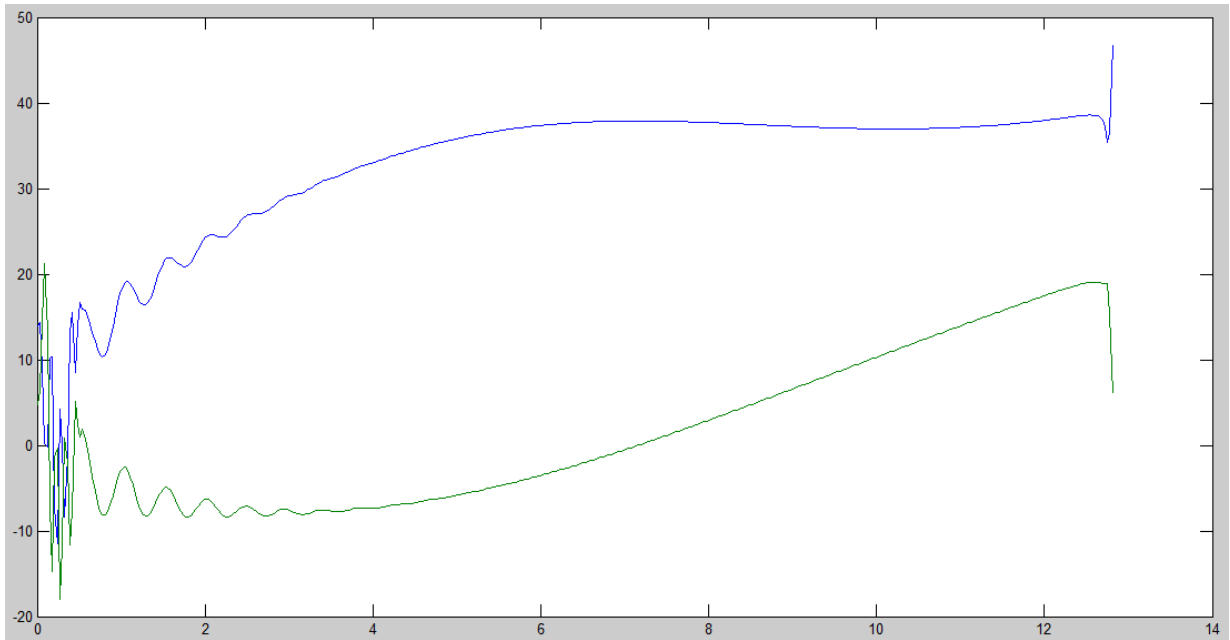


Figura 90 Señal del control PD con compensación de gravedad [1]

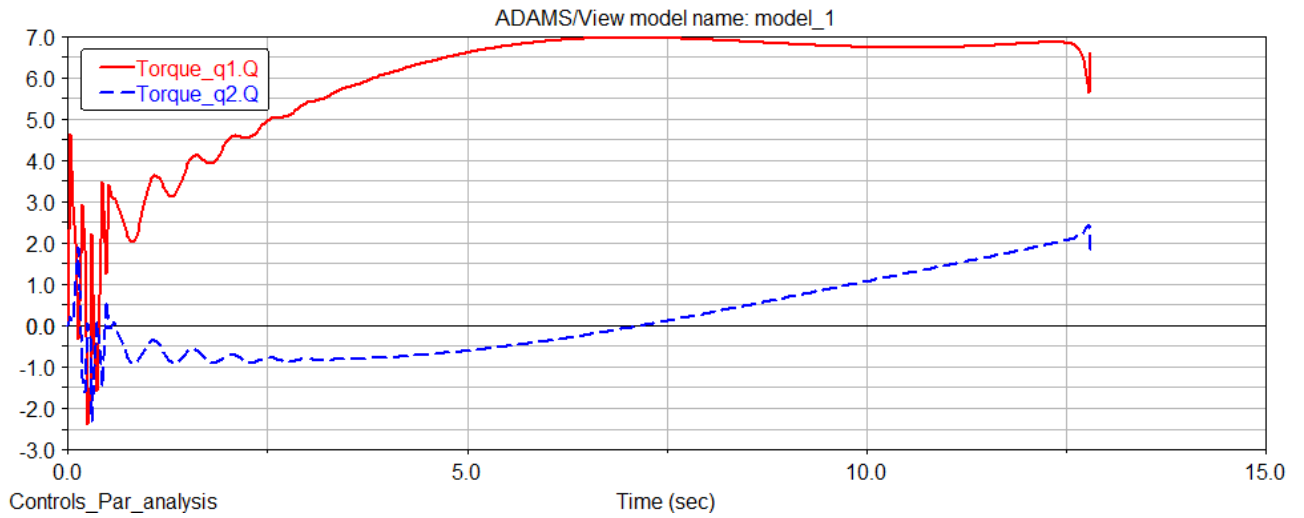


Figura 91 Par generado [1]

Para una posición de 135°

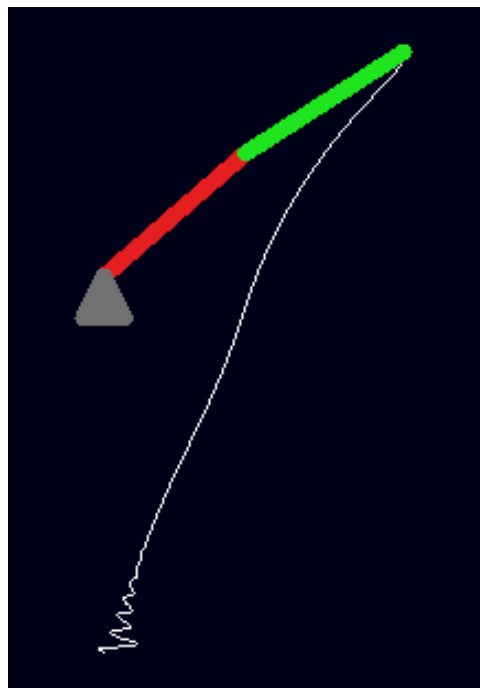


Figura 92 Manipulador a 135° [1]

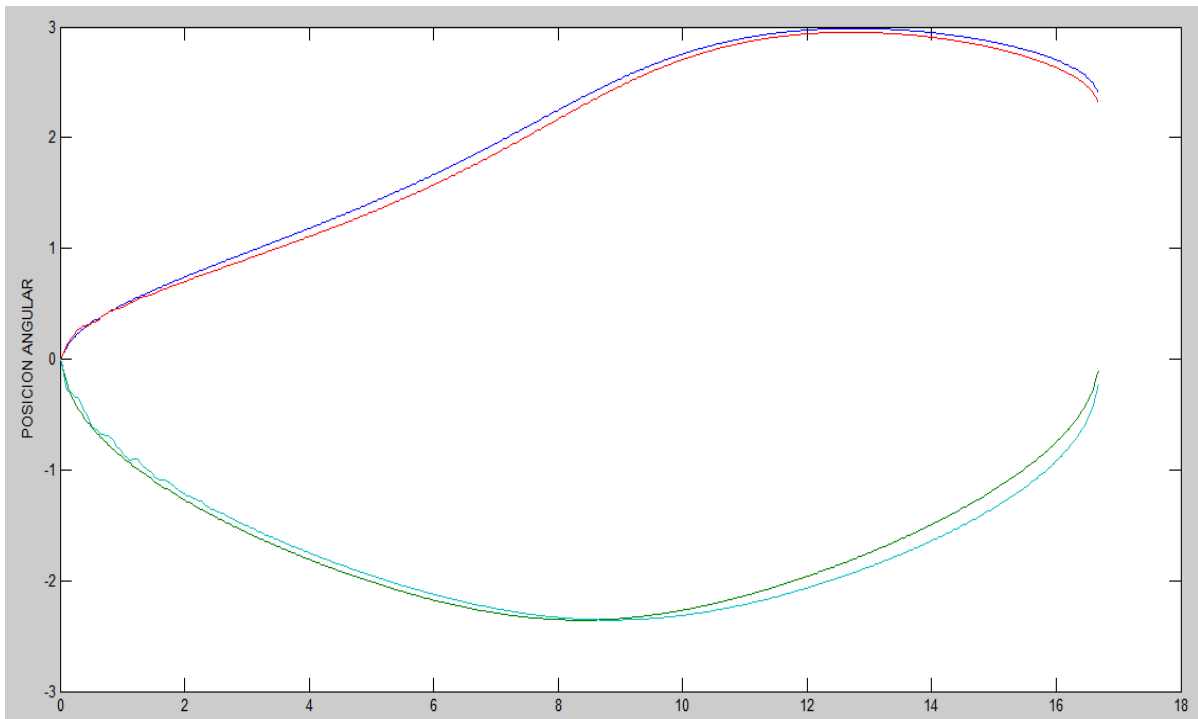


Figura 93 Posición angular actual vs deseada a 135° [1]

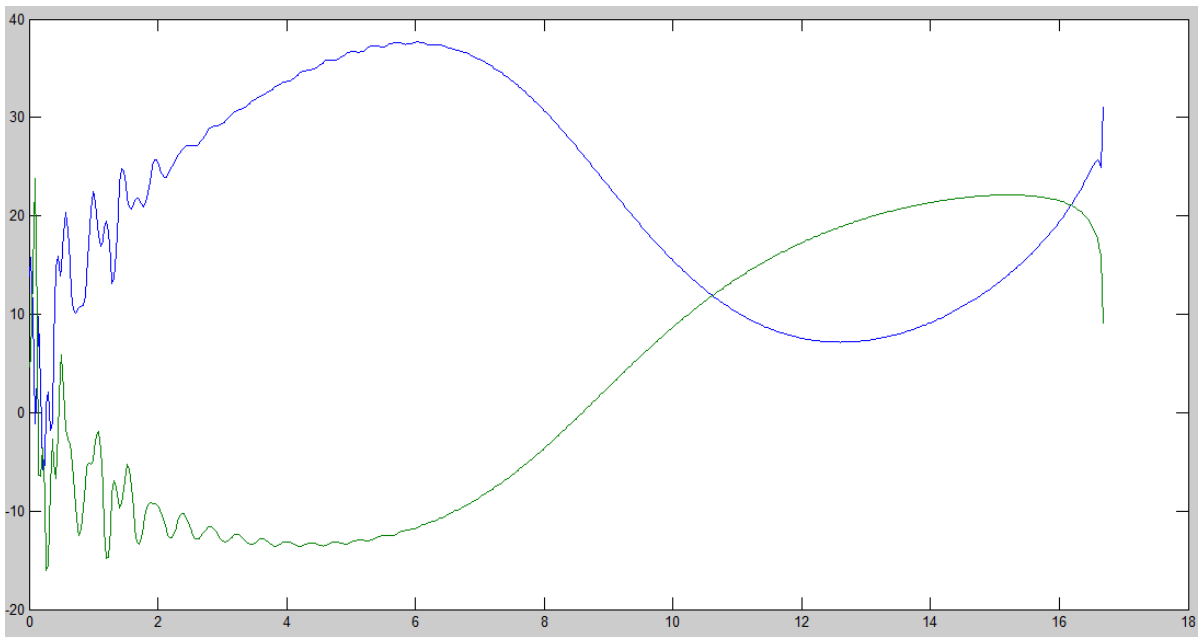


Figura 94 Señal del control para 135° [1]

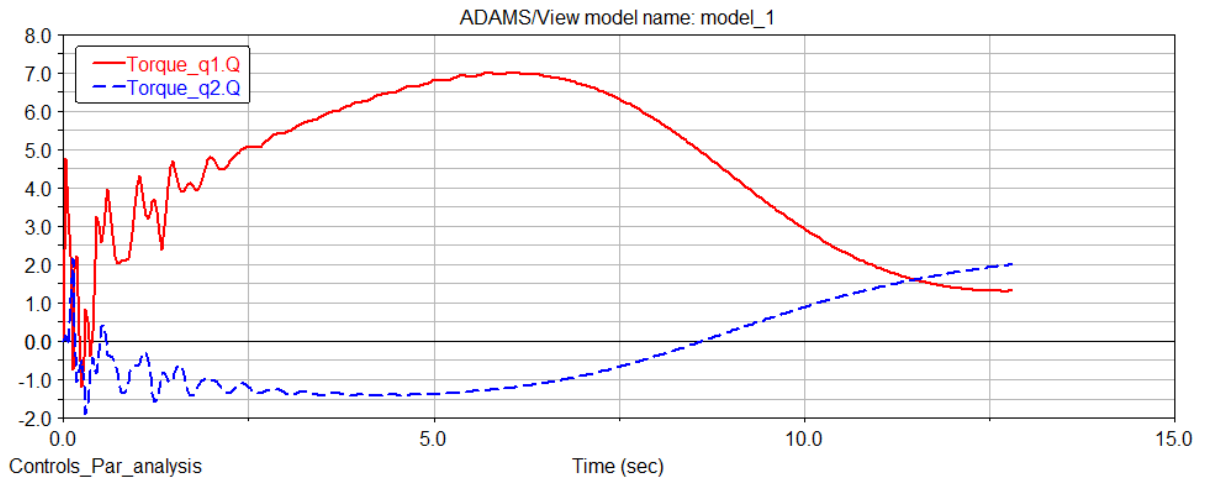


Figura 95 Par generado a 135° [1]

Para una posición de 45° con el brazo doblado

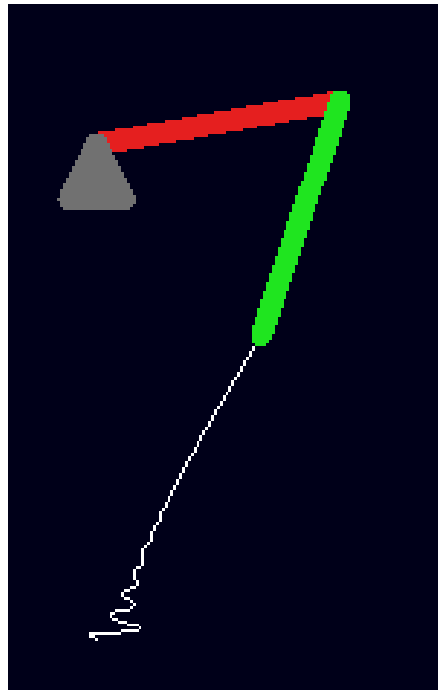


Figura 96 Manipulador a 45° no recto [1]

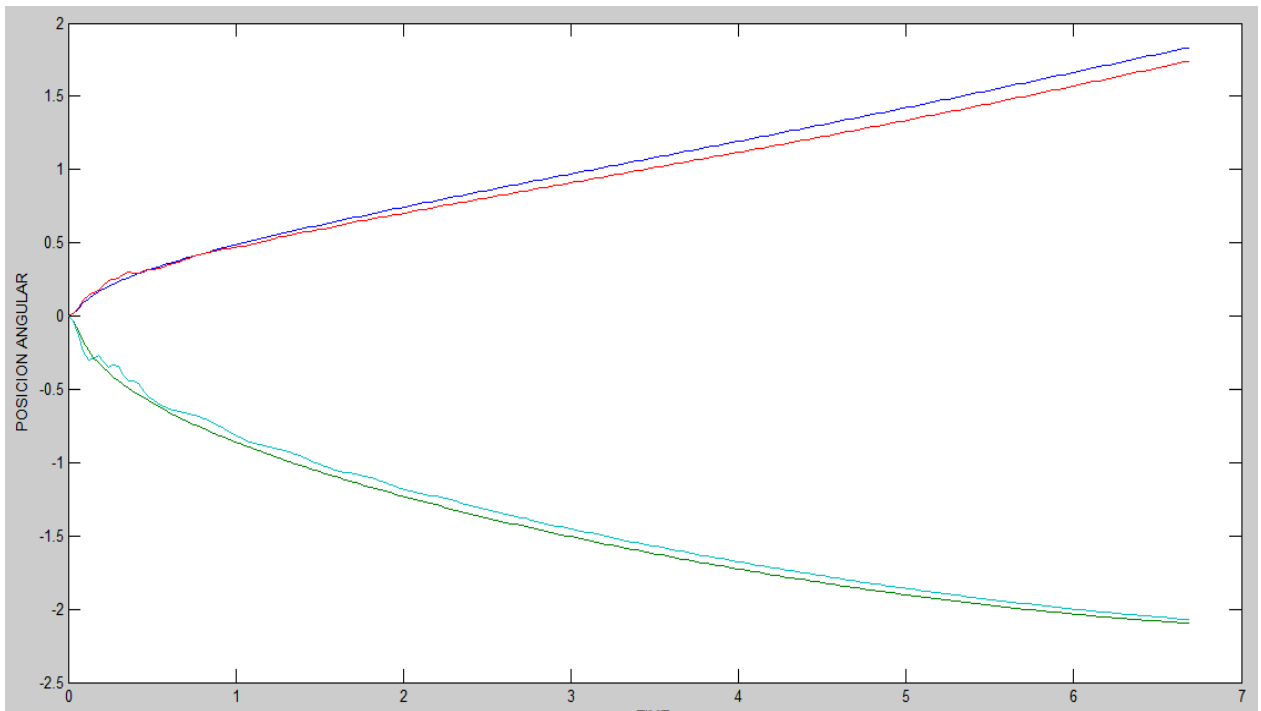


Figura 97 Posición angular actual vs deseada a 45° brazo doblado [1]

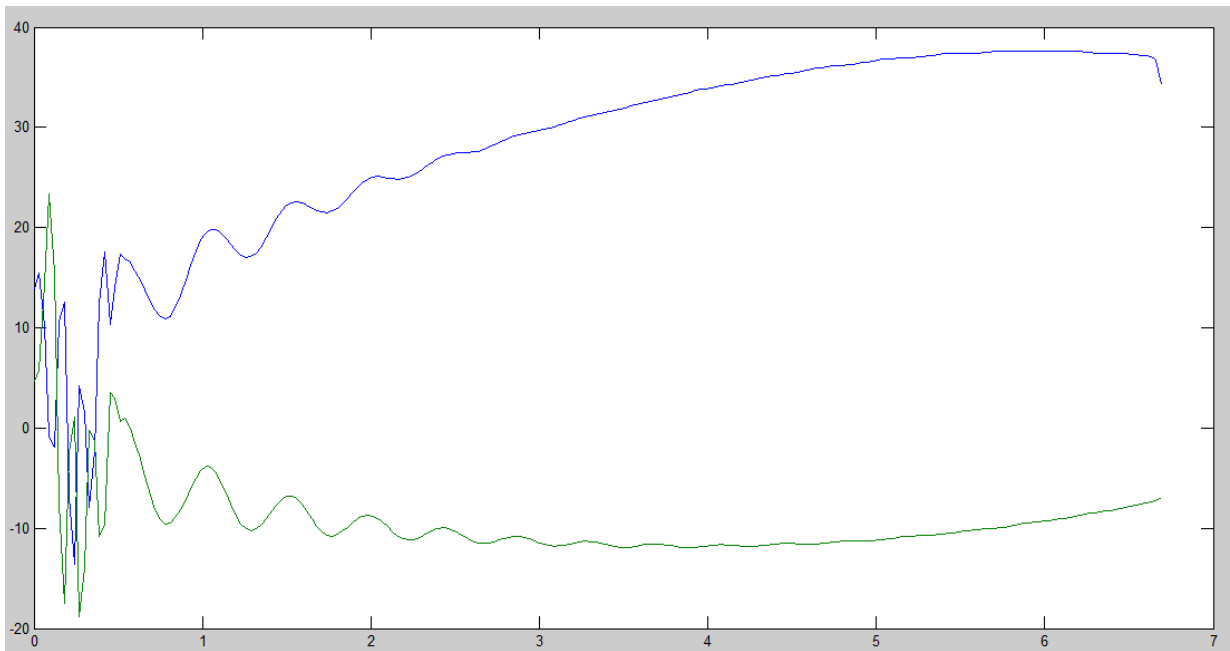


Figura 98 Señal de control a 45° brazo doblado [1]

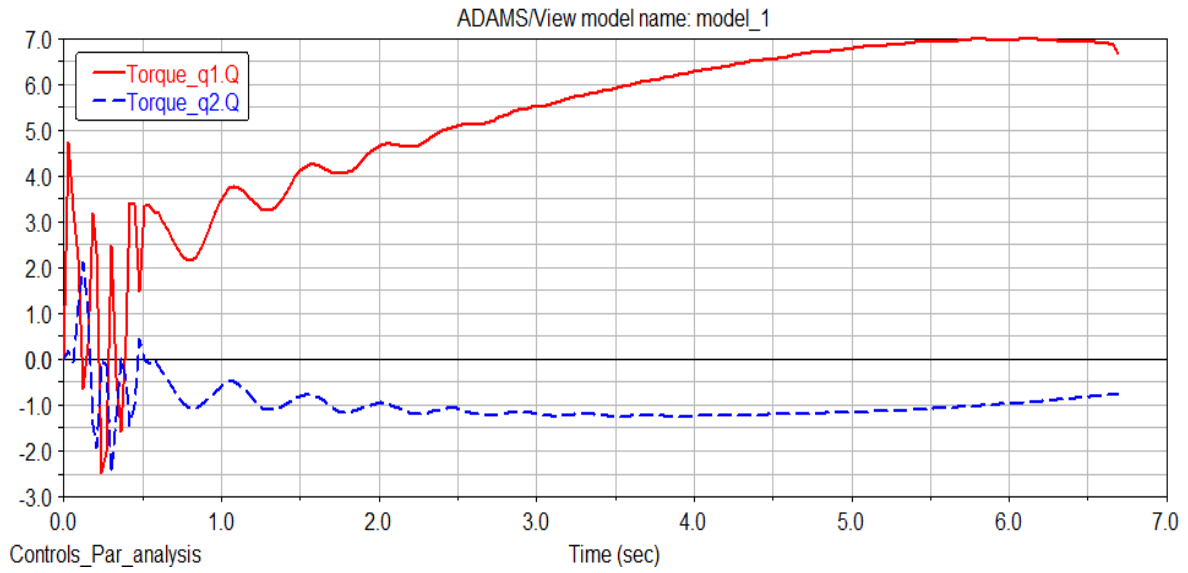


Figura 99 Par generado a 45° brazo doblado [1]

Para una posición de 90° con el brazo doblado

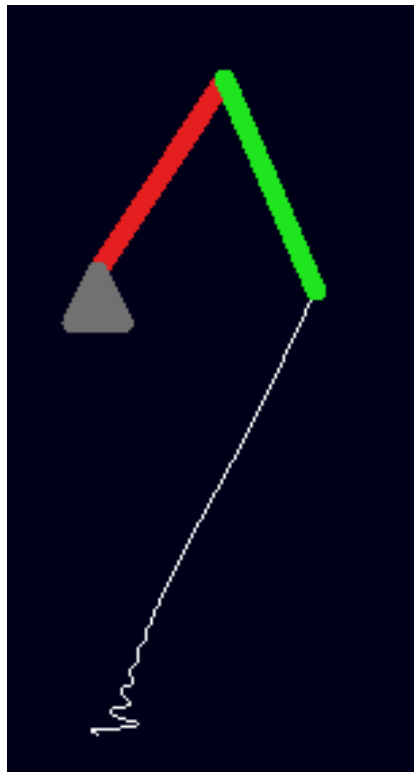


Figura 100 Manipulador a 90° no recto [1]



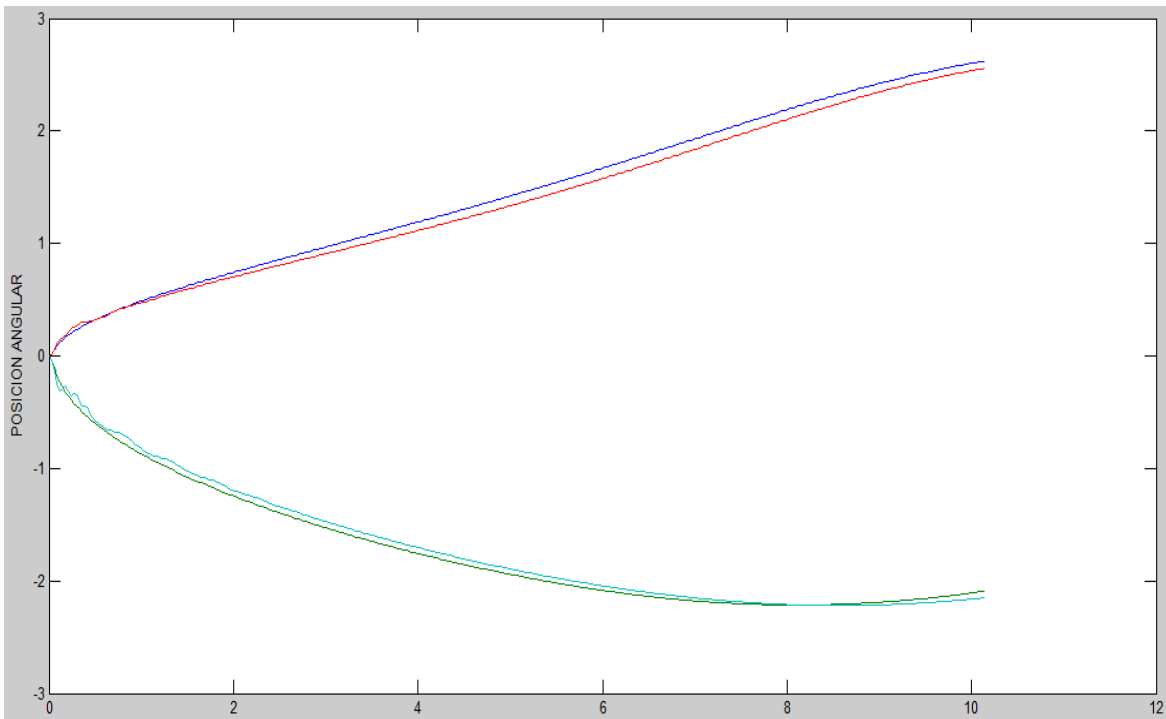


Figura 101 Posición angular actual vs deseada a 90° brazo doblado [1]

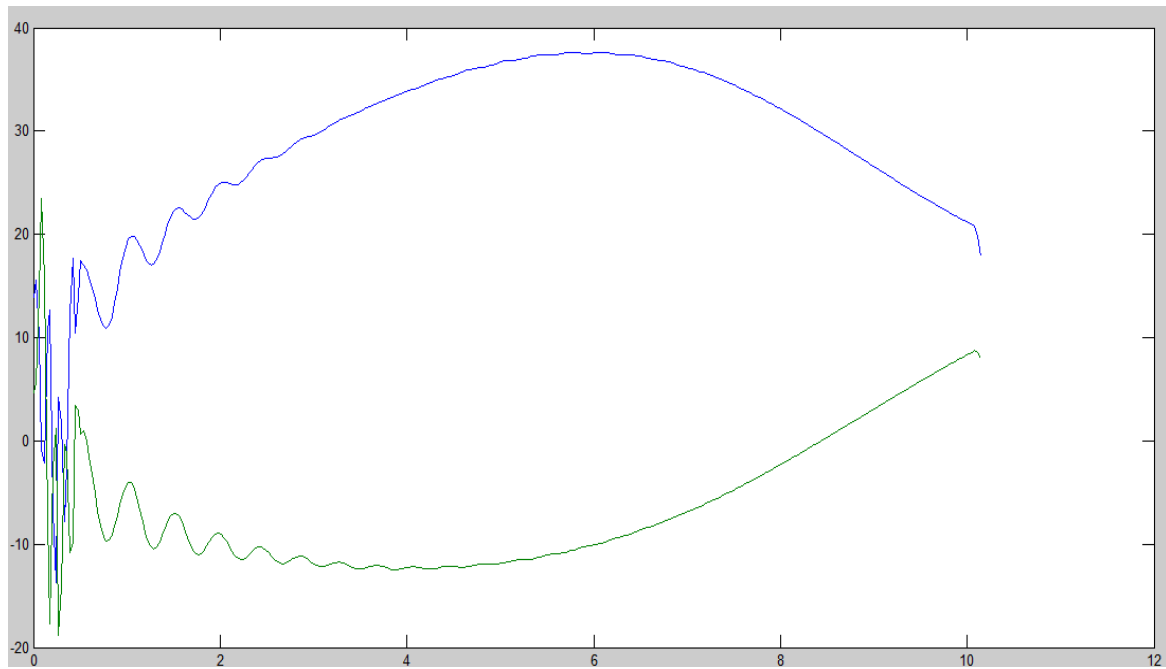


Figura 102 Señal de control a 90° brazo doblado [1]

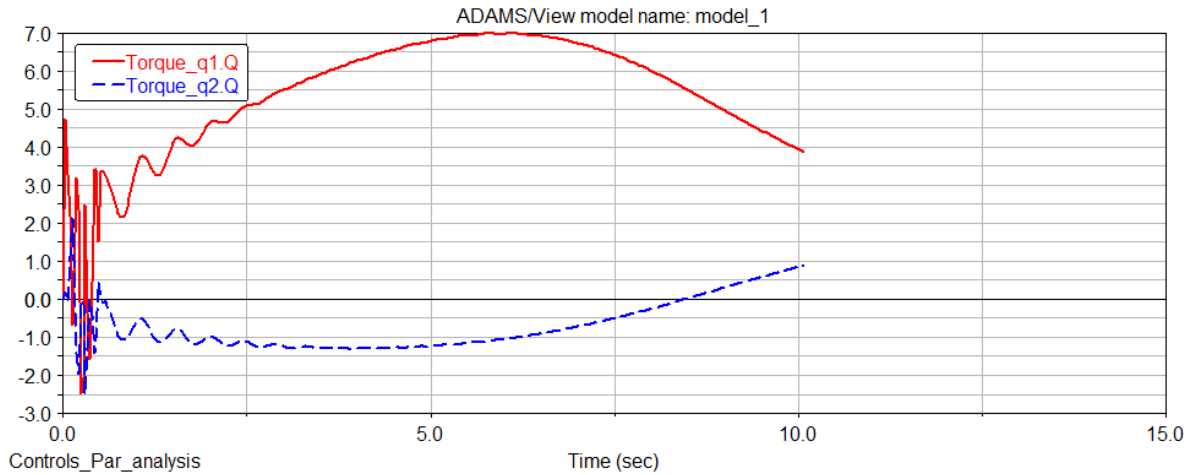


Figura 103 Par generado a 90° brazo doblado [1]

Como se observa en las anteriores gráficas el comportamiento del sistema se mantiene equivalente para cualquier posición. En ocasiones es necesario establecer parámetros de movimientos del manipulador en sectores del plano para evitar movimientos aleatorios, donde no sigue la trayectoria dada desde un comienzo pero aun así, se conserva el comportamiento esperado de los motores logrando su posición final.

## 6. Conclusiones y trabajos futuros

Para lograr un correcto funcionamiento y obtener la información correcta, es necesario tener conocimiento exacto del modelo del robot tanto del modelo dinámico y del modelo cinemático.

Las matrices de sintonización se encuentran de manera empírica debido a que no existe una metodología establecida para su cálculo, observando que el error en el estado estacionario no es cero, debido a que se implementa solo un control PD, este puede ser reducido si se usa un control con parte integral.

El desarrollo de parámetros o limitaciones de movimiento en el espacio a mover es necesario implementarlo cuando se desee fabricar físicamente el manipulador se evite movimientos no deseados y rupturas en la estructura.

Teniendo en cuenta las características del sistema de control desarrollado, basado en un modelo virtual construido en el software de simulación MSC Adams y modelos matemáticos generales, se observa que este puede ser aplicado a cualquier manipulador logrando una correcta evaluación y simulación antes de ser manufacturado, dando un amplio campo para mejoras de sus capacidades y tareas.

Se logró un correcto enlace entre los software de simulación MSC Adams y Matlab satisfaciendo los requerimientos planteados, suministrando información que se ajustan a condiciones reales de un robot manipulador de dos grados de libertad y así comportándose satisfactoriamente a lo esperado teóricamente.

Se comprobó la robustez del sistema implantando una perturbación aleatoria, comprobando que el controlador soporta y mantiene la estabilidad del sistema.

Para trabajos futuros se propone,

Implementar experimentalmente técnicas de control presentadas en este documento, generando un reporte en cuanto a estabilidad del sistema y la reducción del error empleando las mismas condiciones de simulación.

Desarrollar nuevos prototipos virtuales de manipuladores empleando estos controladores y técnicas de control alternativas realizando comparación entre ellas.

Montar físicamente un manipulador y aplicarle diferentes tipos de controladores comprobando la información obtenida con la simulada.

## 7. Bibliografía

1. Mendoza Poveda, Carlos Iván, modelado, simulación y control de un manipulador robótico de 2 grados de libertad empleando Adams/matlab, Universidad Pontificia Bolivariana. 2012
2. Martínez Josué R., Guerrero Gerardo V., Vela Luis G., Control de seguimiento de robots manipuladores, un estudio comparativo, Centro Nacional de Investigación y Desarrollo Tecnológico Cuernavaca, México.
3. Raúl Ñeco Caberta, Néstor Ramírez Morales y Manuel de Jesús Nandayapa Alfaro, Caracterización matemática para la obtención del modelo dinámico del robot Scrobot-ER V Plus, Departamento de Ingeniería Industrial y Manufactura, Universidad Autónoma de Ciudad Juárez.
4. Ángel Silva Luis, Control visual de robots paralelos. análisis, desarrollo y aplicación a la plataforma robotenis, Escuela Técnica Superior de Ingenieros Industriales 2005.
5. Kelly, Rafael y Santibáñez, Control de movimiento de robots manipuladores, Prentice Hall
6. Join Motion, Practica de Robótica utilizando matlab.
7. John J. Craig, Robótica, tercera edición, Prentice Hall.
8. Especificaciones motor 1  
[http://www.specamotor.com/en/Kollmorgen/motors/B-204-A/datasheet\\_motor.html](http://www.specamotor.com/en/Kollmorgen/motors/B-204-A/datasheet_motor.html)[1]
9. Especificaciones motor 2  
[http://www.specamotor.com/en/Kollmorgen/motors/B%28E%29-104-A/datasheet\\_motor.html](http://www.specamotor.com/en/Kollmorgen/motors/B%28E%29-104-A/datasheet_motor.html)[2]