

DESARROLLO DE ESTRATEGIA DE CONTROL PARA UN BRAZO MECÁNICO REPARADOR DE TUBERÍAS HIDROSANITARIAS

JULIÁN DAVID PAREJA GARZÓN



Trabajo de grado para optar por el título de Magister en Ingeniería Electrónica

Directores:

ING. FREDY ORLANDO RUIZ PALACIOS Ph.D.

ING. DIEGO ALEJANDRO PATIÑO GUEVARA Ph.D.

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

INGENIERÍA ELECTRÓNICA

BOGOTÁ, Mayo de 2015

AGRADECIMIENTOS

Deseo expresar mis sinceros agradecimientos a:

Mi familia, por haber estado en todo este proceso de formación, arduo y de mucho esmero. Por haberse convertido en ese soporte único que me ha llevado a alcanzar tantas metas. Gracias a mi Papá y a mi Mamá por hacerme lo que soy hoy, así como a mis hermanos que sin su apoyo no lo hubiera logrado.

- Mis directores, ingenieros Fredy Orlando Ruiz Palacios y Diego Alejandro Patiño Guevara, por brindarme la oportunidad de crecer más no solo en el campo profesional, sino en mi vida personal con sus grandes consejos y experiencia.
- A mi novia María Paula Domínguez, por haber vivido junto a mí todos los momentos complicados de este proceso y por haber sido una fuente motivación grande.
- La Pontificia Universidad Javeriana por permitir alcanzar el título de Magister en Ingeniería Electrónica, al considerar este desarrollo como un proyecto acreditador para el título.

TABLA DE CONTENIDO

Lista de Tablas.....	i
Lista de Figuras.....	ii
1. CAPITULO 1. INTRODUCCIÓN.....	1
1.1 Características de desempeño.....	1
1.2 Características del brazo robótico.....	2
1.3 Requerimientos Brazo robótico.....	3
1.4 Estado del arte	4
2. CAPITULO 2. MODELAMIENTO DEL SISTEMA.....	6
2.1 Concepción modelo CAD.....	6
2.2 Herramienta SIMMECANICHS.....	9
3. CAPITULO 3. ANALISIS DE EQUILIBRIO ESTÁTICO Y DISEÑO DE TRAYECTORIA.....	13
3.1 Análisis Dinámico del robot y selección de trayectoria.....	13
3.2 Cálculo de Ganancias en cada punto de trayectoria.....	14
4. CAPITULO 4. CONTROL LPV (LINEAR PARAMETER VARYING) MEDIANTE (GAINS SCHEDULING).....	21
4.1 Interpolación de valores de Ganancia y Torques.....	22
4.2 Desempeño controlador.....	26
5. CAPITULO 5. SELECCIÓN INSTRUMENTACIÓN.....	31
5.1 Selección motores	31
5.2 Selección Sensores y efector final.....	33
6 CONCLUSIONES.....	36
7 BIBLIOGRAFÍA.....	37

LISTA DE TABLAS

TABLA 1. <i>Tabla de Pesos y dimensiones</i>	8
TABLA 2. <i>Tabla de Valores de desempeño Control LPV.....</i>	29
TABLA 3. <i>Tabla comparativa de motores.....</i>	32

LISTA DE FIGURAS

FIGURA 1. Rango de trabajo Brazo antropomorfo.....	3
FIGURA 2. . Brazo robótico hecho en toolbox robot MATLAB.....	5
FIGURA 3. Base robot	6
FIGURA 4. Primera articulación.....	7
FIGURA 5. Segunda articulación.....	7
FIGURA 6. Diseño Solidworks Brazo robótico	8
FIGURA 7. Robot con plataforma de exploración	9
FIGURA 8. Modelo mecánico en simulink	10
FIGURA 9. Bloques simmecanichs	10
FIGURA 10. Brazo robótico simulado MATLAB.....	11
FIGURA 11. Posición primera articulación acción de gravedad	12
FIGURA 12. Selección trayectoria.....	13
FIGURA 13. Arquitectura general.....	14
FIGURA 14. Diagrama Control con acción integral.....	18
FIGURA 15. . Modelo Linealizado.....	19
FIGURA 16. Modelo simmecanichs	19
FIGURA 17. Respuesta paso segunda articulación modelo lineal	20
FIGURA 18. Respuesta paso segunda articulación modelo no lineal.....	20
FIGURA 19. Arquitectura general Control LPV	21
FIGURA 20. Descripción movimiento	22
FIGURA 21. Superficie primer valor de K por ubicación de polos	23
FIGURA 22. . Superficie primer valor de K por LQR	25
FIGURA 23. Arquitectura Control LPV con Gain scheduling	26
FIGURA 24. Respuesta movimiento articulación 1	27
FIGURA 25. Respuesta movimiento articulación 2	28
FIGURA 26. Respuesta controlador perturbación primera articulación	28
FIGURA 27. Respuesta controlador perturbación segunda articulación.....	29
FIGURA 28. Señal de control primera articulación	30
FIGURA 29. Motor STOGRA IP68	33
FIGURA 30. Codificador óptico Seleccionado	34
FIGURA 31. Cepillo copa de alambre para tuberías	35

1. INTRODUCCIÓN

Con el paso de los años, los brazos robóticos y los procesos automatizados hechos por los mismos han cobrado mucha importancia, tan es así, que ya no se puede pensar en hacer una tarea específica de alta complejidad sin utilizar un brazo robótico programable. Su uso es muy variado y en distintas áreas, tales como procesos de ensamble, soldado, pintado, mecanizado, etc. Se han vuelto herramientas tan útiles y tan indispensables que ya hacen parte de nuestro diario vivir.

Con el fin de hacer un aporte significativo en el mejoramiento de los robots y optimizar e innovar en el campo de reparación de tuberías hidrosanitarias se propone este proyecto que consiste en determinar un controlador adecuado para un brazo manipulador de tres grados de libertad, que se encargará de hacer la labor de lijado y adecuación de las paredes internas de tuberías hidrosanitarias averiadas, para su posterior reparación.

Este trabajo pretende proponer una idea en cuanto a la forma de hacer el lijado antes de comenzar una reparación en tuberías, diseñando un brazo robótico, calculando las fuerzas para mover el brazo, seleccionando instrumentación necesaria (sensores y actuadores) y finalmente una teoría de control que logre estabilizar el brazo robótico para poder recorrer una trayectoria planeada (Trayectoria estándar de lijado elegida según requerimientos), evaluada en simulación.

El lijado es una parte fundamental de la reparación tal como se especifica en el trabajo de investigación hecho previamente por el ingeniero Gabriel Guiza [1]. Sin el proceso de lijado, no sería posible adherir el material necesario para la reparación de grietas y fisuras, y aunque en Colombia no hay una normativa completa del proceso de reparación de tuberías, tan solo existe la norma NS-058 del acueducto de Bogotá, la idea es proponer un método seguro y efectivo para llevar a cabo la tare de lijado [2].

Ceñidos a las especificaciones dadas por la empresa H&U (Hidráulica y Urbanismo), coejecutora del proyecto de investigación con colciencias que se encuentra asociado a esta tesis (Proyecto YNAMI), y a las limitaciones de espacio dentro de la tubería a trabajar (28)", se ha decidido diseñar un brazo robótico acorde a esas limitaciones, con el fin de dar mayor comodidad al usuario final, así como determinar una rutina de reparación para hacer la tarea mucho más sencilla.

Como aporte final se hace la propuesta de la instrumentación necesaria que va en el brazo robótico, cumpliendo con estándares de protección (Certificación NEMA 6P) [22], y acordes a la geometría y funcionalidad del brazo. Dicha instrumentación también tiene sus criterios de selección, los cuales están basados en los cálculos de las fuerzas necesarias para mover el brazo, a su vez estos cálculos son hechos mediante MATLAB, el cual está enlazado al diseño mecánico por la herramienta SIMMECHANICS de SOLIDWORKS.

1.1 CARACTERÍSTICAS DE DESEMPEÑO

Con el fin de hacerse una idea más clara de los parámetros y estándares que debe cumplir el brazo robótico, a continuación se explicará una serie de características y requerimientos, los cuales están basados en modelos ya existentes en el mercado.

1.2 CARACTERÍSTICAS DEL BRAZO ROBÓTICO

En el mercado mundial existen muchos tipos de brazos robóticos y empresas especializadas en producirlos en masa, sin embargo, hay algunos modelos que se toman como base para seleccionar las características más representativas, y así poder diseñar un brazo acorde a las necesidades de la tarea a realizar. Algunos de estos modelos consultados son:

- **Mitsubishi** RV-2f
- **ABB** IRB120
- **KUKA** KR AGILUS sixx WP
- **KUKA** KR 6 R700 FIVVE (KR AGILUS)

Todos estos robots tienen un punto en común, y es que son robots industriales para tareas de media carga, sin embargo el único que posee especificaciones IP de protección es el KUKA AGILUS SIXXWP. Las dimensiones y pesos de cada robot están en el orden de 4Kg por articulación y soportan cargas máximas de hasta 6Kg.

Otro aspecto muy importante es que cada robot tiene su centro de mando y una polarización de 200 a 600V AC debido a las tareas de gran demanda y carga que deben hacer. Como el brazo a diseñar no hará tareas tan complejas ni de tanta demanda, la alimentación eléctrica y hardware para controlar los motores será un factor determinante a la hora de seleccionar la instrumentación adecuada [3] [4] [5].

1.3 REQUERIMIENTOS BRAZO ROBÓTICO

Es claro que el espacio en el que trabajará el brazo robótico es reducido, y por lo tanto se necesita una topología especial que le permita moverse con libertad dentro de la tubería, es por eso que se decide hacer un brazo antropomorfo [6], ya que a diferencia de los brazos cilíndricos, SCARA y esféricos, puede llegar a varias partes de la tubería y moverse con mayor libertad gracias a su morfología, como se ve en la figura 1.

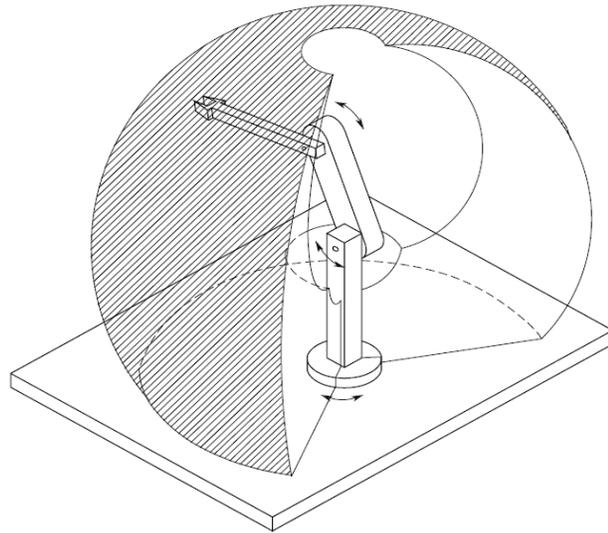


Figura1. Rango de trabajo Brazo antropomorfo. Tomado de: BRUNO SICILIANO, *Robotics modelling, planning and control*, p.7

Otro aspecto importante para el diseño del brazo robótico es cumplir con las normas de protección establecidas, y a petición del cliente la empresa H&U (Hidráulica y Urbanismo) se requiere que el robot tenga certificación NEMA 6P o IP68, por este motivo el material con el que se simulará el modelo será el Aluminio 6061-T6, sugerido por la Universidad Militar Nueva Granada, universidad que participa en el diseño mecánico de la plataforma robótica.

Finalmente, se necesita que los movimientos dentro de la tubería sean suaves y precisos, con lo cual se necesitará de motores capaces de proporcionar esa suavidad y precisión, así como la selección de codificadores ópticos y sensores necesarios para facilitar la tarea al operario que manipule el brazo.

1.4 ESTADO DEL ARTE

En el campo de robótica se pueden encontrar una gran cantidad de controles para brazos robóticos, como controles difusos, de impedancia, H^∞ , y PID [7] [8] [9]. Sin embargo, lo que se busca en este proyecto es seleccionar una estrategia de control acorde a la tarea a realizar, es por eso que de los controles anteriormente presentados se extrae en particular que algunas de esas estrategias utilizan una técnica llamada (Gain Scheduling), la cual permite hacer un uso recursivo de varios valores en los controladores, para poder tener un control dinámico de los brazos robóticos. [10]

Como otra base de soporte para una aproximación acertada al diseño mecánico y selección de instrumentación, se consultaron varios fabricantes de robots y modelos acordes a las dimensiones aproximadas. Es así como se logra tener una inclinación por los modelos fabricados por la empresa KUKA, ya que estos son los únicos que ofrecen en su portafolio robots con sellos IP67 [11]. Por otra parte, estos modelos de KUKA permitieron hacer una selección más acertada de materiales de construcción para evitar el deterioro temprano y posibles averías que pueda causar el estar expuesto a un medio altamente corrosivo.

Para hacer un modelamiento real de un brazo robótico también se hizo el análisis físico del robot SCORBOT [12], el cual se pidió al laboratorio de electrónica de la Pontificia Universidad Javeriana. Al trabajarlo físicamente, desarmarlo y analizar pieza por pieza, se tuvieron las bases principales para seleccionar los grados de libertad y los movimientos para trazar una trayectoria. Una vez obtenidos estos parámetros, se creó un robot propio en el toolbox robot de MATLAB hecho por Peter Corke ver figura 2, teniendo en cuenta parámetros como matiz D-H, dinámicas y ecuaciones básicas de movimiento [13].

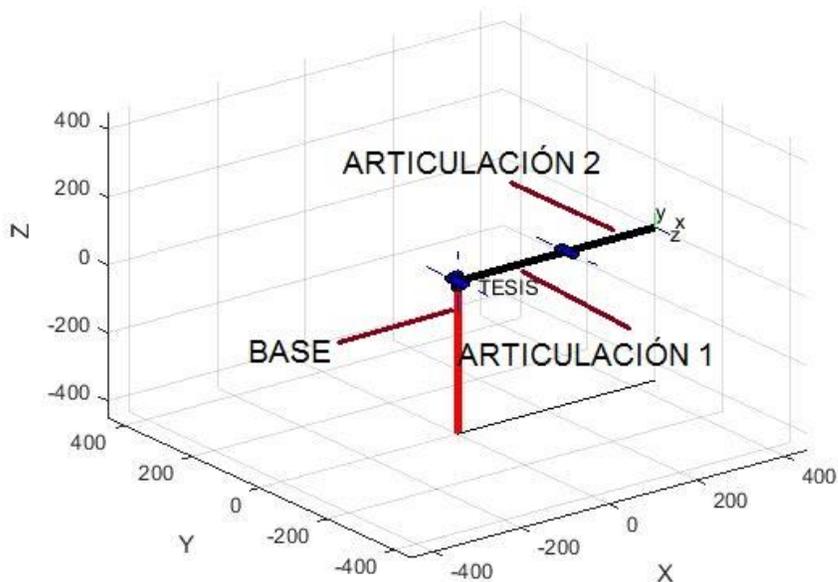


Figura2. Brazo robótico hecho en toolbox robot MATLAB

También se usa la librería de peter corke Para seleccionar las posibles trayectorias a implementar para realizar la tarea de lijado, así como tener una base teórica de las ecuaciones y variables que rigen el movimiento de una articulación.

Finalmente con toda esta información reunida se decide hacer un control LPV (Linear Parametric Variant) basado en (Gain Scheduling) [14], con el fin de suavizar el movimiento y dar robustez al brazo robótico, sabiendo que la tarea a realizar exige precisión y fuerza.

CAPITULO 2 MODELAMIENTO DEL SISTEMA

Para comenzar, es necesario tener un modelo previamente establecido, por lo cual se decide diseñar un brazo robótico en el programa SOLIDWORKS, con el fin de pasar el modelo (CAD) al programa MATLAB y allí poder ser tratado para luego ser controlado.

2.1 CONCEPCIÓN MODELO CAD

El modelo a realizar tiene tres grados de libertad, los cuales están dispuestos así:

Base: Rotacional eje z (ver figura 3)

Primera articulación: Pivota sobre el punto de rotación de la Base (ver figura 4)

Segunda articulación: Pivota sobre el punto de rotación de la primera articulación (ver figura 5)

Teniendo ya los grados de libertad y el tipo de movimiento que se debe hacer, se procede a diseñar cada pieza en SOLIDWORKS, y se comienza por la base, esta debe rotar sobre el eje z y así mismo debe poder soportar el peso de las dos piezas siguientes, además de tener el punto central de pivoteo de la primera articulación, se diseña como se ve en la figura 3.

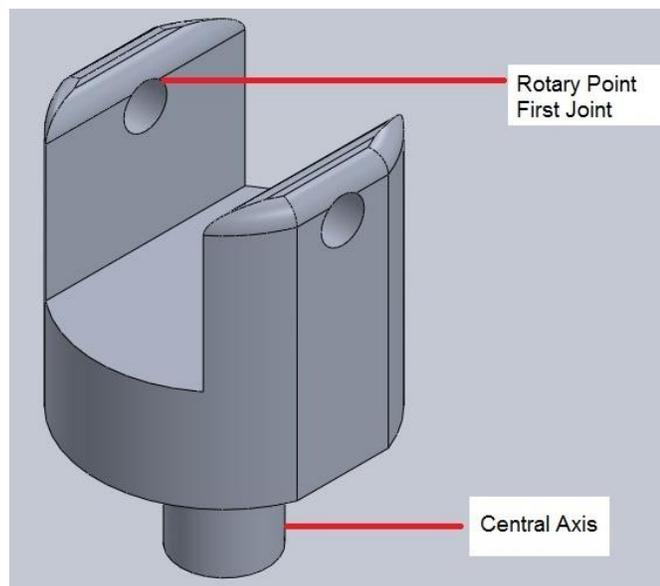


Figura3. Base robot

La primera articulación se diseña con puntos que encajan en la base para poder hacer una relación de posición rotacional, también se tiene un punto de rotación superior para conectar la siguiente articulación, como se ve en la figura 4.

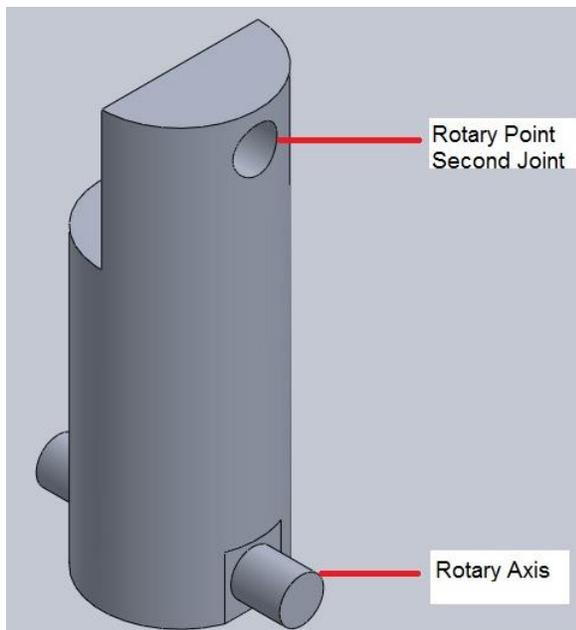


Figura 4. Primera articulación

Finalmente la segunda articulación debe soportar el efector final y debe tener un eje de rotación que encaje con el punto de rotación de la primera articulación como se ve en la figura 6. la parte final de la articulación 2 simula el efector final (Piedra tipo esmeril, lija, etc.).

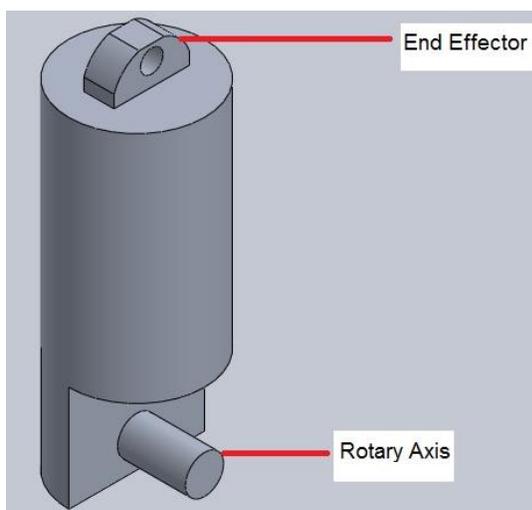


Figura 5. Segunda articulación

Luego de hacer las piezas completas, se configura el material sugerido por la Universidad Militar Nueva Granda, universidad encargada del diseño mecánico del proyecto YNAMI, el material de la estructura es (Aluminio 6061-T6) y se seleccionó este material para soportar el medio altamente corrosivo al que será expuesta la estructura.

Finalmente se ensamblan las piezas anteriormente descritas dando como resultado la estructura de la figura 6.

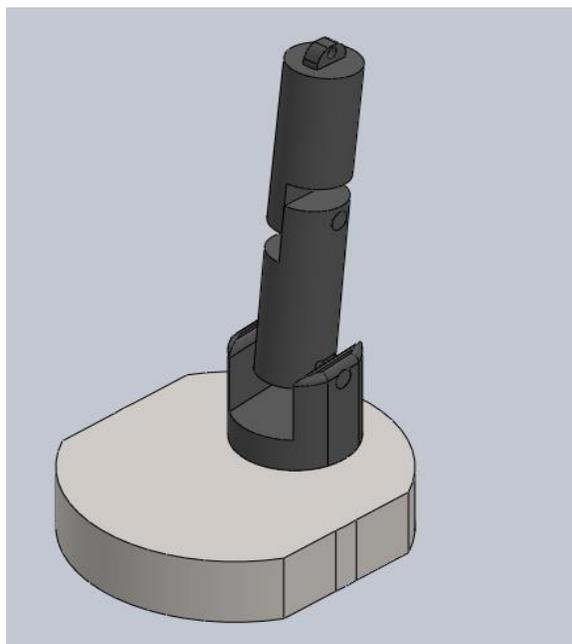


Figura6. Diseño Solidworks Brazo robótico

Las especificaciones tanto de peso como de dimensiones de cada parte de la estructura pueden ser observadas en la tabla 1.

PIEZA	MASA (Kg)	DIMENSIONES (Largo X Diámetro) (cm)
Base	4.14	(13 X 15)
Articulación 1	3.6	(22 X 9)
Articulación 2	3.3	(22 X 9)

Tabla 1. Pesos y dimensiones

Por otra parte el control no puede ser llevado a cabo por el programa de SOLIDWORKS, ya que éste es un programa especializado más en concepción de modelos mecánicos y no de su total análisis matemático, es decir, este programa no posee las mismas herramientas de análisis que ofrece MATLAB con SIMULINK para implementar un control como el que se tiene planeado hacer.

Como la empresa H&U no especifica un diámetro de tuberías en las que se va a entrar a hacer reparaciones, se elige un diámetro comprendido entre las 30" y las 60". Con el fin de no tener problemas a la hora de ingresar a la tubería y a la hora de hacer la reparación, se propone que la altura máxima del brazo sea de 44 cm, así se puede cubrir gran parte de los diámetros y se puede usar una pequeña plataforma de elevación cuando la grieta se encuentre en una altura superior, como se observa en la figura 7.

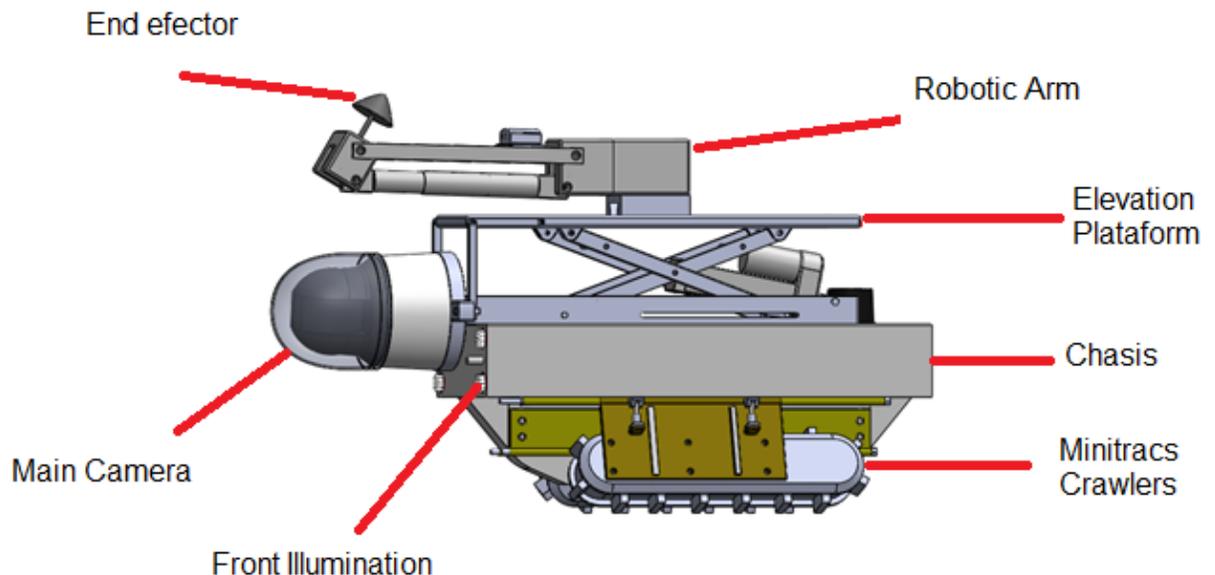


Figura7. Robot con plataforma de exploración (Imagen hecha en SOLIDWORKS por Universidad militar).

2.2 HERRAMIENTA SIMMECANICHS

Como se había dicho con anterioridad, el modelo presentado en la figura 2 en SOLIDWORKS solo puede ser analizado en todas sus características físicas de esfuerzos, resistencia, movilidad, peso etc. Es por eso que se hace necesario el uso de una herramienta más acertada para observar otras características dinámicas y estáticas del modelo. Para ello se utiliza la herramienta SIMMECANICHS, que es un toolbox en MATLAB, el cual se encarga de hacer el enlace total entre el diseño mecánico de SOLIDWORKS y la herramienta SIMULINK de MATLAB. Así pues una vez obtenido y concebido el modelo se pasa a MATLAB, en este entorno se tiene una visión tipo diagrama de bloques del modelo, tal como se ve en la figura 8.

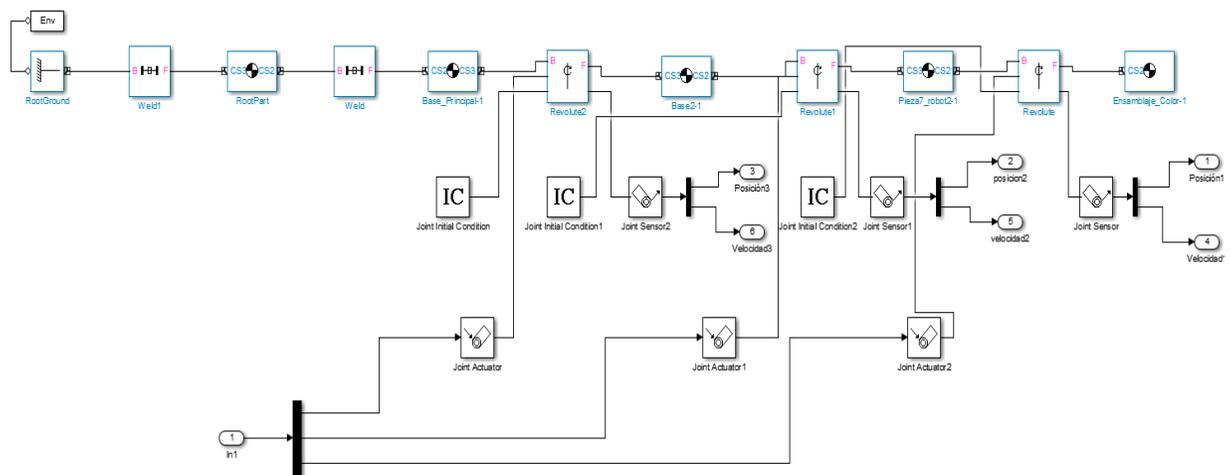


Figura8. Modelo mecánico en simulink

Como se puede ver en la figura 9 el modelo mecánico está representado por bloques en simulink, donde un bloque representa la pieza, otro representa el tipo de movimiento de esa pieza, y otros representan los sensores y actuadores que se agregan al modelo para poderlo controlar, para una mejor resolución ver figura 9.

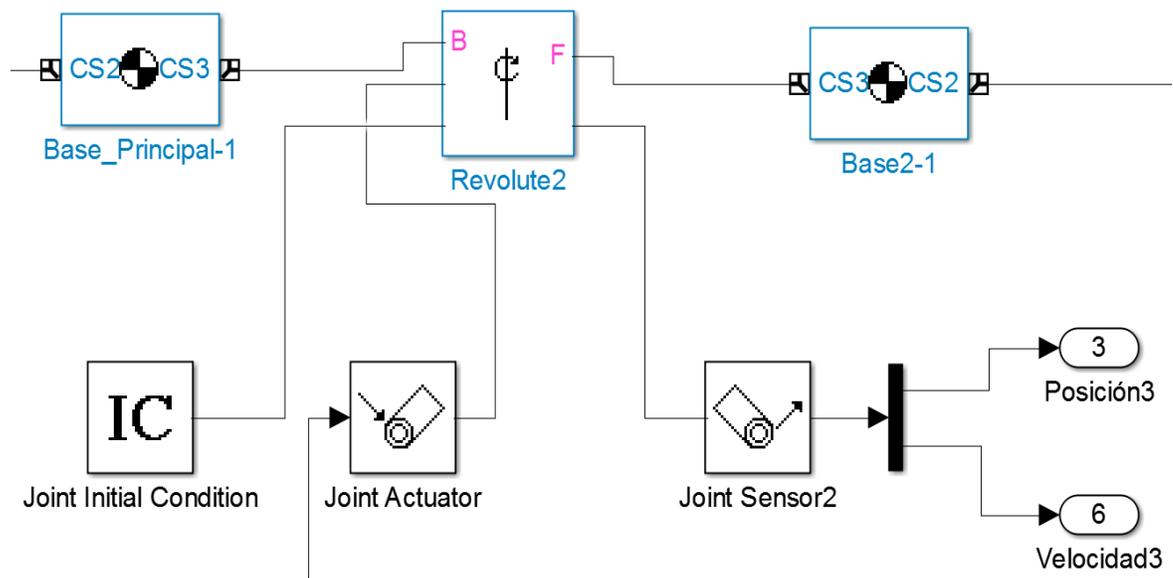


Figura 9. Bloques simmecanichs.

El bloque Base_Principal representa la Base del robot y el bloque Revolute2 representa el tipo de movimiento que hace la pieza, el bloque initial condition permite seleccionar la condición inicial de la Base_principal, Joint actuator representa el motor que actúa sobre la pieza, y finalmente el bloque Joint sensor representa el sensor que mide la salida en este caso posición y velocidad.

La figura 10 muestra cómo se visualiza el modelo simulado en MATLAB, se decide diferenciar cada pieza con un color diferente, así la base es de color gris, la primera articulación amarillo y la segunda articulación verde.

Cabe aclarar que para la adición de sensores y actuadores al modelo de SIMMECANICHS se utilizará la librería de primera generación, debido a que es más sencilla de entender y a que el modelo fue guardado inicialmente para ser trabajado con esa librería.

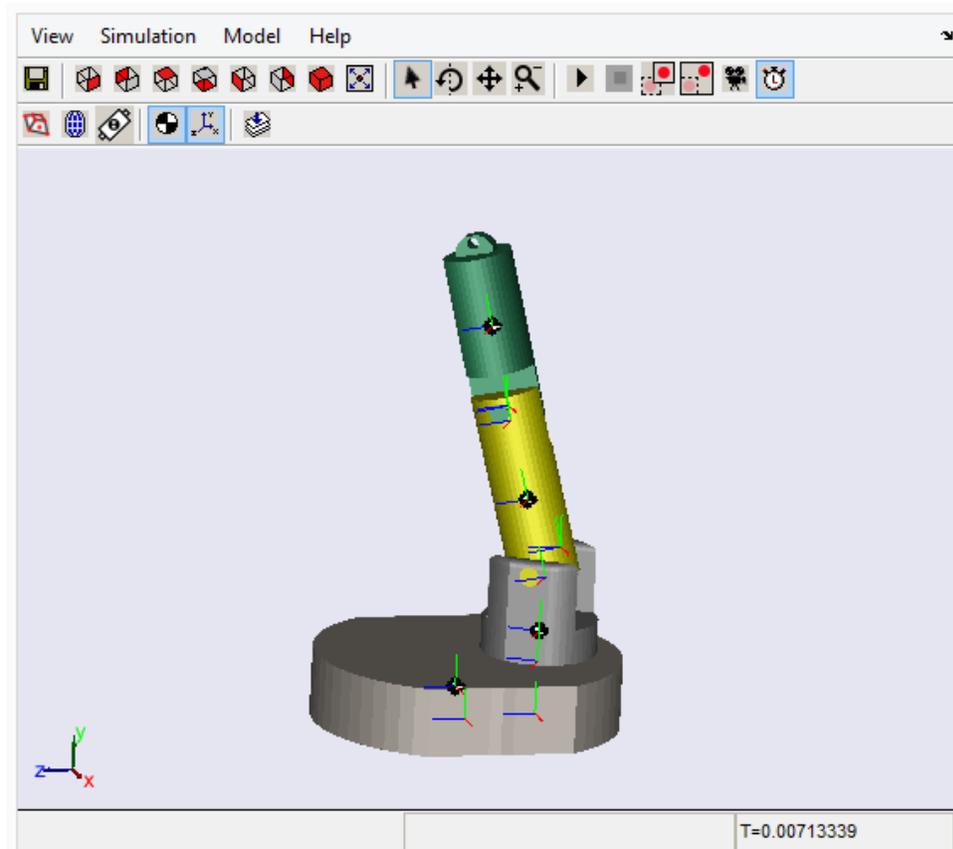


Figura 10. Brazo robótico simulado MATLAB

Al simular el modelo en el entorno de MATLAB sin ninguna acción de control, y con condiciones iniciales en $((0, 0, 0)$ posición vertical) se aprecia como cae el modelo por acción de la gravedad visualizando la salida de posición en grados de la primera articulación ver figura 11.

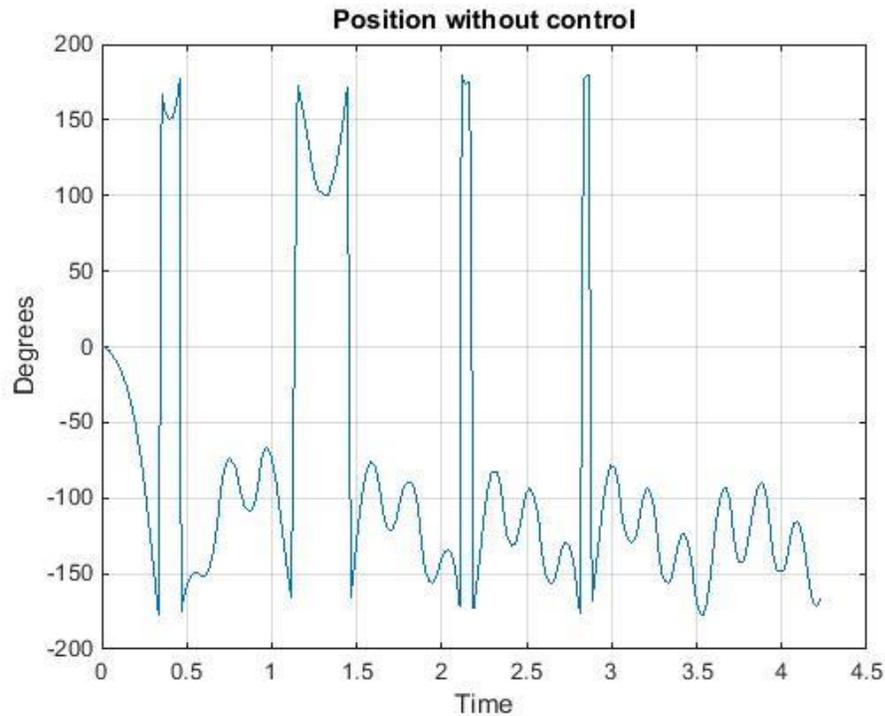


Figura 11. Posición primera articulación acción de gravedad

Analizando la figura 11, el modelo oscila entre valores de posición determinados por la solución de ecuaciones diferenciales que lo describen, es decir, los saltos de posición que se observan en la gráfica son la aproximación del movimiento descrito por las articulaciones ya que no cuenta con ningún tipo de control o fuerza que lo mantenga en una posición fija y a que está gráfica es la salida de uno de los sensores del toolbox SIMMECANICHS. Hay que tener en cuenta que la figura 11 solo corresponde a la posición de la primera articulación, sin embargo se aprecia que oscila rápidamente entre ángulos que van de 180° a -180° evidenciando una ausencia de control.

CAPITULO 3: ANÁLISIS DE EQUILIBRIO ESTÁTICO Y DISEÑO DE TRAYECTORIAS

Para poder implementar el control adecuado para el movimiento del brazo robótico, es indispensable saber qué parámetros se deben controlar, así como que limitaciones se tienen en cuanto a la estructura mecánica y tiempos de respuesta. Para esto, se decide diseñar una trayectoria que se ajusta a lo que se quiere (lijar), y se hace un análisis detallado del modelo en cada punto de dicha trayectoria.

3.1 ANALISIS DINÁMICAS DEL ROBOT Y SELECCIÓN DE TRAYECTORIA

Con el modelo en SIMMECANICHS, se procede a hacer los respectivos análisis de fuerzas y posiciones. Primero que todo se ponen actuadores (Motores) en cada una de las articulaciones del robot, luego se ponen sensores para medir su posición o fuerza de salida y finalmente se pone un bloque de condición inicial para escoger la posición inicial de la pieza cuando inicie la simulación, ver figura 10.

Simulando el modelo, este se mueve dependiendo de la solución de las ecuaciones diferenciales que lo describen y las condiciones que inciden sobre él (Gravedad), ya que no posee ninguna fuerza que lo controle. Para plantear una acción de control acorde al modelo, lo primero es planear una trayectoria adecuada para obtener una idea general de cómo hacer un lijado efectivo, para esto se simula una grieta en una tubería en SOLIDWORKS y sobre esa grieta se planea la trayectoria, ver figura 12.

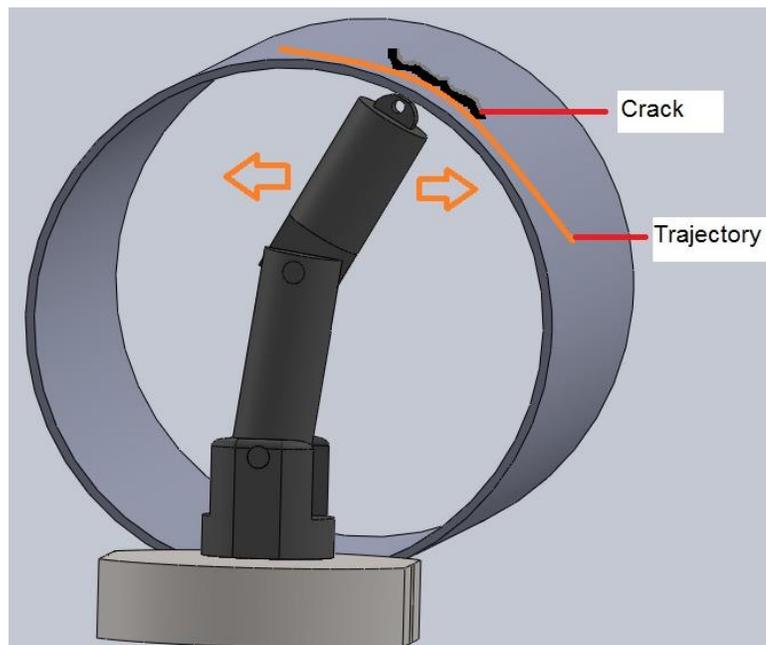


Figura 12. Selección trayectoria (Figura hecha en SOLIDWORKS para fin demostrativo)

Teniendo en cuenta que no hay una normativa específica que regule cómo hacer el lijado de tuberías ni que defina unos requerimientos mínimos, se define que el brazo robótico haga movimientos suaves y precisos de lijado, para este caso se necesita que la trayectoria que recorre el brazo describa la geometría de una grieta se encuentra alrededor de la tubería, como la mostrada en la figura 12, ya que por su geometría implica un grado de complejidad mayor describir un movimiento de esas características y hacer un correcto lijado.

En este caso particular se definen las posiciones: Inicial (Base = 0° , Articulación 1 = -20° , Articulación 2 = 15°). Final (Base = 0° , Articulación 1 = 20° , Articulación 2 = 0°), el paso de 15° a 0° en la segunda articulación, se hace para tener un mejor trazo de la circunferencia dentro de la tubería, en la figura 12 se muestra como debe ir cambiando a medida que se va moviendo la primera articulación.

3.2 CÁLCULO DE GANANCIAS EN CADA PUNTO DE TRAYECTORIA

Simulando el modelo en malla abierta se pudo comprobar que es inestable ver figura 11, por lo tanto necesita ser controlado para seguir una trayectoria de referencia, es decir, el modelo debe ser realimentado con un control que le permita seguir esa trayectoria como se ve en la arquitectura de la figura 13.

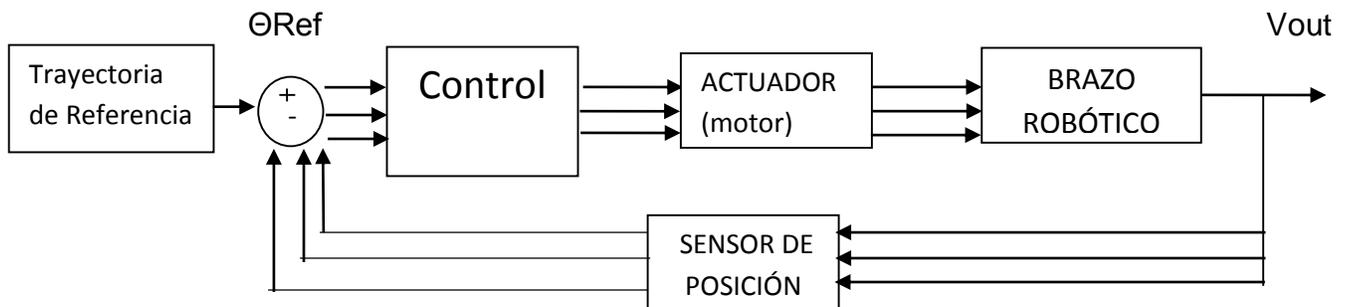


Figura 13. Arquitectura general

Para hacer el cálculo de ganancias de dicho controlador, primero que todo se debe saber cuáles variables se van a controlar y cuáles se van a medir, para esto se describe el modelo no lineal partiendo de sus entradas y salidas, en primer lugar como el brazo tiene motores, estos motores en SIMMECANICHS pueden actuar de dos maneras, una mediante una posición, velocidad y aceleración, y otra mediante un torque que se aplica dependiendo de una referencia, para este caso se usará la

segunda opción, ya que se busca controlar la posición del robot mediante una fuerza aplicada. Así pues para este modelo se tienen 3 entradas de posición (referencia), que describen la trayectoria a trazar. Las salidas del modelo son las posiciones de cada articulación y las velocidades de las mismas, teniendo así 6 estados (3 posiciones, 3 velocidades).

Como el robot es un sistema no lineal, se procede a linealizar en los puntos de operación (cada punto de la Trayectoria), con el fin de obtener las matrices A, B, C, D que describen el modelo, para esto se usa la función Linmod de MATLAB, Dicha función es capaz de obtener las matrices A, B, C, D alrededor de un punto de operación seleccionado para su posterior análisis utilizando la función de MATLAB Linmod.

$$\dot{X} = F(X, U) \quad (1)$$

Con X (Vector de variables de estado y U (Vector variables de entrada), luego se linealiza.

$$\dot{X} = \frac{\partial F}{\partial X} X + \frac{\partial F}{\partial U} U \quad (2)$$

El algoritmo predeterminado de linmod utiliza el jacobiano para linealizar, ya que es un método pre programado en muchos bloques de SIMULINK, es decir, las matrices jacobianas son obtenidas numéricamente por la función linmod. Haciendo la linealización en la posición (0,0,0) se obtienen las matrices:

```

A =
      0      0      0      1.0000      0      0
      0      0      0      0      1.0000      0
      0      0      0      0      0      1.0000
  115.4448 -32.1429 -0.0000      0      0      0
   24.6873  16.6718 -0.0000      0      0      0
    0.1004 -0.0444 -0.0000      0      0      0

>> B

B =
      0      0      0
      0      0      0
      0      0      0
   0.0384  14.1031  47.3094
   0.0095   7.0022  14.1031
   5.9747   0.0095   0.0384

>> C

C =
   57.2958      0      0      0      0      0
      0   57.2958      0      0      0      0
      0      0   57.2958      0      0      0
      0      0      0   57.2958      0      0
      0      0      0      0   57.2958      0
      0      0      0      0      0   57.2958

D =

      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0

```

Donde A es la matriz de estados, como son 6 estados la matriz A posee dimensiones $n \times n$ con ($n = \text{Número de variables de estado}$) para este sistema 6×6 . La matriz B representa las 3 entradas relacionadas a los torques, es decir, de dimensiones ($n \times p$) donde ($p = \text{Número de entradas al sistema}$) luego $\text{dim}(B) = 6 \times 3$. La matriz C representa las salidas del modelo, para q salidas se tiene una matriz C de dimensiones ($q \times n$) ó sea 6×6 . Finalmente la matriz de ceros D de dimensiones ($q \times p$) 6×3 .

Se linealiza el modelo cada 10° partiendo desde -50° hasta llegar a 50° para las variables theta 1 y theta 2 (Posiciones de la primera y segunda articulación

respectivamente), para cubrir todo espacio de trabajo seleccionado, se plantea cada 10° ya que se pueden tener grietas que cubran espacios inferiores a los grados mencionados, y que requieren ser lijados con precisión dentro de la tubería.

Una vez obtenidas las matrices A, B, C, D del modelo en los puntos, lo que se hace es revisar su similitud para saber qué parámetros cambian, y si realmente el cambio en sus valores es considerable. Partiendo de esos valores se evalúan las matrices de controlabilidad y observabilidad de cada uno de los modelos obtenidos en los puntos de la trayectoria, revisando si son de rango completo o degenerado. Así se tiene un criterio más acertado para poder hacer un control por ubicación de polos, si la matriz de controlabilidad da un rango menor a 6 ($\text{Rango} < n$), no se podrá hacer un control por ubicación arbitraria de polos, ya que tendría un estado linealmente dependiente de otro y no sería controlable. En este caso $\text{Rango (Matriz controlabilidad)} = 6$, lo que quiere decir que tiene 6 estados linealmente independientes y es posible calcularle un control.

Como se tienen 3 entradas y 6 estados, se decide hacer un control por realimentación de estados, ya que con él se puede llevar cada variable a un valor deseado.

En primer lugar se sacan los valores propios de cada matriz A y se compara si realmente tienen una desviación considerable de polos que va más allá de una década, ya que de ser así no se podría hallar un solo controlador que estabilice el modelo en todos los puntos.

Una vez obtenidos todos los valores propios, se comparan y se saca el porcentaje de error máximo entre ellos el cual es del 22%, este error realmente no representa una variación significativa, por lo cual se asume un valor promedio de ellos.

```
ans =
    10.3258
   -10.3258
    -5.0491
     5.0491
    -0.0000
     0.0000
```

Se elige ubicar los polos del controlador más lejos del origen, pero estables, y con polos dominantes cerca al mismo para mejorar la respuesta del controlador, los valores son:

$$P = [-25.0000, -20.0000, -9.5000, -0.5000, -7.1000, -2.0000]$$

A continuación se presenta un diagrama de un control por realimentación de estados con acción integral incluida que es lo que realmente se implementará.

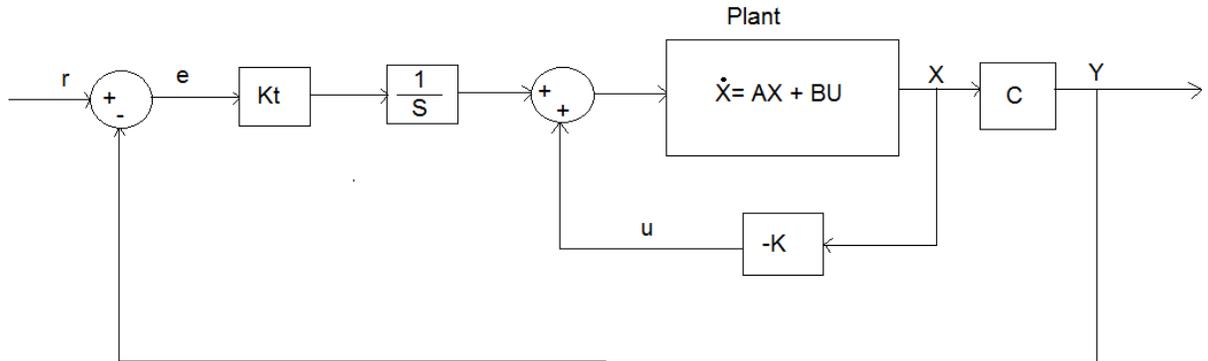


Figura 14. Diagrama Control con acción integral

$$Ae = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}, \quad Be = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad (3)$$

Para este caso las matrices se insertan en el código que se hace para poder hacer el cálculo del controlador de manera automática.

```
Cp =C(1:3,1:6);
Ae=[A zeros(6,3); -Cp zeros(3,3)];
Be=[B ; zeros(3,3)];
K=place(Ae,Be,[P'; -20;-40;-60]);
```

Se usa la función place para hallar el valor correspondiente a K y se usan los valores de -20,-40,-60 para la parte integral, lejos de los polos dominantes para no interferir con la acción de control principal. Una vez se hace el cálculo de K se obtiene el valor.

K =

```
-26.5482  -62.4132  168.0971  -0.6951  -1.9863  11.2022  4.8331  7.5820  -3.8951
131.8935  342.1489  -91.6392  1.8820  20.7204  -3.2955  -15.3501  -24.5636  8.4250
-20.3717  -89.9498  24.4259  0.5598  -5.7363  0.8946  3.4819  6.1233  -2.0678
```

Donde las tres primeras columnas representan los 3 primeros estados (Posiciones), las siguientes 3 columnas representan los siguientes 3 estados (velocidades), y las 3 últimas representan la parte integral asociada a las posiciones. Las filas representan cada una de las articulaciones del modelo, lo cual se deriva en una matriz K de 3 x 9.

Como otro método para Validar que si está bien linealizado el sistema, se construye un modelo de pequeña señal (lineal), descrito por las matrices A, B, C, D y se

implementa con los mismos valores de la ganancia K , en las figuras 15 y 16 se muestran los modelos (Lineal, No lineal) implementados en MATLAB con la arquitectura de control por realimentación de estados con el integrador.

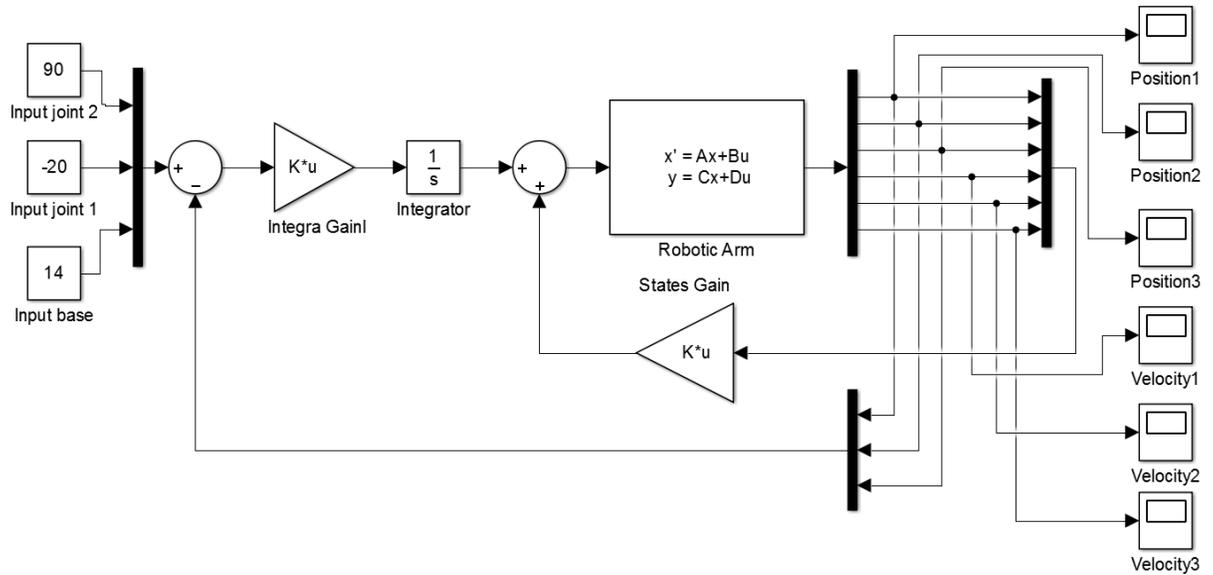


Figura15. Modelo Linealizado

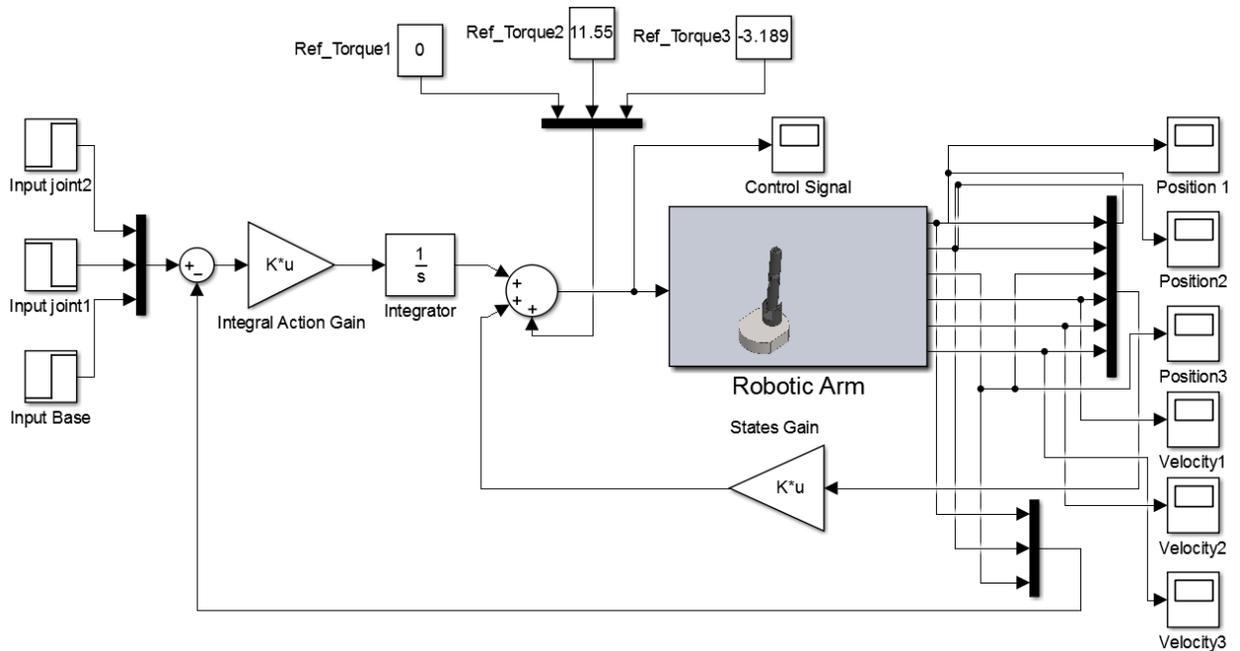


Figura16. Modelo simmecanichs

Para ambos modelos se hace una entrada paso de 30° dando como resultado la figura 17 y 18.

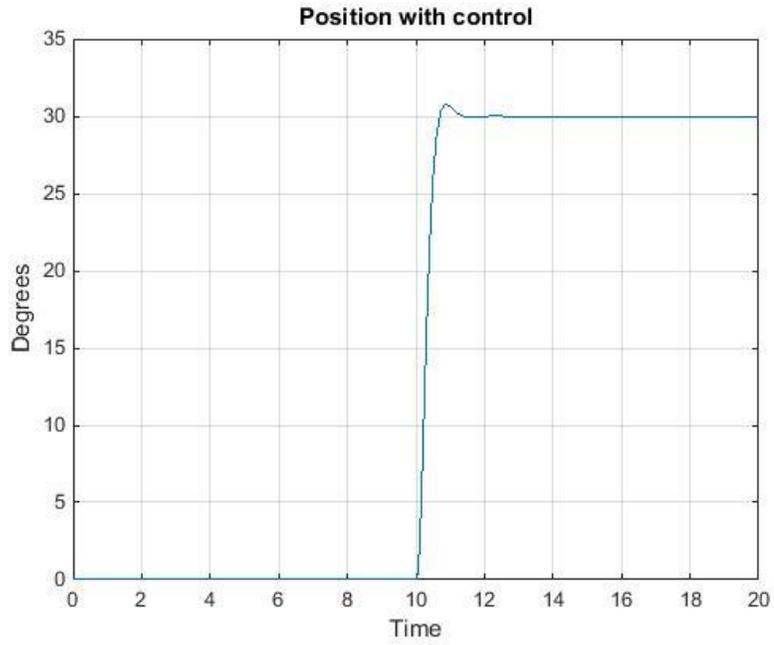


Figura17. Respuesta paso segunda articulación modelo lineal

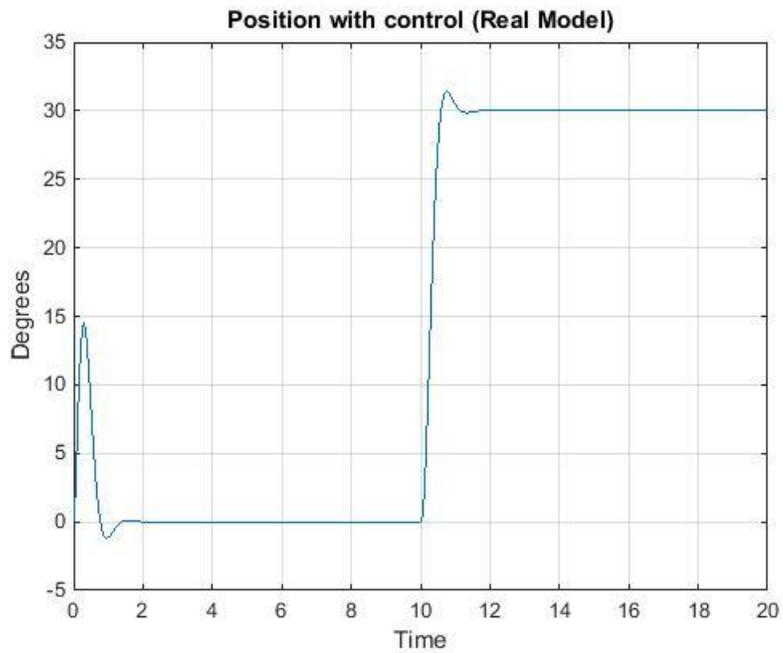


Figura18. Respuesta paso segunda articulación modelo no lineal

En la figura 17 se observa la respuesta para el modelo lineal y en la figura 18 la del modelo no lineal, de manera cualitativa se logra observar que al iniciar la simulación el modelo no lineal debe compensar la caída del brazo por acción de la gravedad, esto le toma aproximadamente 2 segundos, mientras que el modelo lineal no posee ese problema, sin embargo para tener un criterio más exacto de cómo se comportan ambas señales, se hace el cálculo del error cuadrático mediante la función MSE de MATLAB, teniendo como base la respuesta del modelo lineal y se obtiene un error de 6.53%, lo cual quiere decir que la linealización hecha por la función linmod si describe el brazo robótico hecho en SOLIDWORKS.

CAPITULO 4 CONTROL LPV MEDIANTE GAIN SCHEDULING

LPV (Linear parameter-varying) como su nombre lo indica es un modelo descrito por un vector de parámetros del mismo que varían en el tiempo, para este trabajo se compone de los modelos obtenidos en cada punto de trayectoria que cambian sus parámetros a medida que va cambiando la posición. Para cada sistema se calcula un controlador que lo estabilice, con ganancias que conmutan entre ellas a medida que cambian los parámetros del LPV, a esto se le conoce como (Gain scheduling), y permite hacer un control robusto y suavizado en cada punto de movimiento del sistema.

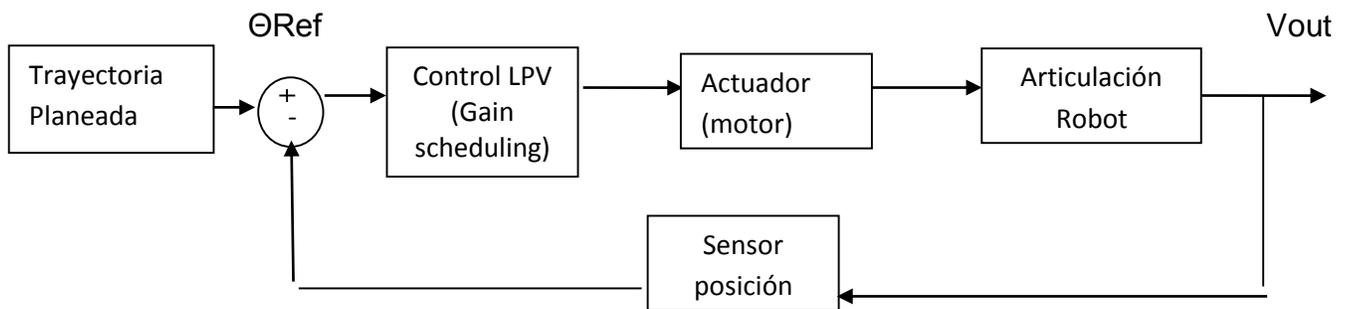


Figura 19. Arquitectura general Control LPV

En la arquitectura mostrada en la figura 19, se tiene el modelo general del control a implementar, en dónde el controlador para la dinámica del robot es un control LPV (Linear parameter- Varying), que va comparando la entrada de referencia y la posición actual para saber en qué momento varían los parámetros del modelo y así seleccionar la K adecuada para el control del brazo robótico, dicho control tendrá

acción principal sobre los actuadores del brazo, que serán posteriormente seleccionados, una vez hecho el análisis de torques y fuerzas del modelo.

4.1 INTERPOLACIÓN DE VALORES DE GANANCIA Y TORQUES

Con la K hallada al calcular el control por realimentación de estados (procedimiento sección 2.2), se seleccionan puntos a evaluar entre -50° y 50° en pasos de 10° (ya que cada 10° se presenta una variación entre parámetros de las matrices A), para cada articulación ver figura 20.

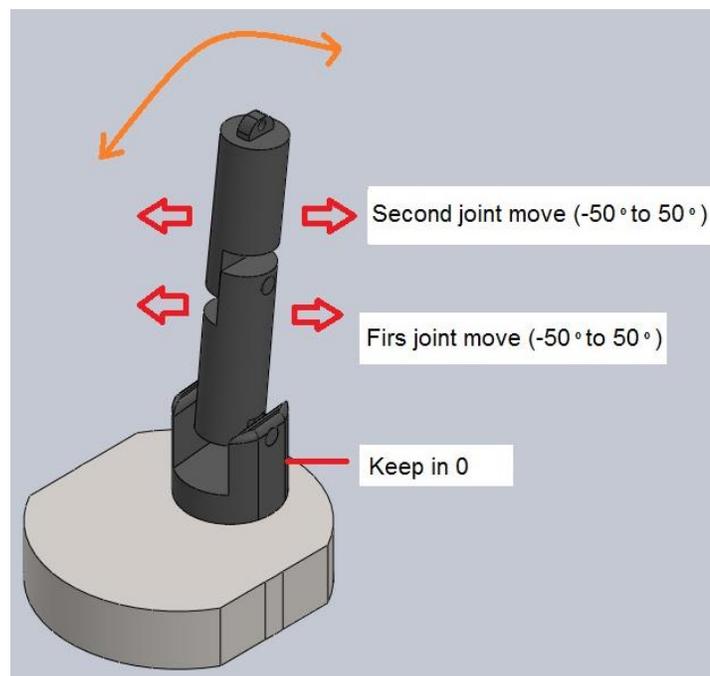


Figura 20. Descripción movimiento

Cada par de valores de θ_1 (Ángulo de movimiento primera articulación) y θ_2 (Ángulo de movimiento segunda articulación) generan un modelo, al cual se le calcula un valor de K (3×9) para controlarlo en ese punto.

Como se desea hacer un control para cada punto de trayectoria, se decide hacer una interpolación de los valores de K , ya que hay muchos valores que no se calculan y deben ser hallados con el fin de evitar un salto brusco que pueda inestabilizar el modelo cuando pasa de un punto a otro, es decir, si el modelo pasa de $(-30^\circ$ a $-25^\circ)$, se debe saber el valor de K que controla en -25° , de lo contrario el modelo se inestabilizará.

Para interpolar los valores de manera correcta, primero se genera una superficie para cada uno de los 27 valores de K cambiando en función de theta 1 y theta 2 ver figura 21.

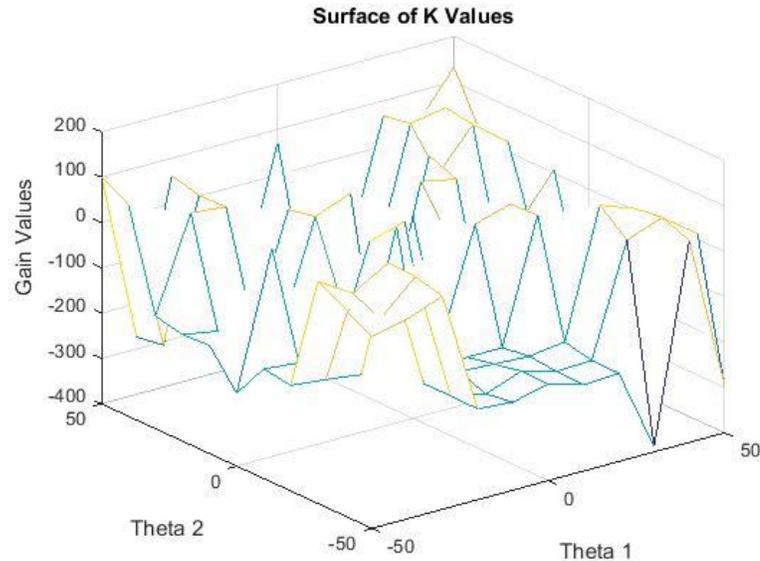


Figura21. Superficie primer valor de K por ubicación de polos

En la figura 20 el eje X representa los valores que toma theta 1, el eje Y representa los valores que toma theta 2, y el eje Z representa los valores de K(1,1) para cada par de posiciones. Cabe anotar que esto se hace para cada uno de los 27 valores de K.

Al analizar la gráfica, se nota una variación muy grande de valores, lo cual genera un problema a la hora de interpolar los valores, al estar tan lejanos es muy complejo hacer una interpolación correcta.

Como alternativa se propone hacer el control del modelo en cada punto por otro método, en este caso el método a utilizar es LQR (Linear Quadratic Regulator), ya que este método resuelve un problema de optimización y genera una superficie convexa entre valores de K. Se tiene el modelo lineal.

$$\dot{X} = AX + BU \quad (1)$$

Para controlarlo se usa la función de costo.

$$J = \int_0^{\infty} (X^T Q x + U^T R u) dt \quad (2)$$

Con la que se busca implementar un control

$$U = -KX \quad (3)$$

En la función de costo la matriz Q penaliza todos los estados, mientras que la matriz R penaliza las entradas del sistema, aunque no hay una ley que diga como seleccionarlas, se usa el criterio de Bryson, porque se conocen los valores máximos que se desean en los estados, esta ley propone de manera general.

$$Q_{ii} = \frac{1}{(t_f - t_0) \max(|u_i(t)|^2)} \quad (4)$$

$$R_{jj} = \frac{1}{(t_f - t_0) \max(|x_j(t)|^2)} \quad (5)$$

Con $i=1, 2, 3, \dots, n$ y $j=1, 2, 3, \dots, n$, cabe anotar que tanto la matriz R como la matriz Q son matrices diagonales, cuyos valores están compuestos por las variables de interés. Para cada valor de estado se debe dar un peso o cuantificación, dando mayor prioridad a las variables que se desean controlar, en este caso son las posiciones de las articulaciones del brazo.

Para minimizar la tarea de cálculo se usa la función de matlab LQR, la cual recibe las matrices Q, R diagonales con sus respectivos valores y las matrices A, y B del modelo.

Una vez obtenido el valor del K óptimo para controlar el brazo, se verifica que la señal de control tenga valores de torque moderados, así como las posiciones de las articulaciones correspondan a la entrada del modelo, se vuelve a graficar la superficie para cada valor de parámetros y se obtiene la figura 22.

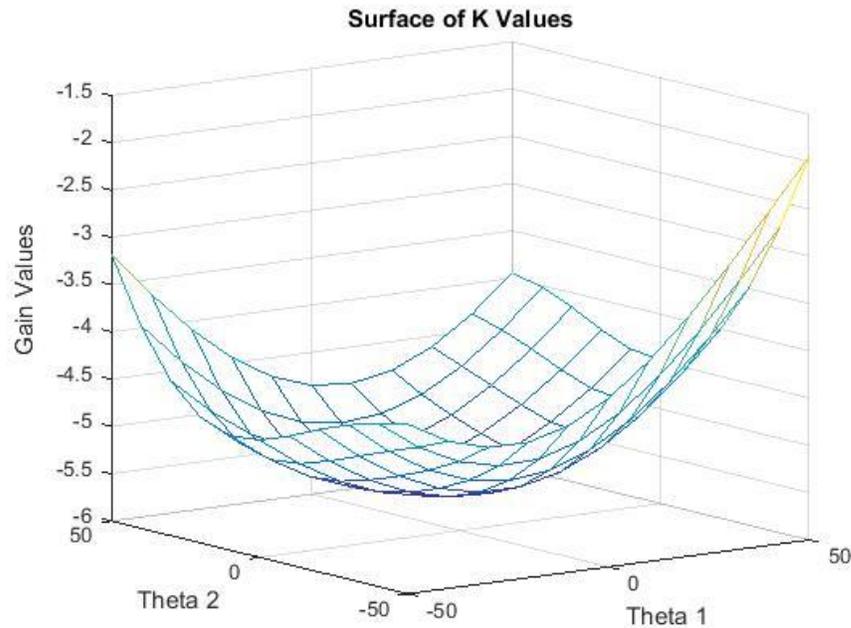


Figura22. Superficie primer valor de K por LQR

Como se puede ver en la figura 21, los valores de K varían de forma suave y en menor proporción que en el caso anterior, generando una región convexa como se estimaba, estos valores son más sencillos de interpolar debido a su pequeña variación. Con estos valores suavizados, ahora se deben interpolar.

La interpolación es la obtención de nuevos valores partiendo del un conjunto discreto de puntos conocidos que describen una función, un ejemplo de interpolación es la interpolación lineal, es el método más sencillo y utiliza dos puntos conocidos (X_a, Y_a) y (X_b, Y_b) para hallar un tercer punto interpolado (X, Y) , matemáticamente.

$$Y = Y_a + (X - X_a) \frac{(Y_b - Y_a)}{(X_b - X_a)} \quad (1)$$

Como se ve en la ecuación (1) se usan los valores conocidos en X y en Y para hallar el tercer valor, sin embargo hacer esta fórmula y programarla para todas las superficies de K generadas sería muy complejo, así que se decide usar la función de MATLAB `interp2` donde se introducen los dos parámetros (En este caso theta 1 y theta 2), los cuales causan el cambio de valor de K al variar la posición, para hallar los valores intermedios de cada K. El método de interpolación usado en la función `interp2` es el método lineal que está por default en la función.

La interpolación de valores se hace para K_{11} es decir entre los 121 valores de K hallados mediante el LQR, y se implementa en una función dentro de la arquitectura de control LPV.

4.2 DESEMPEÑO CONTROLADOR

Una vez obtenidos todos los valores de K , se construye la arquitectura de control mostrada en la figura 23.

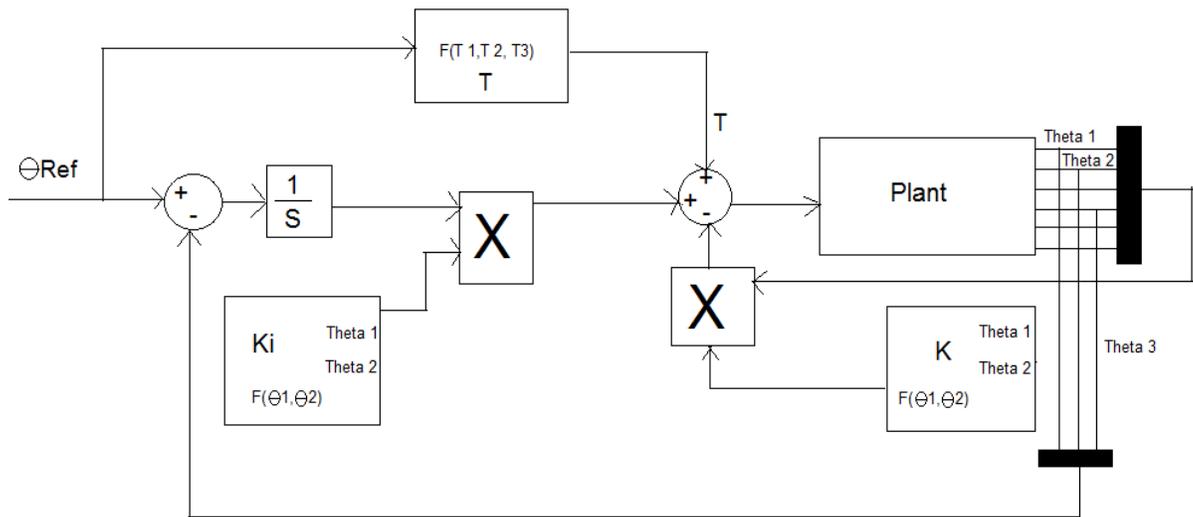


Figura 23. Arquitectura Control LPV con Gain scheduling

Donde K , y K_i son funciones de interpolación que varían su salida dependiendo de las variables θ_1 y θ_2 , mientras que T es una función que interpola los valores de torque y depende de las posiciones de referencia (Posiciones de la trayectoria seleccionada).

Cuando el modelo recibe la señal de referencia, las funciones K_1, K_2 (funciones de interpolación) dependiendo de los valores de θ_1 y θ_2 , buscan el valor apropiado de K para esa posición, mientras que la función de torques busca el valor del torque exacto para la posición deseada, esto le da más velocidad de respuesta al control, y robustez ante perturbaciones, debido a que la función de torques hace un feed forward a la entrada, manteniendo así la posición de referencia, es decir, a la función torques, le entran los valores de θ_1 , θ_2 y θ_3 (Posiciones de

las articulaciones). Se elige hacer un feed forward de los torques para dar mayor velocidad a la acción de control y dar mayor robustez ante perturbaciones externas.

Probando el modelo en la trayectoria seleccionada: Inicial (Base = 0° , Articulación 1 = -20° , Articulación 2 = 15°). Final (Base = 0° , Articulación 1 = 20° , Articulación 2 = 0°). Se obtienen las salidas de las articulaciones para comprobar la acción de control figura 24 y 25. La entrada que se usó para trazar el movimiento fue una entrada paso.

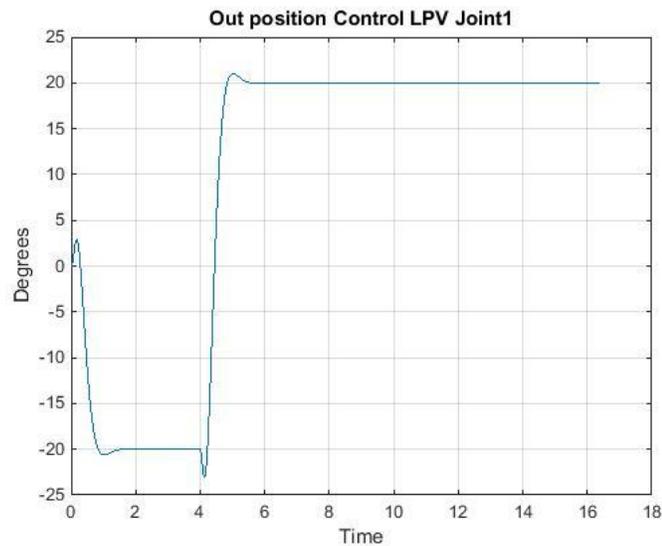


Figura 24. Respuesta movimiento articulación 1.

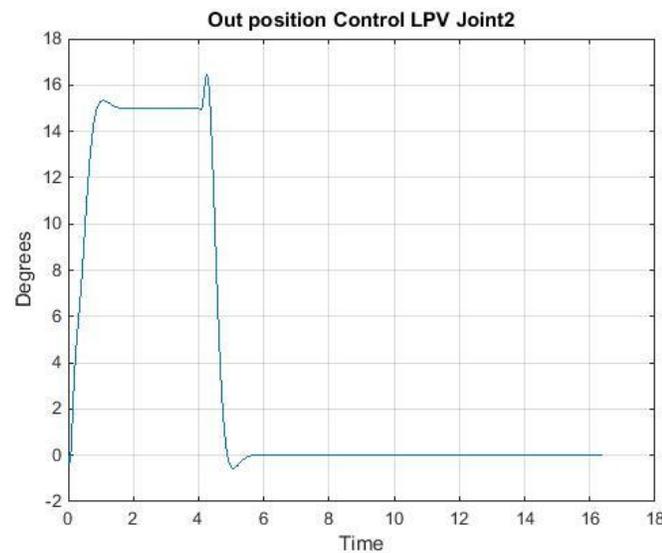


Figura 25. Respuesta movimiento articulación 2.

Como las condiciones iniciales del modelo son (0, 0, 0), el modelo tarda 1.5 segundos en llegar a la posición inicial de la trayectoria compensando los efectos de la gravedad el tiempo de estabilización una vez aplicada la señal paso es de 1.8 segundos y el sobrepico $M_p = 15\%$ para ambas articulaciones.

Aplicando una perturbación de 1Nm a los 5 segundos de iniciada la simulación, se observan de nuevo las posiciones y la respuesta del controlador Ver figuras 26 y 27.

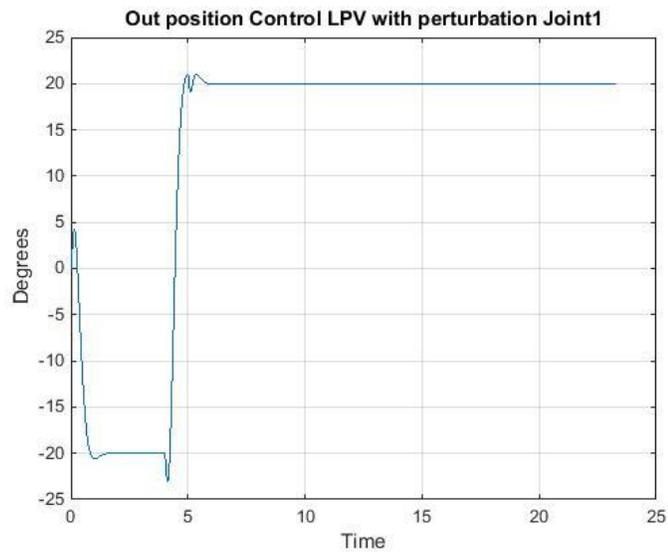


Figura 26. Respuesta controlador perturbación primera articulación

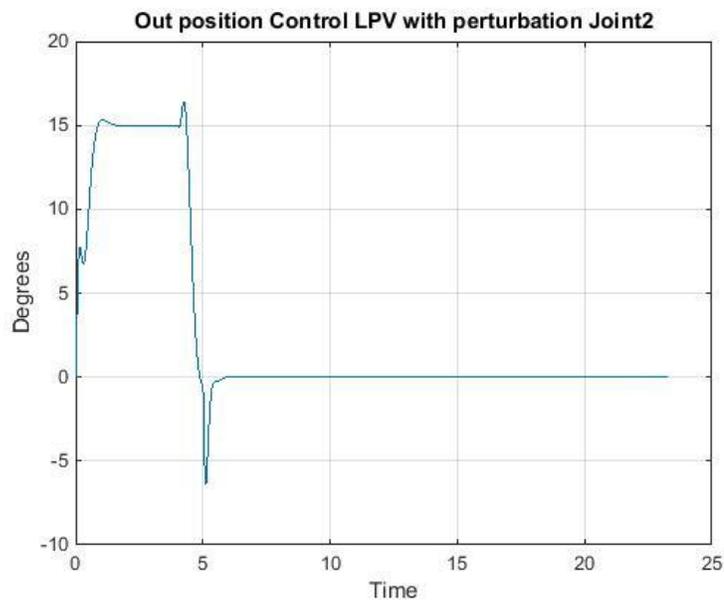


Figura 27. Respuesta controlador perturbación segunda articulación.

En la figura 27 se aprecia como la perturbación altera la respuesta del controlador, aunque el tiempo de estabilización se mantiene en 1.8 segundos, el sobre pico sube considerablemente al 50%, lo que indica que la perturbación afecta más la acción de control para la segunda articulación cuando se recorre la trayectoria planeada.

Aplicando diferentes tipos de señales cuadrada, diente de Sierra y sinusoidal, a una frecuencia de 0.2 Hz y de amplitud 20° sobre la primera articulación del brazo robótico, se hace una comparación de tiempos de establecimiento, respuesta a perturbaciones, error RMS tracking, sobre pico que pueden ser apreciados en la tabla 2.

Tipo de señal de Entrada	Sobre pico (%)	Tiempo estabilización (s)	Error de tracking (%)	Respuesta a perturbación (s)
Cuadrada	2.5	2	2.12	3
Diente de Sierra	–	–	1.8	4
Sinusoidal	–	–	1.17	4.5

Tabla 2. Valores de desempeño Control LPV

Otro aspecto importante a analizar es la señal de control, ya que esta debe estar acorde al peso y tamaño del brazo robótico, y teniendo en cuenta que el material seleccionado para la construcción del brazo robótico no es muy pesado, se esperaría que la señal de control tenga valores acordes a cada una de las piezas a controlar.

Se observa la señal de control en la trayectoria seleccionada y con la señal de perturbación de 1Nm a los 5 segundos, para la primera articulación (la que más torque requiere) figura 28.

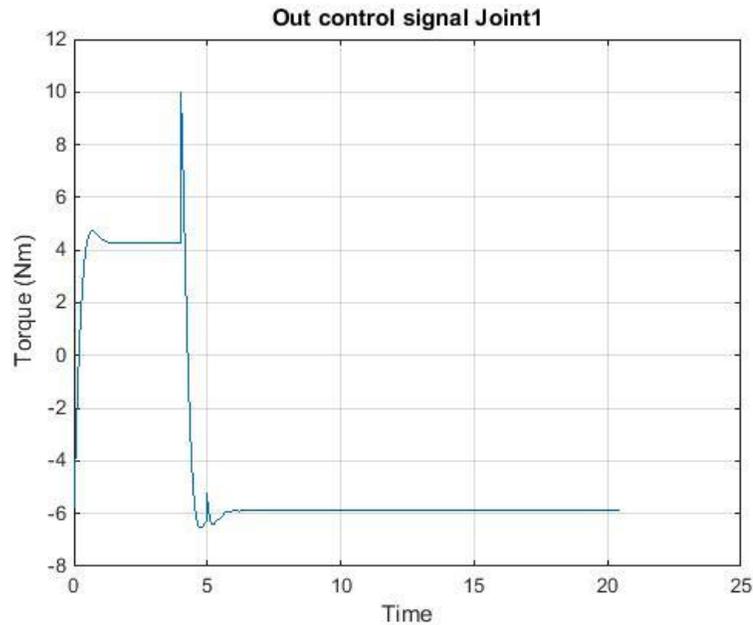


Figura 28. Señal de control primera articulación

En la figura 28 se ve un valor máximo de 10Nm recorriendo la trayectoria seleccionada, y si bien a los 5 segundos se presenta la perturbación, no requiere de más de 1Nm corregir para seguir la trayectoria. Este dato será tenido en cuenta más adelante para la selección de instrumentación. Cabe aclarar que el pico de 10Nm que se aprecia antes de los 5 segundos en la señal de control se da por el tipo de entrada con el que se hizo la prueba (Entrada paso).

CAPITULO 5. SELECCIÓN DE INSTRUMENTACIÓN

Como parte final de este trabajo se deja la selección de instrumentación, ya que se tienen criterios mucho más sólidos para hacer la selección, tales como, torques, posiciones y pesos aproximados de la estructura. Para una buena selección se tienen en cuenta varios fabricantes y se tienen como referencia algunos robots consultados en el estado del arte [3].

5.1SELECCIÓN MOTORES

Para iniciar la selección de motores se deben tener en cuenta muchos factores que inciden en la selección, como son: Protección IP68, tamaño, torque, peso.

Analizando la Figura 28 después de implementado el control LPV, se ve que el torque máximo está en el orden de los 10N.m, y por eso se debe seleccionar un motor que cumpla con estas especificaciones de torque.

Con los valores de la tabla 3, se tiene que la base tiene un peso de 4.14 Kg, ahora para poder hacer la selección del motor adecuado es necesario tener en cuenta los pesos de las otras dos piezas más un aproximado de su instrumentación, con el fin de no sobredimensionar el motor. Dicho esto, el peso de la primera articulación es de 3.6Kg y de la segunda articulación es de 3.3Kg, Consultando un catálogo de motores dc de control numérico de la marca STOGRA de Alemania [15], especializada en motores con protección IP68 para todo tipo de aplicación, se pudo comprobar que las dimensiones se ajustan al modelo diseñado, ya que su alto y diámetro no exceden las dimensiones de las piezas diseñadas (Ver tabla 1).

Sin embargo se decide consultar otras marcas como SIEMENS, IGUS y ZD MOTOR, para hacer una comparación en cuanto a características y así tomar una mejor decisión. Para esto se hace una tabla comparativa (ver tabla 3) de los modelos que más se ajustan a lo requerido, resaltando las características más importantes para la selección, en este caso son: Tamaño, Torque, Nivel de protección, peso.

MARCA	Modelo	Torque Máximo (Nm)	Nivel de Protección	Peso (Kg)	Voltaje (VDC)	Dimensiones (Largo,Ancho cm)
Siemens	1FK7CT	3.4	IP64	2	24	(16.3 ,9.6)
IGUS	MOT-C-AAAM	1.3	IP68	1.51	24-48	(17.1 , 5.64)
ZD MOTOR	90ZY-196-120W	8.76	N.A	3.2	24	(19.6 , 10.4)
STOGRA	SM 88.1.18M4	3	IP68	1.9	24-240	(14.5 , 8.6)

Tabla 3. Tabla comparativa de motores

Como se puede apreciar en la tabla 3 los motores tienen características muy parecidas, sin embargo el motor que más se ajusta a lo requerido es el motor de STOGRA el SM88.1.18M4, ya que cumple con las dimensiones, protección y bajo peso. En cuanto al torque cabe aclarar que para aumentar dicho torque se debe seleccionar un reductor que sea capaz de multiplicar ese torque a razón de 5 veces aproximadamente, esta caja de engranajes aumenta un poco las dimensiones del motor, pero esto no altera la selección ya que el aumento no es superior a los 5cm de largo, por esto y por el nivel de protección se selecciona este motor para la primera, segunda y tercera articulación, ya que observando los torque máximos ejercidos en cada una de esas articulaciones, no superan los 15 Nm.

En la figura 29 se muestra el motor seleccionado.

IP68 motors – Z9 and Z177



Figura 29. Motor STOGRA IP68

5.2SELECCIÓN SENSORES Y EFECTOR FINAL

Como medida principal se tendrá la posición de cada articulación, ya que esta información es importante para hacer el control, precisamente es la variable a controlar, por eso es muy importante saber seleccionar un instrumento capaz de hacer esta medición lo más preciso posible.

El motor seleccionado tiene la opción de integrar un codificador óptico, el cual puede ser seleccionado mediante un código y bajo pedido, sin embargo para la aplicación que se requiere el codificador óptico incorporado no cumple con las especificaciones mínimas, al ser un codificador óptico incremental no permite hacer la medición de posición absoluta del motor. Para solucionar este problema, se decide seleccionar un codificador óptico absoluto de la marca ALLEN-BRADLEY [21]. El modelo seleccionado es 845G-F3D8HC1024T, ya que posee protección IP66 y sus dimensiones son reducidas (2.5 X 2.3 X 2.5 cm).

Otro criterio de selección fue su resolución ya que.

$$(\text{º}) = \frac{\text{Edge_Count}}{x N} 360\text{º} \quad (1)$$

Con N= Número de pulsos generados por giro del eje (en este caso 1024), y x= Tipo de codificación (En este caso es una codificación x1 = 1). Luego la resolución del codificador óptico seleccionado es de 0.3 º, más que suficiente para hacer la medición de posición absoluta de cada articulación del robot. En la figura 30 se puede ver una foto del sensor seleccionado.



Figura 30. Codificador óptico Seleccionado

SELECCIÓN EFECTOR FINAL

Para garantizar un lijado uniforme de la pieza se debe seleccionar una herramienta adecuada que cumpla con esta función, y aunque en la normativa colombiana de inspección de tuberías hidrosanitarias NS-058 del acueducto de Bogotá, no especifica un procedimiento exacto para la adecuación de piezas antes de ser reparadas, si se hizo un estudio previo para este proceso. Este estudio estuvo a cargo del ingeniero Gabriel Guiza [2], en el cual se marca la importancia de lijar la tubería para permitir que se adhiere el material de reparación y así evitar un re trabajo que además de generar un sobrecosto, genera retrasos y desperdicio de material.

Por estas razones y por sugerencia de los colaboradores de la Universidad Militar Nueva Granada, se elige como efector final un cepillo tipo copa de alambre como el que se ve en la figura 31.



Figura 31. Cepillo copa de alambre para tuberías

Este tipo de herramienta permite llegar más fácilmente a los puntos de difícil acceso o de geometría complicada como una tubería. Por otra parte, haciendo pruebas de desempeño se concluye que el efector final girando a 400rpm puede perfectamente lijar y pulir la parte afectada para luego ser reparada.

Para esta última tarea se elige un motor con protección IP68 que gira a altas revoluciones, y de un torque máximo por debajo de 1Nm ya que la tarea a realizar es más de precisión que de fuerza. El motor STOGRA SM56.1.18 J1 tiene un torque

máximo de 0.45Nm y además puede llegar a un máximo de 2100rpm, más que suficiente ya que según pruebas hechas por la universidad militar las revoluciones necesarias para lograr un lijado uniforme están por debajo de las 1000rpm. Como se había dicho anteriormente los motores STOGRA cumplen con dimensiones, protección, torque y peso, por esta razón se elige otro motor de la misma marca para hacer girar el efector final.

6. CONCLUSIONES

Si bien se puede hacer un control mediante gain scheduling, se debe tener especial cuidado con el método que se utiliza para hacer el control en cada punto de trayectoria (LQR, Ubicación de polos, PID, etc), ya que como se vio en la sección de interpolación, si el método usado para calcular el control no genera valores de K uniformes, se tendrán superficies muy complejas de suavizar, y por ende se tendrán problemas de interpolación causando que el control LPV sea inestable.

Al hacer el modelo en SOLIDWORKS y pasarlo al entorno de MATLAB, se hace más sencillo el análisis, ya que muchos parámetros no se tienen en cuenta cuando el modelo es descrito por ecuaciones analíticas, por esta razón si se desea hacer un control a algún complejo mecánico y así mismo tener una herramienta visual de simulación de dicho control, es recomendable primero modelarlo en el programa SOLIDWORKS y luego mediante la herramienta SIMMECANICHS pasarlo al entorno de MATLAB.

La herramienta Linmod de MATLAB evita en gran parte resolver ecuaciones bastante complejas, y ayuda a obtener modelos linealizados exactos, sin esta función no hubiera sido posible linealizar en tan poco tiempo cada modelo en los puntos de operación.

No importa el método de interpolación que se use en la función interp2, el resultado es el mismo, es decir, es lo mismo usar el método spline que linear, u otro diferente, ya que se predicen los valores con igual precisión.

Como trabajo futuro se propone implementar el control en un brazo robótico industrial conocido, con el fin de tener una aproximación más acertada al propuesto. También se propone usar un método diferente a un LPV y Ubicación de polos para hacer el control en cada punto de operación, se propone que pueda ser un sliding mode o PID, con el fin de comprobar que si uno de los métodos usados posee regiones convexas, es posible implementar un gain scheduling mediante interpolación de valores.

7. BIBLIOGRAFÍA

- [1] Ing. Gabriel Guiza. *Reparación de tuberías sin Zanja.determinación de la metodología de reparación. Universidad de los andes [Octubre 2013].*
- [2] *Acueducto de Bogotá. Norma Técnica De Servicio NS-058. [Online]. [Noviembre 6 2012].* <http://www.minminas.gov.co/minminas/downloads/UserFiles/File/ENERGIA/RETILAP/Propuesta-Modificacion-AnexoRETLAPmodificado-11.pdf>
- [3] KUKA. *KR agilus 5 .Specification [Online]. Disponible en:* http://www.kuka-robotics.com/en/products/industrial_robots/small_robots/kr6_r700_fivve/start.htm
- [4] MITSUBISHI. *Configuration manual .F series robot [Online]. Disponible en:* http://mitsubishirobotics.com/pdf/MEAU_product_catalog.pdf
- [5] ABB ELECTRONICS, *Product specification, IR120 [Online]. Disponible en:* <https://library.e.abb.com/public/403d0cc5652905e8c1257e30003e91e8/3HAC048645-en.pdf?filename=3HAC048645-en.pdf>
- [6] Bruno Siciliano, *Robotics modelling, planning and control, Springer, 644p. [2009]*
- [7] *Hybrid Impedance Control of a 3dof Robotic Arm used for Rehabilitation Tratament Jjingguo wang and Yangmin li. 6th anual IEEE Conference on automation science and engineering, Toronto, canada, [agosto 2 2010].*

- [8] *Design Adaptative Fuzzy Interference Controller for Robot Arm.* Mostafamirzadeh, Mohamad Haghghi, Saeed Kherzi, Javad mohamoodi, Hasan Karbasi. University of iran, Department of electrical engineering. [Agosto 2014].
- [9] *Genetic Fuzzy Based Tracking Control of 3dof Robot Arm.* Srinivassan Alavandar, M.j Niggam. Indian Institute of technology Roorkee, India.
- [10] *LPV Aproach to Control and Real-time Scheduling Codesign Aplication to Robot Arm Control,* Oliver sename, Daniel simon, Mohamed El mongi Ben Gaid [Online] Disponible en: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4738778&tag=1 [2009].
- [11] KUKA. Catalogue..Specification [Online]. Disponible en: http://www.kuka-robotics.com/en/products/industrial_robots/small_robots
- [12] SCORBOT ROBOT, User Manual, Specification. [Online]Disponible en: http://www.theoldrobots.com/book45/ER4u_User_Manual.pdf
- [13] Peter Corke. Robotics toolbox,Robotics and visión moocs . [Online]. Disponible en: http://www.petercorke.com/Robotics_Toolbox.html
- [14] *Polytopic LPV Gain Scheduling Control for Flexible Air Breathing HypersonicVehicle.* Haug Yiqing, Sun changyn, Quian Chegshan, Zhang Ruimin Zhangji Jingmei. 31st Chinese Control Conference. [July 2012]
- [15] STOGRA. Stepper motors. [Online]. [2015]. Disponible en: http://www.micromech.co.uk/dir_products/pdf/stogra/stepper_motor.pdf
- [16] STOGRA, Stepper motors. stogra sm 88.1.18m4 [Online]. [2015]. Disponible en: http://www.stoegra.de/pdf/sm_daten_elektr_abmasse_e.pdf
- [17] *Measurement Specialities, Todo tipo de sensores de medida,* [Online]. [Agosto 30 2012]. Disponible en: <http://www.meas-spec.com>
- [18] OPTORESOLVERS, Resolver vs Encoders [Online]. [Septiembre 22 2012]. Disponible en: http://www.optoresolver.com/help/tutorials/resolver_v_encoder.htm
- [20] NEMA, the Association of Electrical Equipment and Medical Imaging Manufacturers. [Online]. [Septiembre 2 2012]. Disponible en: <http://www.nema.org/Products/Pages/Enclosures.aspx>
- [21] Allen Bradley, Encoders absolutos.Especificaciones, [Online]. [2015]. Disponible en: http://literature.rockwellautomation.com/idc/groups/literature/documents/in/845g-in001_-en-p.pdf

[22] NEMA, *NEMA enclosure Types [Online]*. [2015]. Disponible en: <https://www.nema.org/Products/Documents/nema-enclosure-types.pdf>