



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

**FACULTAD DE CIENCIAS DE LA ELECTRÓNICA
MAESTRÍA EN CIENCIAS DE LA ELECTRÓNICA
OPCIÓN EN AUTOMATIZACIÓN**

**“Validación de Algoritmos de Control
de Impedancia en un ROTRADI”**

TESIS PRESENTADA OBTENER EL TÍTULO DE:

**MAESTRO EN CIENCIAS DE LA ELECTRÓNICA OPCIÓN
EN AUTOMATIZACION**

Presenta: Lic. Oswaldo Tepal Nieto **

Directores: Dra. Olga Guadalupe Félix Beltrán [FCE-BUAP]
Dr. Fernando Reyes Cortés [FCE-BUAP]
Dr. Jaime Cid Monjaraz [FCE-BUAP]

Puebla, México

Diciembre de 2016

**Becario CONACYT

**Becario VIEP

BUAP®

*“Los Asuntos del día deben resolverse
con Sangre y Hierro,
pero nosotros como científicos
debemos resolverlos solo con Hierro.”*

AGRADECIMIENTOS

Quiero expresar a través de estas líneas mi más profundo y sincero agradecimiento a todas aquellas personas que con su ayuda han colaborado en la realización del presente trabajo de tesis.

A mis asesores de tesis Dra. Olga Guadalupe Félix Beltrán, Dr. Fernando Reyes Cortés y Dr. Jaime Cid Monjaraz por todo el apoyo y dedicación brindada a lo largo de la maestría.

Al jurado evaluador por sus comentarios que ayudaron a enriquecer este trabajo.

A la Facultad de Ciencias de la Electrónica por haberme dado la oportunidad de realizar mis estudios de maestría.

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca otorgada para realizar mis estudios de maestría.

RESUMEN

Para el presente trabajo de tesis se implementaron tres algoritmos de control de impedancia en un robot de transmisión directa de 3 grados de libertad donde se empleó una tarjeta Arduino Mega, así como una interfaz electrónica diseñada a partir del análisis de funcionamiento del robot.

En este documento se describen diferentes sistemas dinámicos de robots manipuladores comenzando con un simple sistema de una centrifuga y un péndulo, considerados ambos sistemas de 1 grado de libertad para propósitos de robótica, así como un robot de 2 y 3 grados de libertad a través de la metodología de Euler-Lagrange de donde se obtienen los parámetros necesarios para obtener el modelo dinámico de cada robot.

El modelo del robot de tres grados de libertad es aplicado en conjunto con los parámetros físicos del robot de transmisión directa (ROTRADI VI) para la aplicación de varios esquemas de control, entre ellos el control de posición articular, control de posición cartesiano, control de fuerza y control de impedancia. El ROTRADI VI es un robot compuesto de eslabones de aluminio unidos por motores de transmisión directa, los cuales tienen la particularidad que no utilizan ningún tipo de transmisión mecánica como engranes o cadenas para obtener incremento de torque o par. Estos motores a su vez pueden ser configurados de diferentes maneras para la lectura de la posición angular destacando principalmente el modo diferencial de señales A-B (*encoder* de cuadratura) y modo de pulsos por direccionamiento.

Como parte del desarrollo del sistema de control se llevó a cabo el diseño e implementación un sistema electrónico para el brazo robótico de tres grados de libertad, compuesto por tarjetas electrónicas DAC, filtros de ruido y una tarjeta Arduino Mega que en conjunto componen la interfaz de control que permite el funcionamiento de cada uno de los actuadores del ROTRADI a través de valores de señales analógicas de voltaje proporcionales a cierto torque de cada actuador y de la obtención de la posición angular acondicionada por las señales de diferenciales de cada *encoder*.

Para las estrategias de control de fuerza y control de impedancia, se aplica el uso un sensor de fuerza conformado por una celda de carga diferencial con una capacidad de 5 kg y la

electrónica de adquisición de datos correspondiente formada por un sistema de amplificadores de instrumentación el cual devuelve un valor de voltaje analógico proporcional al peso aplicado sobre la celda. Estos datos son utilizados para obtención de fuerza.

Finalmente se diseñó un programa que permite la implementación de múltiples estrategias de control modificando un único bloque correspondiente a la ecuación de control experimental. Estas estrategias de control culminaron en el diseño de una aplicación de control “manual” para rutinas de fisioterapia la cual consiste en el movimiento manual de los eslabones por parte del usuario, permitiendo que el ROTRADI sea capaz de reproducir dicha rutina para la aplicación de múltiples ejercicios según requería algún paciente que requiera de estas rutinas.

ÍNDICE GENERAL

AGRADECIMIENTOS.....	3
RESUMEN.....	4
INTRODUCCIÓN.....	9
OBJETIVOS.....	13
CAPÍTULO 1: ANTECEDENTES.....	14
1.1 PARTE MECÁNICA	14
1.2 COMPONENTES DE LA PLANTA.....	16
1.3 DEFINICIÓN DE IMPEDANCIA MECÁNICA.....	19
1.4 MODELADO DINÁMICO DEL ROBOT.....	20
1.5 FORMULACIONES PARA LA IMPEDANCIA MECÁNICA.....	21
1.6 ROTRADI VI.....	21
1.6.1 CARACTERISTICAS DEL ROTRADI VI.....	22
1.6.2 CONFIGURACION.....	24
1.7 MICROCONTROLADOR ARDUINO.....	24
1.7.1 ARDUINO MEGA 2560.....	25
1.8 SENSOR DE FUERZA/PAR.....	27
1.8 MODELO DINÁMICO DE UNA CENTRÍFUGA.....	28
1.9 MODELO DINÁMICO DE UN ROBOT DE 2 GDL (RR).....	30
CAPÍTULO 2: MODELADO DEL ROTRADI VI.....	34
2.1 MODELADO DINÁMICO DE UN ROBOT DE 3 GDL.....	34
CAPÍTULO 3: ELECTRÓNICA DE CONTROL.....	37
3.1 TRATAMIENTO DE LAS SEÑALES DE LOS ENCODERS	37
3.2 ANÁLISIS DEL ALGORITMO DE LECTURA DEL ENCODER	40
3.3 OBTENCIÓN DE DATOS DE POSICIONAMIENTO	42
3.4 CONVERTIDOR DIGITAL ANALÓGICO DE 10 BITS.....	43
3.4 SENSOR DE FUERZA.....	46

	7
3.4.1 IMPLEMENTACIÓN.....	48
CAPÍTULO 4: ALGORITMOS DE CONTROL DE ARDUINO	54
4.1 CONTROL DEL POSICIÓN ARTICULAR.....	54
4.1.1 CONTROLADORES PROPUESTOS.....	56
4.2 CONTROL DE POSICIÓN CARTESIANO.....	56
4.2.1 CONTROLADORES PROPUESTOS.....	57
4.3 CONTROL DE FUERZA.....	58
4.3.1 CONTROLADORES PROPUESTOS.....	58
4.4 CONTROL DE IMPEDANCIA	58
4.4.1 CONTROLADORES PROPUESTOS.....	60
4.5 RUTINA DE CONTROL FISIOTERAPÉUTICA.....	61
CAPITULO 5: RESULTADOS	63
5.1 RESULTADOS DE CONTROL DE POSICIÓN ANGULAR	63
5.2 RESULTADOS DE CONTROL CARTESIANO.....	68
5.3 RESULTADOS DE CONTROL DE FUERZA.....	73
5.4 RESULTADOS DE CONTROL DE IMPEDANCIA.....	76
CONCLUSIONES	82
APÉNDICE A: MODELO DINÁMICO DE UN ROBOT ANTROPOMÓRFICO DE 3 GRADOS DE LIBERTAD	84
APÉNDICE B: CONFIGURACIÓN DE DRIVERS.....	89
B.1. PUESTA EN OPERACIÓN DE SERVO DRIVERS.....	89
B.1.1. PREPARACIÓN DE SERVO DRIVERS	89
B.1.2. CONFIGURACIÓN PRELIMINAR DEL DRIVER.....	96
B.2. CONFIGURACIÓN ELECTRÓNICA	98
B.2.1. ANÁLISIS DEL ALGORITMO DE LECTURA DEL ENCODER.....	98
B.2.2 CONFIGURACIÓN Y ESCALAMIENTO DE ENCODER DEL ROTRADI... 99	99
APÉNDICE C: PRUEBAS DE PLATAFORMA EXPERIMENTAL DE UN PÉNDULO DE TRANSMISIÓN DIRECTA.....	103
C.1. OBTENCIÓN DE PARÁMETROS DE UN ROBOT DE 1 GRADO DE LIBERTAD (PÉNDULO).....	103

	8
C.2. PROGRAMA DE CONTROL.....	105
C.3. PRUEBAS.....	107
APÉNDICE D: PROGRAMA DE CONTROL EN ARDUINO.....	117
APÉNDICE E: PONENCIAS EN CONGRESOS Y ARTÍCULOS.	125
BIBLIOGRAFÍA.....	131

INTRODUCCIÓN

La robótica es un área de investigación y desarrollo de sistemas mecánicos conocidos como robots, que de acuerdo con el Robot Institute of America (RIA) [1]: un robot es un manipulador multifuncional reprogramable, diseñado para mover materiales, partes, herramientas o dispositivos especializados a través de movimientos programados para la ejecución de una variedad de tareas. Esta definición sigue siendo válida actualmente a pesar de que fue dada en 1980.

Un robot de transmisión directa se define como un brazo mecánico donde todas sus articulaciones son manejadas con transmisión directa [2]; a su vez la transmisión directa consiste en que el rotor del motor esté directamente acoplado al eslabón, sin necesitar de sistemas de transmisión mecánicos, como engranes.

El primer robot de transmisión directa fue desarrollado en 1981 por Asada y Kanade en el Instituto de Robótica de la Universidad de Carnegie-Mellon [3], en la Benemérita Universidad Autónoma de Puebla se construyó el primer ROTRADI en la Facultad de Ciencias de la Electrónica en 1998 por F. Reyes Cortés [4]. Actualmente el Laboratorio de Robótica (FCE-BUAP) cuenta con seis robots experimentales ROTRADIS's, los cuales sirven de equipos de estudio y base experimental para los alumnos de licenciatura y maestría de esta facultad.

Uno de los principales problemas en el control de robots manipuladores es el lograr que el efector final llegue a una posición deseada. Para la mayoría de las metodologías de control es necesario conocer el modelo dinámico del robot para realizar el diseño del controlador. Aunque la dinámica del robot puede ser estudiada desde diferentes principios como las leyes de Newton, el principio de D'Alembert, las ecuaciones de Lagrange, y las ecuaciones y principios de Hamilton, tradicionalmente en robótica se emplean dos tipos de formulaciones para resolver este problema: la formulación de Lagrange, que permite obtener las ecuaciones de movimiento de cada articulación a partir de la energía potencial y cinética del sistema; la formulación de Newton-Euler, que se fundamenta en efectuar el balance de fuerzas y pares existentes [5]. Por otro lado, el robot es de tres grados de libertad y con la formulación de Newton se complica el problema al aumentar los grados de libertad.

Por otra parte, el diseño de nuevos algoritmos de control es una de las actividades de investigación en la que constantemente se está trabajando dentro del área de robótica, el fin es mejorar el desempeño de controladores experimentalmente aprobados. Es bien sabido que la mayoría de los manipuladores industriales que se usan actualmente están equipados con controladores simples, tales como el Proporcional-Derivativo (PD, por sus siglas en inglés) y el Proporcional-Integral-Derivativo (PID, por sus siglas en inglés), los cuales son efectivos para alcanzar el propósito de posicionamiento del efector final en una tarea específica [6].

El control PD es más ampliamente usado como estrategia para robots manipuladores, esto debido a su simplicidad. El control PID es otra estrategia popular, sin embargo, carece de una prueba de estabilidad asintótica global [6]. En contraste, el control PD simple con compensación gravitatoria propuesto en la literatura puede ser considerado como un hito en control de robots, ya que permite una estabilidad asintótica global para el sistema de lazo cerrado, con una selección trivial de ganancias proporcionales y derivativas, lo que demostró que era efectivo para el control posicional [7].

En años recientes, diversos esquemas de control tipo PD han sido desarrollados para control posicional de robots manipuladores [6, 8]. La estrategia general para un manipulador es controlar su movimiento y además dar una respuesta de disturbio para desviaciones de ese movimiento que tienen la forma de una impedancia. La interacción puede ser regulada, modulada y controlada cambiando la impedancia, así que a la aproximación propuesta se le llama control de impedancia [9].

El control de fuerza por impedancia mecánica [8], también denominado control de fuerza implícito, es una estrategia de control adecuada tanto para el movimiento libre como para el restringido. La impedancia mecánica relaciona la fuerza que actúa sobre un cuerpo con su movimiento. Se considera que el cuerpo se opone a la fuerza externa, fuerzas de inercia, amortiguamiento y rigidez [10]. El propósito del control de impedancia es imponerle las características deseadas, es decir, la masa, el amortiguamiento y la rigidez.

La principal ventaja del control por impedancia reside en el hecho de que el mismo controlador puede ser utilizado tanto para la fase de movimiento libre como la de movimiento restringido [10]. Esto abre la posibilidad de aplicar dicho control en tareas en las que un ROTRADI interactúa con diferentes medios externos, ya sean con una rigidez constante o aquellos que presentan una variación de esta dependiente del tiempo.

En este trabajo de tesis se trabaja con un robot manipulador antropomorfo de tres grados de libertad de transmisión directa, por tal motivo se presenta el diseño integral. Se estudia el modelo dinámico de un ROTRADI de tres grados de libertad, considerando el ancho de los motores. Además, se propone el estudio y validación de diversos algoritmos de

control de impedancia aplicados a un robot manipulador antropomorfo de 3 grados de libertad de transmisión directa usando una tarjeta de adquisición de datos Arduino [11, 12]. Para ello se toma como partida un algoritmo de control ya propuesto donde se implementa un control PD y un sensor de fuerza/torque que nos permita realizar tareas específicas donde es indispensable conocer la interacción robot-entorno. Adicionalmente, se propone desarrollar la interfaz electrónica y de programación para la puesta en operación de la plataforma experimental, sensor de fuerza y torque [13-15] en el robot manipulador de 3 grados de libertad, implementando una tarjeta Arduino [16] y el software de programación MATLAB [17, 18].

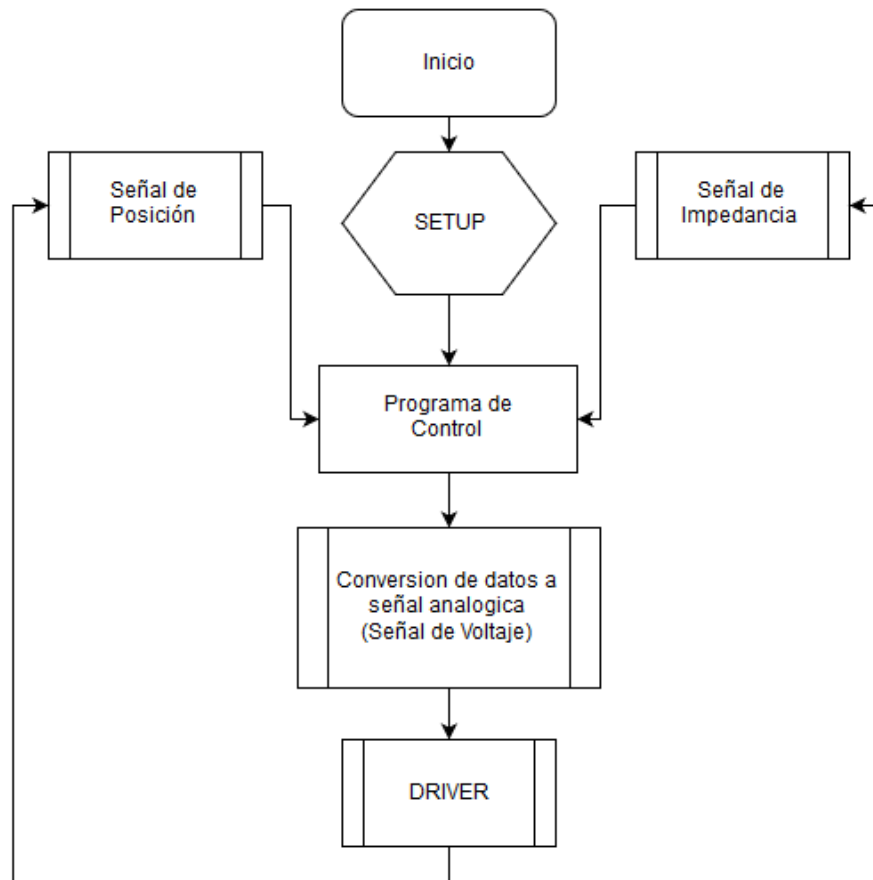


Figura 1: Diagrama de flujo de la propuesta del programa de control.

Este documento está estructurado de la siguiente manera, en el Capítulo 1 se incluye la mayor parte de las definiciones, teoría y trabajos realizados previamente con el objetivo de un fácil entendimiento del contenido de la tesis. En el Capítulo 2 se presenta el estudio del sistema dinámico de un robot de 3 grados de libertad en donde se obtienen los parámetros necesarios para los esquemas de control. En el Capítulo 3 se presenta el análisis de las señales de entrada, así como la forma de control de cada actuador y la instrumentación electrónica necesaria. En el Capítulo 4 se presenta la implementación de los algoritmos de control básicos

y necesarios para llegar al esquema de control deseado para los objetivos de la tesis. En el Capítulo 5 se muestra un análisis de los resultados obtenidos de cada estructura de control junto con los datos pertinentes ilustrados por diversas gráficas, así como tablas de los parámetros ocupados para dichas estructuras de control. Finalmente en se muestran las conclusiones pertinentes en donde se expresan tanto los aciertos como las fallas y limitaciones del sistema implementado para el desarrollo de la tesis.

OBJETIVOS

Los objetivos planteados para el presente trabajo de tesis se describen a continuación.

Objetivo general:

Evaluar experimentalmente algoritmos de control de impedancia en un robot manipulador antropomorfo de 3 grados de libertad.

Objetivos específicos:

1. Revisar bibliografía de algoritmos de control de impedancia en robots manipuladores de transmisión directa.
2. Estudiar el modelo dinámico de un robot antropomórfico de tres grados de libertad.
3. Estudiar la impedancia en robots manipuladores y los diferentes algoritmos de control de impedancia propuestos en la literatura.
4. Desarrollar la interfaz electrónica y programación para la puesta en operación de la plataforma experimental, sensor de fuerza y torque en un robot manipulador de 3 grados de libertad, implementando una tarjeta Arduino y MATLAB.
5. Evaluar experimentalmente al menos tres esquemas de control de impedancia viables en un robot manipulador de tres grados de libertad.
6. Realizar una aplicación robótica de control de impedancia en fisioterapia.

CAPÍTULO 1: ANTECEDENTES

En este capítulo se muestra una serie de conceptos básicos previamente revisados cuyo objetivo es principalmente el entendimiento de los componentes tanto del robot manipulador, como de la teoría requerida para implementación de los diferentes esquemas de control propuestos para el trabajo de tesis, principalmente la metodología necesaria que nos llevara a la obtención de los parámetros del modelo dinámico correspondientes al robot de transmisión directa que será empleado.

1.1 PARTE MECÁNICA

Los robots manipuladores son esencialmente brazos articulados; en forma más precisa es una cadena cinemática abierta formada por un conjunto de eslabones o elementos de la cadena interrelacionados mediante juntas. Las articulaciones permiten el movimiento relativo entre los sucesivos eslabones. A continuación se describen algunos conceptos básicos necesarios para comprender el diseño de un robot antropomorfo.

* **Eslabón**, es un cuerpo rígido que posee por lo menos dos nodos, que son los puntos de unión con otros elementos.

* **Junta**, es una conexión entre dos o más eslabones, la cual permite algún movimiento entre los eslabones conectados.

* **Cadena cinemática**, es un ensamblaje de eslabones y juntas interconectadas, de modo que proporcionen un movimiento de salida controlado en respuesta a un movimiento de entrada proporcionado. Las cadenas cinemáticas pueden ser abiertas o cerradas. Un mecanismo abierto con más de un eslabón tendrá siempre más de un grado de libertad y con esto necesitara tantos actuadores como grados de libertad. En la Figura 1.1 se presentan ambos mecanismos, donde se puede identificar los eslabones y las juntas.

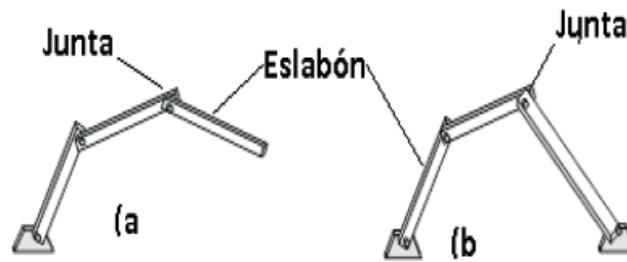


Figura 1.1: (a) cadena cinemática abierta y (b) cadena cinemática cerrada.

* **Grados de libertad (g.d.l):** según Robert L. Norton, son igual al número de parámetros independientes (medidas) que se necesitan para definir unívocamente su posición en el espacio en cualquier instante. Sin embargo, para el caso del diseño de un brazo mecánico, la definición más adecuada es la dada por Mark W. Spong: “el número de grados de libertad está determinado por el número de uniones de un manipulador”.

Los tipos de movimiento que se pueden tener en un cuerpo rígido son los que se describen a continuación:

1. **Traslación pura:** todos los puntos en el cuerpo describen trayectorias paralelas.

Así, una línea de referencia trazada en el cuerpo cambia su posición lineal pero no su orientación angular.

2. **Rotación pura:** el cuerpo posee un punto o centro de rotación, que no tiene movimiento con respecto un marco de referencia estacionario. Todos los demás puntos del cuerpo describen arcos respecto a ese centro. Esto es, una línea de referencia trazada en el cuerpo, y que pasa por su centro, cambia únicamente su orientación angular.

3. **Movimiento complejo:** es una combinación simultánea de rotación y traslación, cualquier línea de referencia trazada en el cuerpo cambiará su posición lineal y su orientación angular.

***Eector final:** un eector final es un dispositivo que se une a la muñeca del brazo del robot y activa el robot con el propósito general de realizar una tarea específica. La mayoría de las máquinas de producción requieren artefactos especiales y herramientas diseñadas para una operación particular, y un robot no es la excepción. El eector final es parte de las herramientas de uso especial para un robot.

Existe una amplia gama de efectores finales necesarios para realizar una gran variedad de funciones de trabajo diferentes. Estos tipos se pueden dividir en dos categorías principales: pinzas y herramientas.

***Volumen de trabajo:** El espacio o volumen de trabajo, está formado por todas las posiciones del espacio potencialmente accesibles por el extremo del robot. Este está

determinado por el tipo de configuración de cada robot, por ejemplo en la figura 1.2 se observa que el volumen de trabajo que forma un robot cartesiano es cubica, mientras que el cilíndrico ocupa el volumen de un cilindro, el esférico presenta un espacio de trabajo irregular y finalmente el robot articulado tiene un volumen cilíndrico. Una vez realizada una breve descripción de los elementos y características de la estructura mecánica del robot

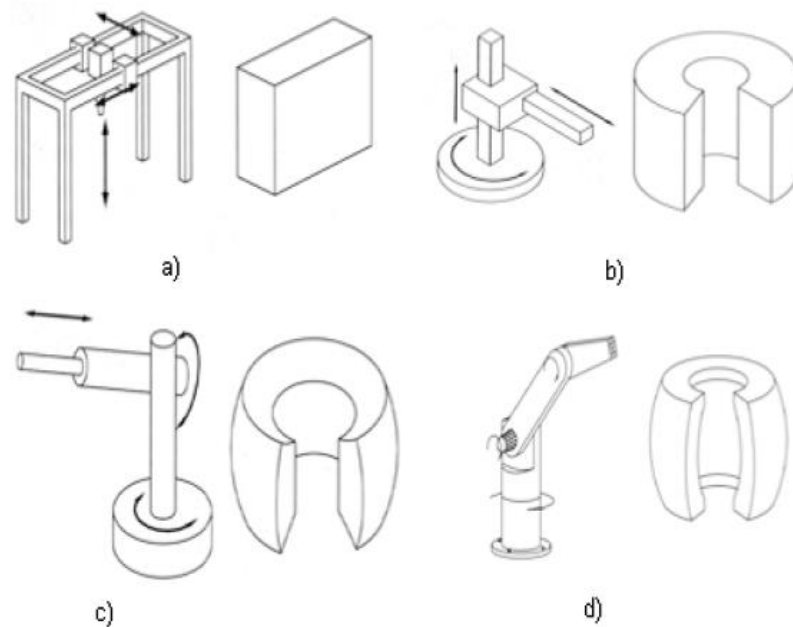


Figura 1.2: Volumen de trabajo para: a) robot cartesiano, b) robot cilíndrico, c) robot esférico, y d) robot articular [3].

1.2 COMPONENTES DE LA PLANTA

Esta parte tiene como objetivo la descripción de los dispositivos de mando y control que integran el sistema electrónico de control de posición y velocidad de un brazo robótico de tres grados de libertad, como el que se muestra en el diagrama a bloques de la figura 1.3. Los elementos son: 1) computadora, 2) interfaz de control, 3) amplificador de potencia (Drivers) y 4) motor. A continuación se describen cada uno de ellos.

1) COMPUTADORA

En la PC se lleva a cabo la programación de los algoritmos de control, y se graban las variables del robot como velocidad, posición y par o torque. Este dispositivo sirve adicionalmente para el monitoreo y revisión general del comportamiento del sistema, permitiendo el control del tipo de función que debe realizarse, por ejemplo control de

trayectoria angular, control de posición cartesiano, control de fuerza o control de impedancia entre otros.

2) INTERFAZ DE CONTROL

Para gestionar la transferencia de datos entre los aplicadores de potencia y el programa de control es necesaria una tarjeta de control y la electrónica necesaria para que señales digitales de control provenientes sean correctamente interpretadas por los drivers de cada motor a controlar, además esta interfaz debe recibir las señales de los codificadores de posición, por lo que esta tarjeta debe contar con entradas y salidas, digitales y analógicas o al menos una forma para realzar tal conversión. El decodificador de posición es un dispositivo que convierte movimiento en pulsos digitales. Este puede dar la posición relativa por medio de una secuencia de bits o la posición absoluta por medio de bits codificados, por ejemplo el código Grey. Puede tener configuración lineal o rotatoria, aunque la más común es la rotatoria.

3) AMPLIFICADOR DE POTENCIA (DRIVER)

Los amplificadores se encargan de recibir las señales de retroalimentación de los decodificadores de posición (*encoder*) integrados en el interior de cada motor, así como de proporcionar las señales de potencia necesarias para excitar las fases del motor, y de esta manera provocar un movimiento que se refleje en el cambio de posición del robot.

4) MOTORES

El motor es el actuador que proporciona movimiento rotacional a la articulación; los motores pueden ser de corriente directa o corriente alterna, según las necesidades de torque y velocidad, además pueden incluir un decodificador de posición interna, el cual proporciona señales de retroalimentación para conocer la posición y velocidad de cada articulación en tiempo real.

Para un ROTRADI, los actuadores utilizados son motores de transmisión directa, cuya característica principal consiste en que el rotor del motor esté directamente acoplado al eslabón sin necesitar de sistemas de transmisión mecánicos, como engranes [6].

5) DECODIFICADOR DE POSICIÓN

El decodificador de posición es un dispositivo que convierte movimiento en pulsos digitales. Este puede dar la posición relativa por medio de una secuencia de bits o la posición absoluta por medio de bits codificados, por ejemplo el código Grey. Puede tener configuración lineal

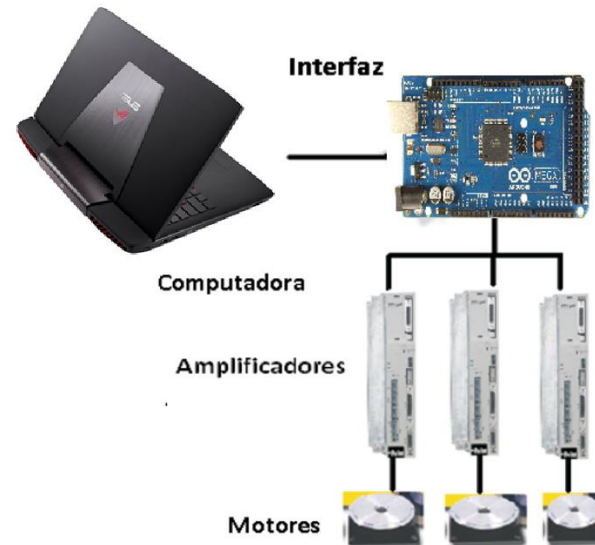


Figura 1.3: Elementos que integran la instrumentación electrónica: computadora, interfaz, amplificadores y motores.

o rotatoria, aunque la más común es la rotatoria. En la figura 1.4 se muestra la construcción interna de un *encoder*, se observa que tiene dos leds emisores y dos optoreceptores, la señal de salida cuadrada es la secuencia de bits los cuales están desfasando por un cuarto del periodo total.

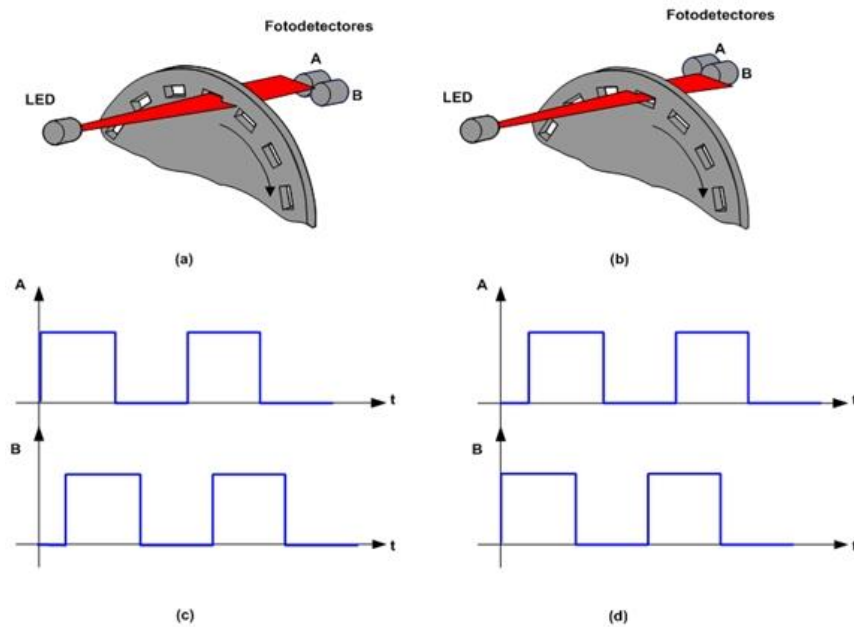


Figura 1.4 Construcción interna de un *encoder* óptico donde se observa la señal de salida A y la señal de salida B, desfasadas por un cuarto de la señal.

1.3 DEFINICIÓN DE IMPEDANCIA MECÁNICA

El control de fuerza por *impedancia mecánica* activa [9]¹, también denominado implícito, es una estrategia de control adecuada tanto para el movimiento libre como para el restringido. La impedancia mecánica relaciona la fuerza que actúa sobre un cuerpo con su movimiento. Se considera que el cuerpo opone a la fuerza externa fuerzas de inercia, amortiguamiento y rigidez. La definición de impedancia mecánica es:

$$Z(s) = \frac{F(s)}{v(s)} = \frac{Ms^2X + BsX + KX}{sX} = Ms + B + \frac{K}{s}, \quad (1.1)$$

donde Z es la impedancia, F la fuerza, v la velocidad, X la posición, M la masa, B el amortiguamiento y K la rigidez del sistema.

El propósito del control por impedancia es imponerle al sistema las características de masa, amortiguamiento y rigidez deseadas.

Durante el movimiento restringido, el objetivo del control por impedancia no es seguir una referencia de fuerza o de posición, sino hacer que el proceso tenga el comportamiento dinámico deseado. Por esta razón se llama también control implícito de fuerza, dado que esta no se controla directamente.

La principal ventaja de esta estrategia de control reside en el hecho de que el mismo regulador puede ser utilizado tanto para la fase de movimiento libre como la de movimiento restringido. Esto es posible gracias al hecho de que si la fuerza de interacción es nula (en movimiento libre), los reguladores de impedancia activa se reducen a controladores por posición. De hecho, la forma más sencilla de implementar el control implícito es como control por posición puro.

Sin embargo, hay que tener en cuenta la principal limitación del control implícito, que es el hecho de que no garantice que se va alcanzar la referencia de fuerza. Por ello el control por impedancia ha sido utilizado en tareas de montaje que no requieren un control preciso de fuerza. El ejemplo más típico es la inserción de un objeto en un agujero.

También se utiliza para el control de los robots teleoperados que entran en contacto con el entorno. Estas aplicaciones son adecuadas para el control por impedancia puesto que requieren, más que un seguimiento de la referencia de fuerza, un determinado comportamiento del sistema [6,8-10].

¹ N. Hogan, *Impedance control: an approach to manipulation: part I-theory, part II-Implementacion and part III-applications*, ASME J Dyn, Syst. Meas, Vol. 107, pp. 1-24 (1985).

1.4 MODELADO DINÁMICO DEL ROBOT

La descripción cinemática del efector final de un robot, puede ser realizado en el espacio articular o en el espacio de trabajo. En el primer caso, \mathbf{q} representa las coordenadas articulares del manipulador en el espacio de trabajo, \mathbf{x} representa la postura entre el efector final y el marco de coordenadas cartesianas revisar [8,9].

El mapeo de la cinemática directa $\mathbf{f}: R^n \rightarrow R^m$, relaciona la posición articular del vector $\mathbf{q} \in R^n$ con el vector posición de trabajo $\mathbf{x} \in R^m$, y es dado por:

$$\mathbf{x} = \mathbf{f}(\mathbf{q}). \quad (1.2)$$

El número de grados de libertad del robot manipulador es representado por n mientras que m es la dimensión del espacio de trabajo. Las primeras y segundas derivadas de la ecuación (1.2), corresponden a la relación diferencial entre las velocidades y las aceleraciones respectivamente, de las articulaciones y el espacio de tareas:

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad (1.3)$$

$$\ddot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}}, \quad (1.4)$$

donde $\mathbf{J}(\mathbf{q}) = \partial \mathbf{f} / \partial \mathbf{q} \in R^{m \times n}$ es la matriz Jacobiana del robot manipulador y su derivada con respecto al tiempo es $\dot{\mathbf{J}}(\mathbf{q}) = d\mathbf{J}/dt \in R^{m \times n}$.

La interacción de tareas requiere la relación entre los torques articulares y las fuerzas externas aplicadas correspondientes. En la representación del espacio articular, el torque articular está dado por $\boldsymbol{\tau}$. Por otra parte, en el espacio de tareas, el vector \mathbf{F} consiste en componentes de fuerza que operan en el efector final respecto a un sistema de coordenadas cartesiano fijo. Basado en el principio del trabajo virtual, los torques aplicados $\boldsymbol{\tau} \in R^n$, necesarios para generar una fuerza $\mathbf{F} \in R^m$ en el efector final, están dados por la siguiente relación [7]:

$$\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q})\mathbf{F}. \quad (1.5)$$

El espacio dinámico articular de un robot manipulador de n -eslabones que interactúa con el medio, puede ser escrita de la siguiente forma:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}_r(\dot{\mathbf{q}}) = \boldsymbol{\tau} - \mathbf{J}^T(\mathbf{q})\mathbf{F}_e, \quad (1.6)$$

donde $\mathbf{M}(\mathbf{q}) \in R^{n \times n}$ es la matriz de inercia, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in R^{n \times n}$ es la matriz de Coriolis y aceleración centrípeta, $\mathbf{g}(\mathbf{q}) \in R^n$ es el vector de fuerzas gravitacionales, $\mathbf{f}_r(\dot{\mathbf{q}}) \in R^n$ es el vector de fricción viscosa y $\mathbf{F}_e \in R^m$ denota el vector de fuerzas de contacto [7].

1.5 FORMULACIONES PARA LA IMPEDANCIA MECÁNICA

El enfoque de control de la impedancia propuesto por Hogan [8], tiene ahora varias interpretaciones. Tal como se presenta en la referencia [7], el objetivo de control de la impedancia es que el efector final del robot responda a fuerzas aplicadas externamente, de acuerdo a una dinámica bien definida. Más específicamente, la relación deseada entre el efector final de la posición $\mathbf{x} \in R^m$, la referencia de trayectoria del efector final $\mathbf{x}_d \in R^m$, y el vector de fuerzas de contacto $\mathbf{F}_e \in R^m$ es especificada a continuación:

$$\mathbf{F}_e = \mathbf{k}_d(\mathbf{x}_d - \mathbf{x}) + \mathbf{B}_d(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{M}_d(\ddot{\mathbf{x}}_d + \ddot{\mathbf{x}}), \quad (1.7)$$

donde $\mathbf{k}_d \in R^{m \times m}$ es la matriz de rigidez, $\mathbf{B}_d \in R^{m \times m}$ es la matriz de amortiguamiento y $\mathbf{M}_d \in R^{m \times m}$ es la matriz de inercia. De la ecuación (1.7) podemos obtener la impedancia mecánica $\mathbf{Z}_d \in R^{m \times m}$ dada en el dominio de la frecuencia por:

$$\mathbf{Z}_d = \mathbf{K}_d + \mathbf{B}_d s + \mathbf{M}_d s^2. \quad (1.8)$$

Los parámetros de impedancia $\mathbf{K}_d = \text{diag}\{k_d\}$, $\mathbf{B}_d = \text{diag}\{b_d\}$ y $\mathbf{M}_d = \text{diag}\{m_d\}$ pueden ser seleccionados de acuerdo a

$$\omega_n = \sqrt{\frac{k_d + k_e}{m_d}}, \quad (1.9)$$

$$\zeta = \frac{b_d}{2\sqrt{m_d(k_d + k_e)}}, \quad (1.10)$$

donde la frecuencia natural ω_n y el factor de amortiguamiento ζ determinan el comportamiento dinámico durante la fase de transición entre el movimiento libre y la interacción. La rigidez del entorno $\mathbf{K}_e = \text{diag}\{k_e\}$ se selecciona de acuerdo con el tamaño y la capacidad del torque del robot.

1.6 ROTRADI VI

Un ROTRADI (Robot de transmisión directa) es una plataforma experimental que se mueve en su espacio tridimensional equipado con tarjetas electrónicas y programado para realizar aplicaciones de automatización. Estas plataformas poseen características que pueden competir en desempeño y versatilidad con las mejores marcas mundiales de robótica. No obstante su principal característica es el hecho de tener arquitectura abierta, esto significa que se puede evaluar experimentalmente cualquier estrategia de control, en contraste con los robots industriales con arquitectura cerrada que no permiten programar nuevos esquemas de control y quedan confinados a lo que permita realizar su sistema operativo [6], en la Figura 1.5 se muestra la plataforma ROTRADI VI que será empleada en esta tesis.



Figura 1.5: Diseño en “Solidworks” del ROTRADI VI a lado de una fotografía del mismo.

1.6.1 CARACTERISTICAS DEL ROTRADI VI

La plataforma de prueba consta principalmente de tres elementos:

- a) El robot manipulador, en donde se contempla el sistema mecánico a controlar y la interfaz electrónica que permitirá el control del robot.
- b) El controlador, el algoritmo de control que gobernará el comportamiento del robot.
- c) El sensor, los *encoder* que permitirán conocer el posicionamiento del robot.

Para propósitos de la tesis, se propone un diagrama a bloques ilustrado en las Figuras 1.6 y 1.7 en donde se contemplan los componentes del ROTRADI VI.

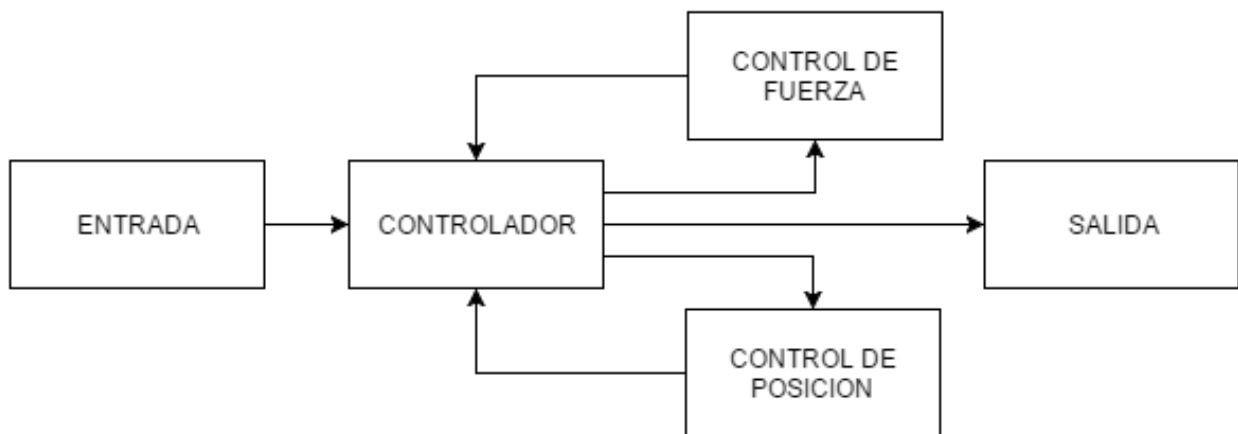


Figura 1.6: Diagrama a bloques de la propuesta, en donde los bloques de control de fuerza y posición se muestran en la figura 2.

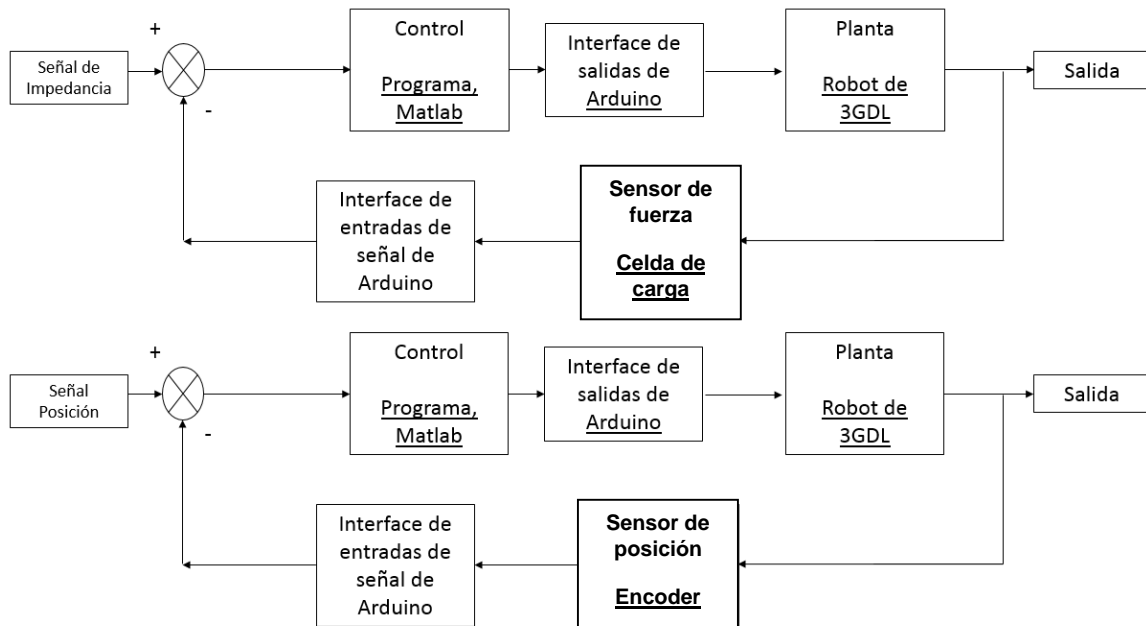


Figura 1.7: Diagramas a bloques del planteamiento de tesis, el primero correspondiente al control de impedancia y el segundo al control de posición.

Entre las características del robot se tiene que cada uno de los actuadores son motores de transmisión directa cuyas características se presentan a continuación:

Hombro: DM1A-050G con un par máximo de 50 Nm y un *encoder* con una resolución máxima de 2,261,440 pulsos por revolución.

Codo: DM1C-004F con un par máximo de 4 Nm y un codificador con una resolución máxima de 4,096,000 pulsos por revolución.

Base: DM1B-015G con un par máximo de 15 Nm y un codificador con una resolución máxima de 4,096,000 impulsos por revolución.

Estos motores son controlados por drivers DrvG3 que permiten la programación de los diversos parámetros de funcionamiento del motor en función del modo que se requiere, tales como control de la posición, la velocidad o el torque [24].

Finalmente se muestra en la Tabla 1.1 las características de cada eslabon del ROTRADI VI los cuales seran empleados mas adelante para propositos de control.

TABLA 1.1. TABLA DE CARACTERISTICAS DEL ROTRADI VI.

Eslabón	Variable	Valor
1 (BASE)	m_1	0
	l_1	1.19 m
	l_{c1}	0
2 (HOMBRO)	m_2	8.43 kg
	l_2	0.35 m
	l_{c2}	0.269 m
3 (CODO)	m_3	1.2 kg
	l_3	0.35 m
	l_{c3}	0.051 m

1.6.2 CONFIGURACION

Para la implementación de algoritmos de control, se configura la planta en modo torque y adicionalmente se programa el driver para que el *encoder* funcione en modo diferencial con señales A-B en cuadratura desfasadas 90 grados y con una resolución máxima de cada *encoder* a 10000 pulsos por vuelta, como se muestra en la Figura 1.6. Esta configuración permite conocer la dirección de movimiento de los actuadores y la combinación misma de las señales permite el uso de un recurso específico del microcontrolador conocido como ISR (Interruption Service Rutine).

**Para mayor información sobre la configuración del sistema, revisar el Apéndice B.

1.7 MICROCONTROLADOR ARDUINO

Arduino es un sistema de programación con capacidad de admitir múltiples tipos de lenguaje de programación, principalmente C/C++, con plataformas electrónicas que utilizan microcontroladores para formar sistemas empotrados especializados en la automatización de procesos, con un amplio espectro de aplicaciones en ciencias exactas y en ingeniería. Este tipo de plataformas se ha popularizado en todo el mundo, gracias al desempeño, costo y arquitectura abierta de sus modelos; esto último, es la característica principal que lo hace atractivo para desarrollar aplicaciones de automatización, ya que admite adaptabilidad para automatizar cualquier proceso físico. Otra de las ventajas del sistema Arduino, es que tiene software libre (gratuito), así como el acceso y adecuación del código fuente y de la electrónica de sus tarjetas.

El hardware generalmente consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168,

Atmega328, Atmega1280, ATmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado en la placa.

Las placas Arduino pueden tomar información del entorno a través de sus entradas analógicas y digitales, puede controlar luces, motores y otros actuadores, así como la lectura de diversos sistemas sensoriales para obtener conocimiento del entorno. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un computador [21].

1.7.1 ARDUINO MEGA 2560

El Arduino Mega 2560 es una placa electrónica basada en el Atmega2560. Cuenta con 54 pines digitales de entrada / salida (de los cuales 15 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UARTs (puertas seriales), un oscilador de 16MHz, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar al microcontrolador; basta con conectarlo a un ordenador con un cable USB o el poder con un adaptador de CA o la batería a CC para empezar. La tarjeta Arduino Mega es compatible con la mayoría de los escudos diseñados para el Arduino Duemilanove o Diecimila [22].

El Mega2560 difiere de todas las placas anteriores en que no utiliza el chip controlador de USB a serial FTDI. En lugar de ello, cuenta con la ATmega16U2 (ATmega8U2 en las juntas de revisión 1 y revisión 2) programado como convertidor USB a serie [22].

Revisión 2 de la junta Mega2560 tiene una resistencia tirando de la línea 8U2 HWB a tierra, por lo que es más fácil de poner en modo DFU [22].

Revisión 3 de la placa tiene las siguientes características nuevas:

pinout: SDA añadido y pines SCL que están cerca al pin AREF y otros dos nuevos pasadores colocados cerca del pin RESET, la instrucción IOREF que permiten a los escudos para adaptarse a la tensión suministrado desde la pizarra. En el futuro, escudos serán compatibles tanto con la placa que utilizan el AVR, que operan con 5V y con el Arduino Debido que operan con 3.3V. El segundo es un pin no está conectado, que se reserva para usos futuros [22].

En la Figura 1.8 se muestra un ejemplar de la tarjeta Arduino Mega 2560, mientras que en la Tabla 1.2 se muestran las características de la misma.

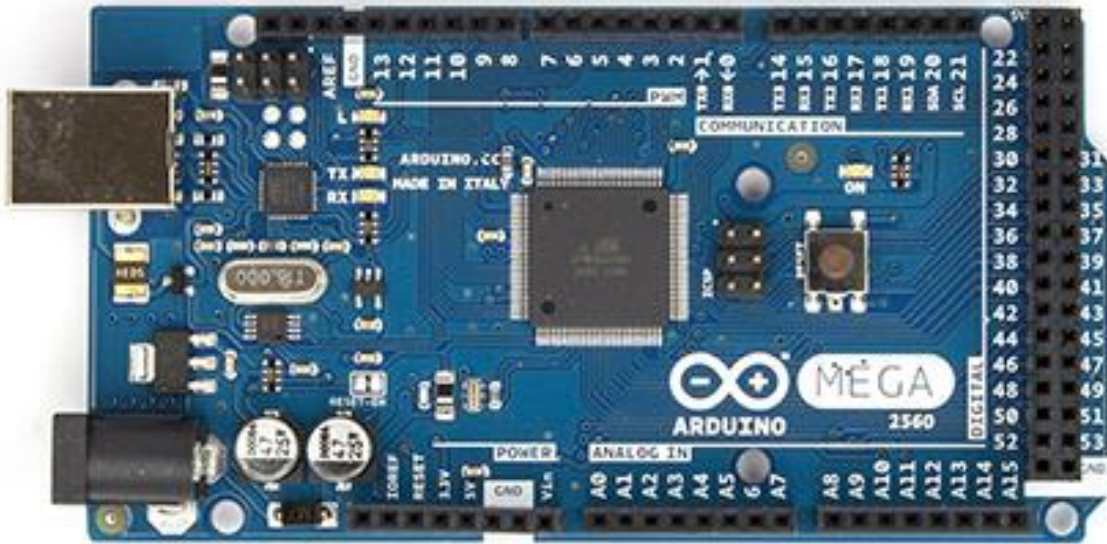


Figura 1.8: Ejemplar de una tarjeta Arduino Mega.

TABLA 1.2: TABLA DE LAS CARACTERÍSTICAS DE LA TARJETA ARDUINO ATMEGA2560 [22].

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 kB of which 8 kB used by bootloader
SRAM	8 kB
EEPROM	4 kB
Clock Speed	16 MHz

Tomando en cuenta estas características, el modelo de Arduino Mega R3 es una opción viable para el control de ROTRADI VI, pues pese al hecho de que no posee salidas de voltaje analógico, es posible utilizar conjuntos de sus salidas digitales para generar un valor binario

el cual a través de la electrónica pertinente será posible controlar los niveles de voltaje analógico requerido para los drivers del sistema.

1.8 SENSOR DE FUERZA/PAR

Existen diversos tipos de sensores de fuerza, por lo general celdas de par y carga, también conocidos como transductores de fuerza, son dispositivos útiles para la medición de pares y fuerzas generadas dentro de un sistema mecánico. La medición de una fuerza o par se reduce usualmente a medir una tensión inducida por la fuerza o par aplicado a un elemento extensible. Por tanto, una medición indirecta de la fuerza se obtiene a través de la medición de pequeños desplazamientos. El componente básico de un sensor de fuerza es un indicador de tensión, que es una pequeña espira plana de alambre conductor que cambia su resistencia cuando se ejerce una fuerza sobre ella. Este indicador se coloca sobre una placa y se mide el cambio en la resistencia para determinar la tensión, debido a que esta está directamente relacionada con la fuerza aplicada sobre la placa. El indicador de tensión se elige de tal forma que su resistencia R_X varíe linealmente en un rango de tensión admisible para el elemento extensible [6], En la Figura 1.9 se muestra el arreglo clásico de un puente de Wheatstone.

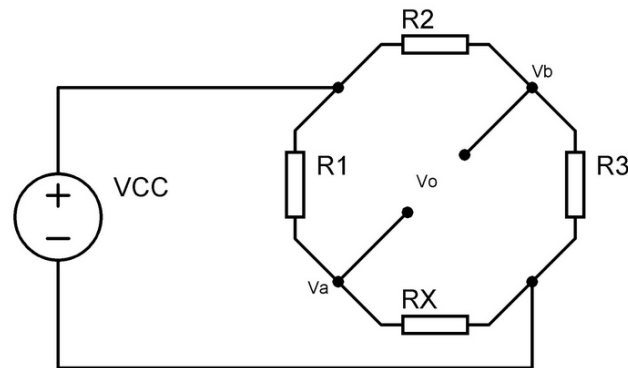


Figura 1.9: Puente de Wheatstone para el acondicionamiento de una señal.

Un indicador típico de tensión tiene un cambio de resistencia muy bajo cuando se ejerce fuerza sobre él. Los indicadores dentro del transductor de fuerza, usualmente un múltiplo de cuatro, son conectados dentro de una configuración de puente de Wheatstone, como se muestra en la Figura 1.9, con la finalidad de convertir la pequeña señal de cambio en resistencia en una señal eléctrica útil. Además, algunos componentes pasivos como resistencias y cables dependientes de la temperatura, son utilizados para compensar y calibrar la señal de salida del puente. La salida de voltaje es negativa para una determinada dirección de la fuerza, y positiva para la dirección opuesta. El balance de voltaje en el puente

esta descrito por la siguiente expresión, nótese que las variables corresponden al arreglo de la Figura 1.9:

$$v_o = \left(\frac{R_x}{R_1 + R_x} - \frac{R_3}{R_2 + R_3} \right) v_{cc}.$$

Generalmente los sensores de fuerza utilizados en robots manipuladores son colocados en la muñeca, agregando algún elemento sobre el sensor que actúa como efector final durante la interacción con el entorno. Los rangos de medición de fuerza oscilan entre 50 y 500 N, mientras que para la medición de pares están entre 7 y 70 Nm. La resolución típica es del orden de 0.1% de la fuerza máxima y de 0.05% del par máximo, respectivamente. En cuanto al procesamiento de las señales medidas y la generación de salida, se tiene una frecuencia de muestreo del orden de 1 kHz [6].

Aplicaciones de sensores de fuerza:

- Verificación de ensamble de componentes mecánicos y eléctricos.
- Procesos como desbastado, pulido y esmerilado.
- Retroalimentación de fuerza para robots en ambientes nucleares.
- Investigación sobre rehabilitación física y prótesis.

1.8 MODELO DINÁMICO DE UNA CENTRÍFUGA

A continuación se describe el procedimiento o los pasos para obtener el modelo dinámico de una centrífuga, cuyos datos serán obtenidos tomando como base la Figura 1.10 descrita a continuación.

En términos generales el sistema consta de un solo punto rotacional, el cual gira sobre el eje z, así mismo y a una altura definida, se encuentra un elemento fijo al mecanismo de rotación el cual genera el movimiento.

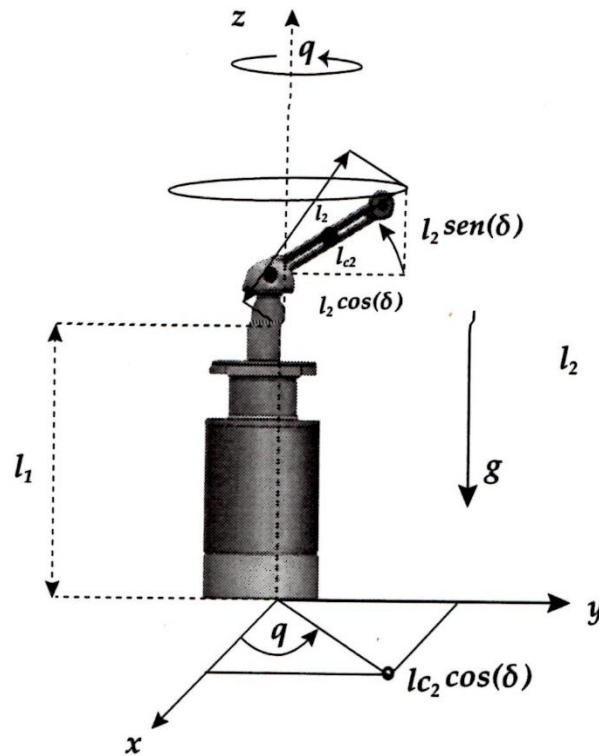


Figura 1.10: Centrífuga de referencia.

Parámetros del sistema centrífuga:

m : Masa del eslabón.

l_1 : Altura de la base.

l_2 : Longitud del eslabón.

l_{c2} : Longitud del centro de masa del eslabón.

I : Inercia del eslabón.

g : Aceleración de la gravedad.

b : Coeficiente de fricción viscosa.

Modelo cinemática directa

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = f(\delta, l_1, l_{c2}, q) = \begin{bmatrix} l_{c2} \cos(\delta) \cos(q) \\ l_{c2} \cos(\delta) \sin(q) \\ l_1 + l_{c2} \end{bmatrix} \text{sen}(\delta) . \quad (1.11)$$

Cinemática diferencial

$$\mathbf{v} = \frac{d}{dt} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -l_{c2} \cos(\delta) \sin(q) \\ l_{c2} \cos(\delta) \cos(q) \\ 0 \end{bmatrix} \dot{q} , \quad (1.12)$$

$$\mathbf{v}^T \mathbf{v} = (-l_{c2} \cos(\delta) \sin(q) \dot{q})^2 + (l_{c2} \cos(\delta) \cos(q) \dot{q})^2 = l_{c2}^2 \cos^2(\delta) \dot{q}^2.$$

Modelo de energía

$$K(q, \dot{q}) = \frac{1}{2} m \mathbf{v}^T \mathbf{v} + \frac{1}{2} I \dot{q}^2 = \frac{1}{2} [m l_{c2}^2 \cos^2(\delta) + I] \dot{q}^2, \quad (1.13)$$

$$U(q) = mgz = mg[l_1 + l_{c2} \sin(\delta)]. \quad (1.14)$$

El lagrangiano de la centrifuga está dado por:

$$\begin{aligned} L(q, \dot{q}) &= K(q, \dot{q}) - U(q) \\ &= \frac{1}{2} m \mathbf{v}^T \mathbf{v} + \frac{1}{2} I \dot{q}^2 \\ &= \frac{1}{2} [m l_{c2}^2 \cos^2(\delta) + I] \dot{q}^2 + mg[l_1 + l_{c2} \sin(\delta)]. \end{aligned} \quad (1.15)$$

Ecuaciones de movimiento de Euler Lagrange

$$\tau = \frac{d}{dt} \left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}} \right] - \left[\frac{\partial L(q, \dot{q})}{\partial q} \right] + b \dot{q}. \quad (1.16)$$

Tomando en cuenta el lagrangiano de la ecuación (1.15) se tiene que el modelo de la centrifuga incluyendo el fenómeno de fricción está dado por:

$$\tau = [m l_{c2}^2 \cos^2(\delta) + I] \ddot{q} + b \dot{q}. \quad (1.17)$$

Convirtiendo la ecuación diferencial a una ecuación diferencial de primer orden de la forma $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$

$$\frac{d}{dt} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ [m l_{c2}^2 \cos^2(\delta) + I]^{-1} [\tau - b \dot{q}] \end{bmatrix},$$

donde la variable de estado $\mathbf{x} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$.

1.9 MODELO DINÁMICO DE UN ROBOT DE 2 GDL (RR)

De manera similar se realiza el mismo procedimiento para el robot planar de 2 GDL, rotacional rotacional (RR), cuyos parámetros se especifican en la Figura 1.11.

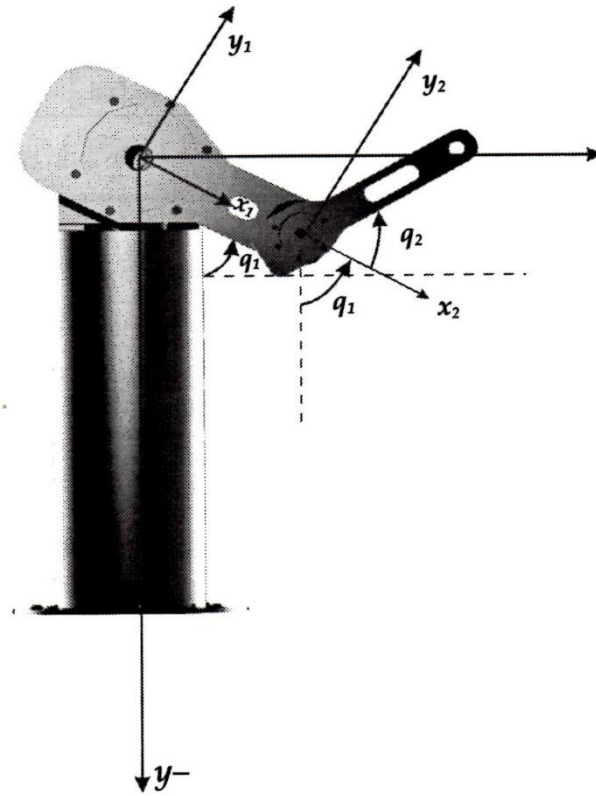


Figura 1.11: Brazo robótico planar de 2 GDL.

Parámetros del sistema brazo robótico planar de 2 GDL:

m_1 : Masa del eslabón 1.

m_2 : Masa del eslabón 2.

l_1 : Longitud del primer eslabón.

l_2 : Longitud del segundo eslabón.

l_{1c} : Longitud del centro de masa del primer eslabón.

l_{c2} : Longitud del centro de masa del segundo eslabón.

I_1 : Inercia del primer eslabón.

I_2 : Inercia del segundo eslabón.

g : Aceleración de la gravedad.

b_1 : Coeficiente de fricción viscosa.

b_2 : Coeficiente de fricción viscosa.

Cinemática directa.

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = f(l_{c1}, q_1) = \begin{bmatrix} l_{c1} \text{sen}(q_1) \\ -l_{c1} \text{cos}(q_1) \end{bmatrix} \quad (1.18)$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = f(l_1, l_2, q_1, q_2) = \begin{bmatrix} l_1 \text{sen}(q_1) + l_{c2} \text{sen}(q_1 + q_2) \\ -l_1 \text{cos}(q_1) - l_{c2} \text{cos}(q_1 + q_2) \end{bmatrix} \quad (1.19)$$

Cinemática diferencial.

$$\mathbf{v}_1 = \begin{bmatrix} -l_{c1} \text{cos}(q_1) \\ l_{c1} \text{sen}(q_1) \end{bmatrix} \dot{q}_1 \quad (1.20)$$

$$\mathbf{v}_2 = \begin{bmatrix} l_1 \text{cos}(q_1) + l_{c2} \text{cos}(q_1 + q_2) & l_{c2} \text{cos}(q_1 + q_2) \\ l_1 \text{sen}(q_1) + l_{c2} \text{sen}(q_1 + q_2) & l_{c2} \text{sen}(q_1 + q_2) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (1.21)$$

$$\mathbf{v}_1^T \mathbf{v}_1 = l_{c1}^2 \dot{q}_1^2$$

$$\mathbf{v}_2^T \mathbf{v}_2 = [l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \text{cos}(q_2)] \dot{q}_1^2 + l_{c2}^2 \dot{q}_2^2 + 2[l_1 l_{c2} \text{cos}(q_2) + l_{c2}^2] \dot{q}_1 \dot{q}_2$$

Modelo de energía.

$$\begin{aligned} K(q, \dot{q}) &= \frac{1}{2} m_1 \mathbf{v}_1^T \mathbf{v}_1 + \frac{1}{2} I_1 \dot{q}^2 + \frac{1}{2} m_2 \mathbf{v}_2^T \mathbf{v}_2 + \frac{1}{2} I_2 [\dot{q}_1^2 + \dot{q}_2^2] \\ &= \frac{1}{2} [m_1 l_{c1}^2 + I_1 + I_2 + m_2 l_1^2 + m_2 l_{c2}^2 + 2m_2 l_1 l_{c2} \text{cos}(q_2)] \dot{q}_1^2 + \frac{1}{2} [I_2 + m_2 l_{c2}^2] \dot{q}_2^2 \\ &\quad + [m_2 l_1 l_{c2} \text{cos}(q_2) + m_2 l_{c2}^2 + I_2] \dot{q}_1 \dot{q}_2 \end{aligned} \quad (1.22)$$

$$U(q) =$$

$$m_1 g l_{c1} [1 - \text{cos}(q_1)] + m_2 g [(l_1 + l_2) - (l_1 \text{cos}(q_1) + l_{c2} \text{cos}(q_1 + q_2))] \quad (1.23)$$

Ecuaciones de movimiento de Euler-Lagrange.

Los torques aplicados del robot de 2 GDL incluyendo el fenómeno de fricción:

$$\begin{aligned} \tau_1 &= [m_1 l_{c1}^2 + I_1 + I_2 + m_2 l_{c2}^2 + 2m_2 l_1 l_{c2} \text{cos}(q_2)] \ddot{q}_1 + [m_2 l_1 l_{c2} \text{cos}(q_2) + I_2 + m_2 l_{c2}^2] \ddot{q}_2 \\ &\quad + b_1 \dot{q}_1 - 2m_2 l_1 l_{c2} \text{sen}(q_2) \dot{q}_1 \dot{q}_2 - m_2 l_1 l_{c2} \text{sen}(q_2) \dot{q}_2^2 + m_1 g l_{c1} \text{sen}(q_1) \\ &\quad + m_2 g (l_1 \text{sen}(q_1) + l_{c2} \text{sen}(q_1 + q_2)), \end{aligned}$$

$$\begin{aligned} \tau_2 &= [m_2 l_{c2}^2 + I_2] \ddot{q}_2 + [m_2 l_1 l_{c2} \text{cos}(q_2) + m_2 l_{c2}^2 + I_2] \ddot{q}_1 + m_2 l_1 l_{c2} \text{sen}(q_2) \dot{q}_1^2 \\ &\quad + m_2 g l_{c2} \text{sen}(q_1 + q_2) + b_2 \dot{q}_2. \end{aligned}$$

Agrupando términos y expresándolos en la forma.

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

donde $\mathbf{B}^T = (b_1, b_2)$. Así, tenemos:

I. $\mathbf{M}(q) \in R^{2 \times 2}$, la matriz de inercia:

$$\mathbf{M}(q) = \begin{bmatrix} m_1 l_{c1}^2 + I_1 + I_2 + m_2 l_{c2}^2 + 2m_2 l_1 l_{c2} \text{cos}(q_2) & m_2 l_1 l_{c2} \text{cos}(q_2) + I_2 + m_2 l_{c2}^2 \\ m_2 l_1 l_{c2} \text{cos}(q_2) + m_2 l_{c2}^2 + I_2 & m_2 l_{c2}^2 + I_2 \end{bmatrix}$$

II. $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in R^{2 \times 2}$, la matriz de fuerzas centrípetas y de Coriolis:

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -2m_2 l_1 l_{c2} \text{sen}(q_2) \dot{q}_2 & -m_2 l_1 l_{c2} \text{sen}(q_2) \dot{q}_2 \\ m_2 l_1 l_{c2} \text{sen}(q_2) \dot{q}_1 & 0 \end{bmatrix}$$

III. $\mathbf{g}(\mathbf{q}) \in R^2$, el vector de par gravitacional:

$$\mathbf{g}(\mathbf{q}) = g \begin{bmatrix} m_1 l_{c1} \text{sen}(q_1) + m_2 (l_1 \text{sen}(q_1) + l_{c2} \text{sen}(q_1 + q_2)) \\ m_2 l_{c2} \text{sen}(q_1 + q_2) \end{bmatrix}$$

El modelo dinámico del robot manipulador de 2GDL es una ecuación de segundo orden, mediante un adecuado cambio de variables de estado, la ecuación diferencial ordinaria de primer orden $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ resulta de la siguiente forma:

$$\frac{d}{dt} \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{M}^{-1}(\mathbf{q}) [\tau - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{B} \dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})] \end{bmatrix}$$

donde la variable de estado \mathbf{x} está compuesta por el vector de posición $\mathbf{q} = [q_1, q_2]^T \in R^2$ y el vector de velocidad $[\dot{q}_1, \dot{q}_2]^T \in R^2$, es decir $\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \in R^4$.

CAPÍTULO 2: MODELADO DEL ROTRADI VI

En esta sección se presenta la metodología e Euler-Lagrange aplicada a robots manipuladores en general, con la cual se obtendrán los parámetros necesarios para obtener el modelo dinámico del robot.

En esta parte hay que remarcar la importancia del correcto cálculo del modelo cinemático, ya que esto radica en el hecho de que con este se obtiene el posicionamiento del eslabón final del robot en base a sus parámetros dentro de un espacio de trabajo definido, así mismo la cinemática directa es de suma importancia, ya que nos permitirá obtener el Jacobiano el cual es un parámetro necesario para varios esquemas de, principalmente en el control cartesiano.

2.1 MODELADO DINÁMICO DE UN ROBOT DE 3 GDL

A continuación se muestra la metodología necesaria para obtener el modelo dinámico del robot, en donde los parámetros a calcular están basados en las Figuras 2.1 y 2.2.

Paso 1: obtener la cinemática directa del centro de masa de cada uno de los eslabones. Para el i -ésimo eslabón tomar en cuenta las longitudes anteriores l_{i-1} y ángulos q_{i-1} :

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = f_R(l_i, l_{i-1}, l_{ci}, q_{i-1}, q_i)$$

Para el caso del ROTRADI VI, la cinemática directa de cada eslabón corresponde a:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ l_1 \end{bmatrix} \quad (2.1)$$

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} l_{c2} \text{sen}(q_2) \cos(q_1) + w_1 \text{sen}(q_1) \\ l_{c2} \text{sen}(q_2) \text{sen}(q_1) + w_1 \cos(q_1) \\ l_1 - l_{c2} \cos(q_2) \end{bmatrix} \quad (2.2)$$

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} \cos(q_1) (l_2 \text{sen}(q_2) + l_{c3} \text{sen}(q_2 + q_3)) + (w_1 + w_2) \text{sen}(q_1) \\ \text{sen}(q_1) (l_2 \text{sen}(q_2) + l_{c3} \text{sen}(q_2 + q_3)) + (w_1 + w_2) \cos(q_1) \\ l_1 - l_2 \cos(q_2) - l_{c3} \cos(q_2 + q_3) \end{bmatrix} \quad (2.3)$$

Paso 2: obtener la cinemática diferencial del i-esimo eslabón y deducir la rapidez lineal:

$$\mathbf{v}_i = \frac{d}{dt} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}. \quad (2.4)$$

La rapidez lineal del centro de masa de cada eslabón se calcula de la siguiente forma:

$$\mathbf{v}_i^T \mathbf{v}_i = \dot{x}_i^2 + \dot{y}_i^2 + \dot{z}_i^2 \quad (2.5)$$

Paso 3: obtener el modelo de energía:

$$\mathcal{H}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{U}(\mathbf{q}) \quad (2.6)$$

La energía cinética $\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}})$ incluye el movimiento de traslación y rotación.

Obtener el lagrangiano: $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{U}(\mathbf{q})$

Paso 4: aplicar las ecuaciones de movimiento de Euler-Lagrange e incluir el modelo de fricción.

$$\boldsymbol{\tau}_i = \frac{d}{dt} \left[\frac{\partial L(\mathbf{q}_i, \dot{\mathbf{q}}_i)}{\partial \dot{\mathbf{q}}_i} \right] - \frac{\partial L(\mathbf{q}_i, \dot{\mathbf{q}}_i)}{\partial \mathbf{q}_i} + b_i \dot{\mathbf{q}}_i + f_{ci} \text{signo}(\dot{\mathbf{q}}_i) + f_{ei} [1 - |\text{signo}(\dot{\mathbf{q}}_i)|] \quad (2.7)$$

donde f_c es la fricción de Coulomb, f_e es la fricción estática, b es el coeficiente de fricción viscosa y la función $\text{signo}(\dot{\mathbf{q}})$ está definida como:

$$\text{signo}(\dot{\mathbf{q}}) = \begin{cases} 1 & \text{si } \dot{\mathbf{q}} > 0, \\ -1 & \text{si } \dot{\mathbf{q}} < 0. \end{cases}$$

**Para ver el modelo dinámico completo del robot de 3 grados de libertad, revisar el Apéndice A.

El correcto cálculo de los parámetros del sistema, permitirá un funcionamiento óptimo por parte del programa de control, por lo que es recomendable seguir los pasos descritos en esta sección para poder llegar a un resultado deseado.

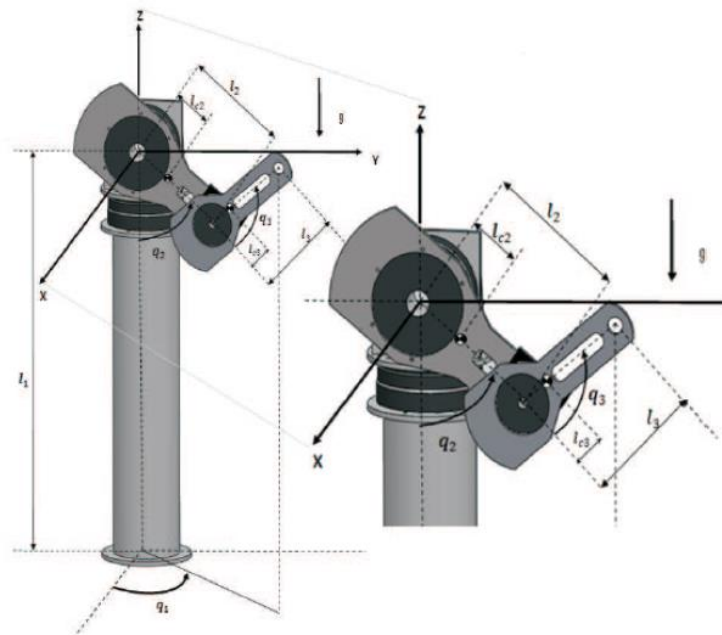


Figura 2.1: Robot manipulador de 3 GLD usado como referencia.

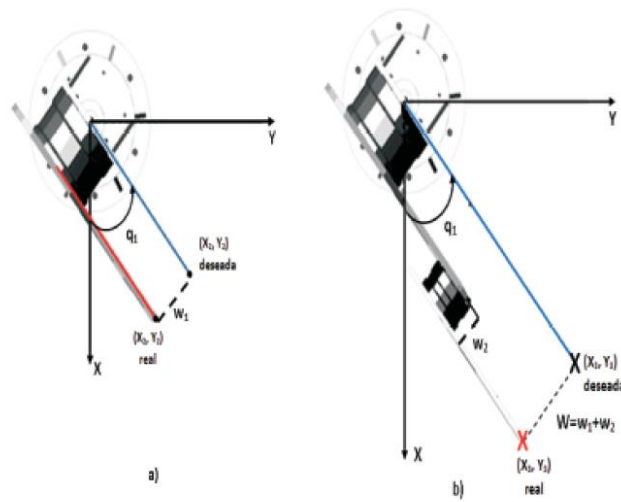


Figura 2.2: Robot manipulador de 3 GDL visto de desde arriba.

CAPÍTULO 3: ELECTRÓNICA DE CONTROL

En este capítulo se realiza un análisis del funcionamiento de las señales de control del robot, así como del método por el cual serán interpretadas por el microcontrolador según la configuración que más convenga en el caso de las señales de salida diferencial de cada uno de los encoders A-B. De igual manera se analizará la forma más apropiada en la cual se controlará el comportamiento de los actuadores a través de señales analógicas y como controlar el equivalente en torque de los mismo.

A su vez se identifican los componentes necesarios para el tratamiento y ajuste de las señales que permita una comunicación apropiada entre la tarjeta Arduino Mega, las tarjetas electrónicas diseñadas y el ROTRADI VI. Para esto se establece un esquema de funcionamiento ilustrado en la Figura 3.1 en donde se toman en cuenta los dispositivos necesarios y las características de funcionamiento de cada actuador del sistema para su control.

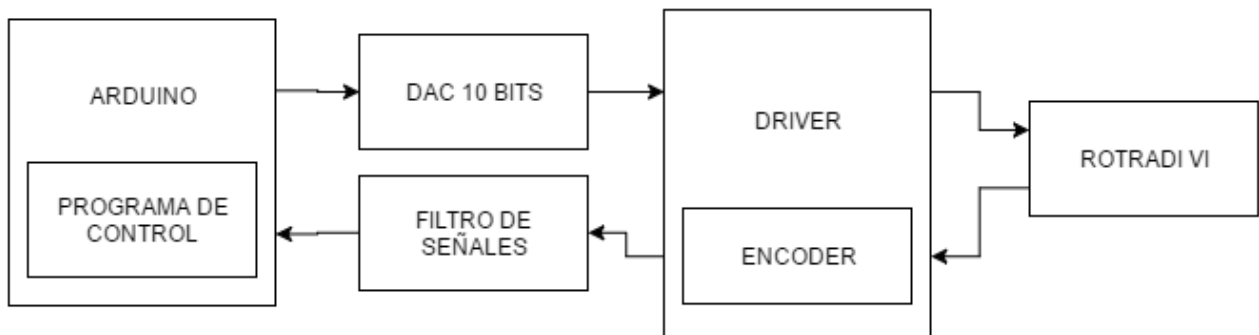


Figura 3.1: Esquema de funcionamiento del sistema.

3.1 TRATAMIENTO DE LAS SEÑALES DE LOS ENCODERS

Parte indispensable del control de la planta radica en el posicionamiento del robot, en este caso la configuración de la señales en cuadratura A-B.

Debido a una gran cantidad de factores externos, principalmente al campo electromagnético generado por dispositivos electrónicos aledaños, la señales provenientes del *encoder* pueden encontrarse contaminadas por señales de ruido, impidiendo que el microcontrolador sea

capaz de realizar correctamente la lectura de posición, por tanto parte de la electrónica que compone la interfaz de control radica en hacer un tratamiento a las señales entregadas por el driver de los motores de transmisión directa, en este caso se tiene que las señales provenientes del *encoder* presentan ruido tanto en la señal A como en la señal B. Este ruido genera pulsos no deseados que hacen que el micro controlador realice conteos que provocan un incremento en la posición, presentes aun cuando el motor no está en movimiento, por lo tanto es necesario utilizar filtros y comparadores para tener una lectura sin fallos o alteraciones.

Dentro del tipo de operaciones que realizan los amplificadores operacionales se encuentra el procesamiento de señales mediante filtrado. Para mejorar la respuesta del sensor se emplean filtros que eliminan frecuencias altas inmersas en la señal de dicho sensor; para lograr esta finalidad se selecciona un parámetro del filtro denominado frecuencia de corte (filtro pasa bajas)[23]. En la figura 3.2 se muestra el diagrama electrónico de un filtro pasa bajas en conjunto con un comparador [23].

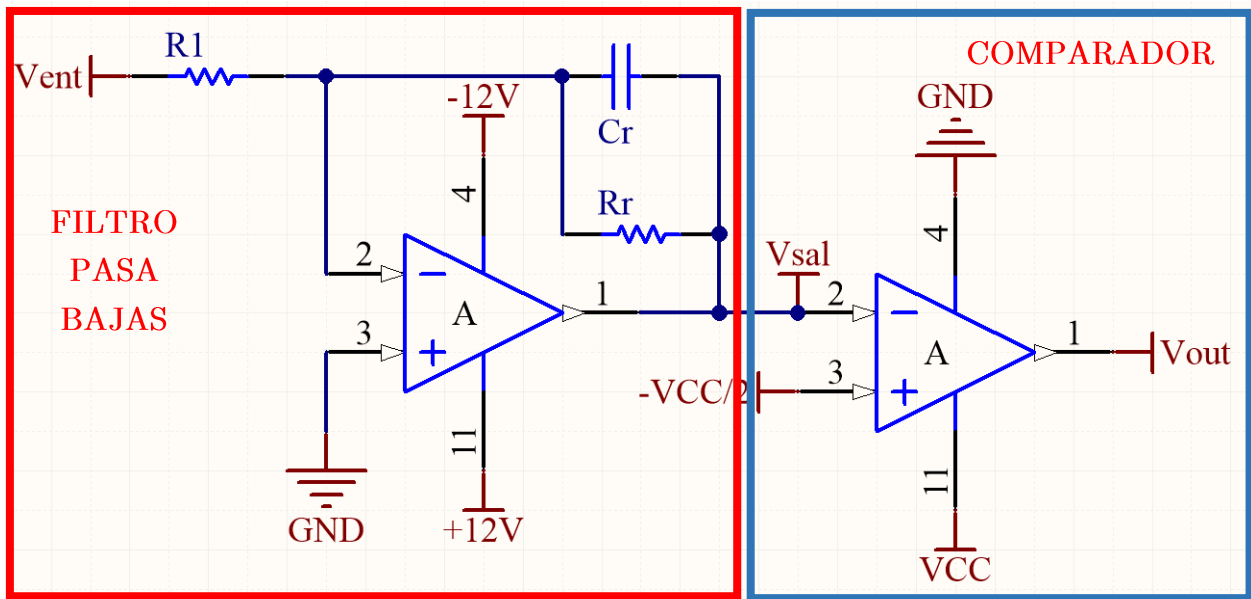


Figura 3.2: Esquema de conexiones de la etapa de filtrado para las señales A y B del *encoder*.

Para llevar a cabo el análisis del filtro pasa bajas se requiere del siguiente conjunto de ecuaciones [23]:

$$\frac{v_{ent} - v_1}{R_1} = \frac{v_1 - v_{sal}}{Z_r} \quad (3.1)$$

$$Z_r = \frac{R_r \frac{1}{sC_r}}{R_r + \frac{1}{sC_r}} = \frac{R_r}{1 + R_r C_r s} \quad (3.2)$$

donde Z_r es la impedancia eléctrica que representa los elementos de retroalimentación negativa R_r y C_r . Además a que el nodo de la entrada inversora es una tierra virtual, entonces se cumple $v_1 = v_2 = 0$; obteniéndose:

$$\frac{v_{ent}}{R_1} = -\frac{v_{sal}}{Z_r} \quad (3.3)$$

$$v_{sal} = -\frac{R_r}{R_1} \frac{1}{1+R_r C_r s} v_{ent} \quad (3.4)$$

La función de transferencia del filtro pasa bajas es:

$$\frac{v_{sal}}{v_{ent}} = -\frac{R_r}{R_1} \frac{1}{1+R_r C_r s} \quad (3.5)$$

$$= -\frac{R_r}{R_1} \frac{1}{1+Ts} = -\frac{R_r}{R_1} \frac{\omega_c}{s+\omega_c} \quad (3.6)$$

donde ω_c se denomina frecuencia de corte:

$$\omega_c = \frac{1}{T} = \frac{1}{R_r C_r} \text{ y el termino } \frac{R_r}{R_1} \text{ representa la ganancia en DC.}$$

El ruido presente en las señales A-B medido experimentalmente a través del osciloscopio es de aproximadamente 7 kHz, por lo que se diseña un simple filtro activo pasa bajas, con una frecuencia de corte de aproximadamente 15 kHz (16.66 kHz), suficiente para limpiar el ruido presente en las señales. Cuyos valores de cada componente son: $R_1 = R_r = 500 \Omega$ y $C_r = 0.00000012 F$ o $C_r = 0.12 \mu F$. La segunda parte del circuito corresponde a un comparador cuya función principal es el enviar una señal de salida en alto (Vcc) o bajo (Gnd) dependiendo si la señal de voltaje entrada (Vsal) es mayor al voltaje de referencia (-Vcc/2). Los voltajes de referencia Vcc y -Vcc, están dados dependiendo del voltaje máximo de entrada del microcontrolador usado, en este caso es de Vcc=5 V.

En la Figura 3.2 se muestra el esquema de conexiones, hay que tomar en cuenta que se requieren de 2 juegos, uno para cada señal del *encoder* A y B de cada motor.

Algo que hay que mencionar, es que el uso de arreglos con OPAMPs, no es la única forma de filtrar el ruido presente en las señales, dependiendo de la robustez del sistema a utilizar podría no ser requerido el uso de filtros para la adquisición de la posición angular, esto va a depender del tipo de motor utilizado y del tipo de *encoder* que tenga instalado. En la Figura 3.3 se muestra la tarjeta de filtrado de señales armada sobre una placa fenólica.

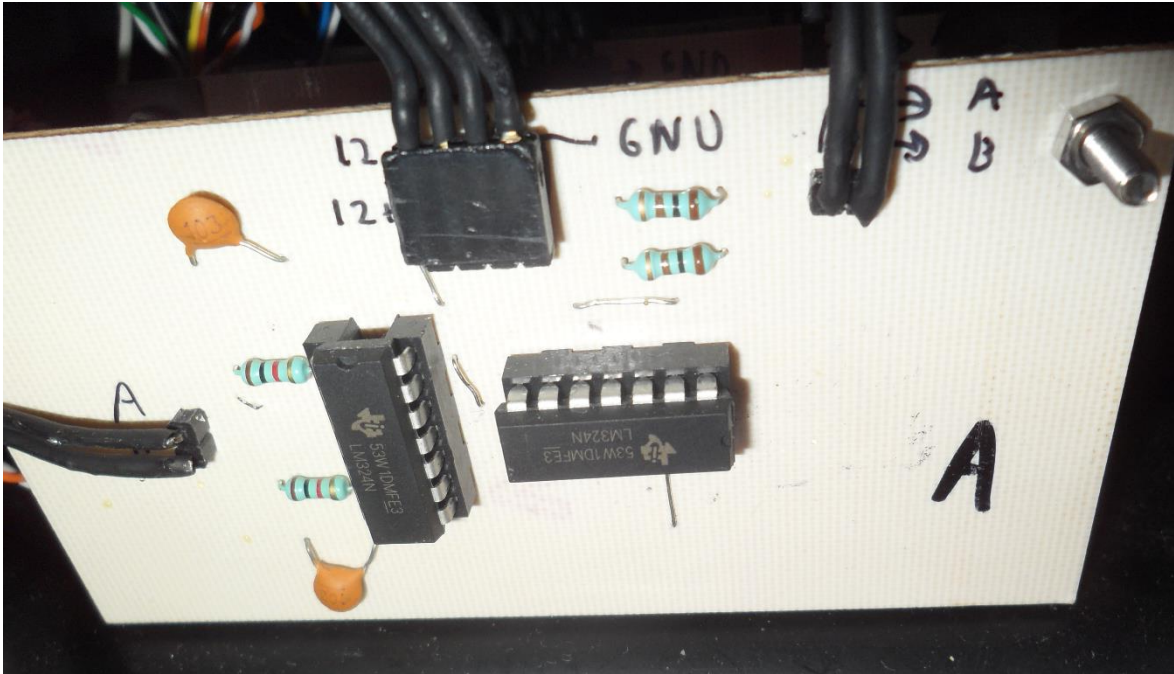


Figura 3.3: Tarjeta de filtrado A de señales aplicado para la señal A+ y B+.

3.2 ANÁLISIS DEL ALGORITMO DE LECTURA DEL ENCODER

Como parte fundamental del trabajo de tesis, radica el hecho de usar dispositivos comerciales y de fácil acceso para el control de otros dispositivos más avanzados y de un grado tecnológico mucho más refinado, en este caso el micro controlador Arduino y el control de los motores de transmisión directa.

Por una parte tenemos que los dispositivos Arduino son tarjetas programables que en los últimos años se han vuelto populares para el manejo de sistemas domóticos[25] y robóticos[25] entre otros, esto por su fácil entorno de programación y su inmensa cantidad de soporte tanto en hardware (“Shields”) y software (Librerías) que va creciendo año tras año. Por otro lado tenemos a los motores de transmisión directa, los cuales son controlados por servo drivers. Estos motores tienen la particularidad de tener un comportamiento lineal con respecto a un voltaje de control aplicado, convirtiéndolos en un punto importante para el estudio de algoritmos de control en robots manipuladores construidos con esta clase de motores. Además de su característica principal ya mencionada, otra de las características que tienen estos motores es una enorme resolución en su *encoder*, por lo general superior al millón de pulsos por vuelta, lo que nos da escalas de micro grados por pulso. Tomando esto en cuenta se tiene que enfrentar al hecho de que las características del Arduino podrían no

ser las suficientes para usar estos motores a toda su capacidad de no tener en cuenta las siguientes consideraciones:

La forma de trabajo del Arduino es secuencial, lo cual significa que se activan los comandos uno tras otro, por tanto se requiere un determinado tiempo para ejecutar los comandos completos, este tiempo medido en ciclos de reloj debe considerarse con respecto a la frecuencia máxima de la señal de entrada para conocer la frecuencia de trabajo del micro controlador mínima requerida para poder realizar la tarea requerida sin fallas en la lectura, esta frecuencia puede ser calculada atreves de la siguiente expresión.

$$f_{Lectura} = f_{señal_max} C_{clock},$$

donde $f_{señal_max}$ es la frecuencia máxima de la señal de posición del *encoder* de cuadratura, debe tomarse en cuenta que este valor es obtenido de la frecuencia máxima de una de las señales multiplicada por el número de estados o combinaciones que se forman con ambas señales en cuadratura. C_{clock} es el número de ciclos de reloj que tarda nuestro micro controlador en realizar las tareas deseadas. Tomando en cuenta la frecuencia máxima de las señales del *encoder* del motor, en este caso de 750 kHz para cada señal A y B [24], y el desfase de 90° entre las señales, la frecuencia para los estados generados por las señales son 4, por lo que la frecuencia máxima de la señal en cuadratura completa es de 3 MHz, en la Figura 3.4 se muestra el comportamiento de las señales de un *encoder* de cuadratura.

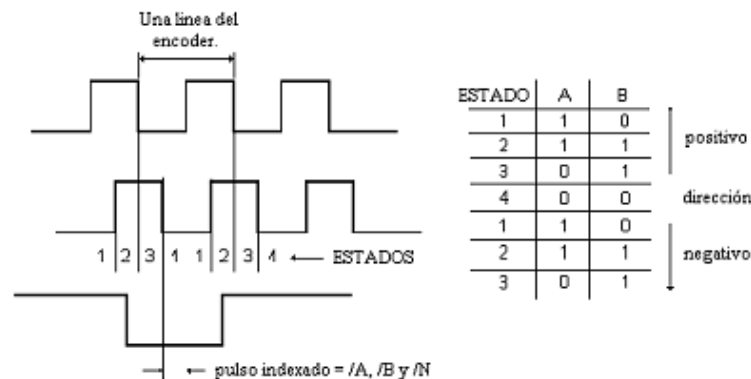


Figura 3.4: Combinaciones que conforman los estados encontrados en el *encoder* de cuadratura.

Ahora, tomando en cuenta el tiempo de lectura y almacenaje del programa en Arduino que es en total de **120 ciclos de reloj** con los comandos estándar, tenemos que la velocidad de reloj del microcontrolador debe ser mínimo de **360 MHz** para usar el motor a la máxima resolución del *encoder*, por lo que las únicas versiones de Arduino que podrían ejecutar la tarea sería el Arduino Galileo o el Arduino Edison, cuyas frecuencias de reloj son de 400 MHz y 500 MHz, respectivamente. Adicionalmente hay que reiterar que esto es solo para la lectura de un solo *encoder*, por lo que se requeriría un dispositivo de no menos de 1080 MHz para realizar la lectura de los tres *encoders* que conforman el ROTRADI completo o el uso

de 3 microcontroladores separados que permitan la lectura individual de los *encoder*. Adicionalmente, hay que considerar que se requeriría de ciclos de reloj adicionales para llevar a cabo todas las conversiones, cálculos y demás transformaciones requeridos para el programa de control, por lo que lo más conveniente para este caso es el escalamiento de los *encoders* del ROTRADI a una resolución general de **10000** pulsos por vuelta para cada uno de los motores.

3.3 OBTENCIÓN DE DATOS DE POSICIONAMIENTO

En un inicio, el programa de control para el posicionamiento angular del ROTRADI consiste en la obtención de la posición actual del robot a través del uso de los servicios de interrupción, los cuales son recursos o mecanismos del microcontrolador para responder a eventos, permitiendo suspender temporalmente el programa principal para ejecutar una subrutina de servicio de interrupción (ISR, por sus siglas en inglés) [25]. La tarjeta Arduino Mega tiene capacidad para ejecutar seis rutinas de interrupción a la vez y cada *encoder* requiere de dos rutinas de interrupción para realizar una correcta lectura de la posición; una vez terminada dicha subrutina se reanuda la ejecución del programa principal. Como se ha mencionado, la combinación de las señales A-B de cada señal va a determinar el sentido de movimiento en base al estado anterior, ver Figura 3.5.

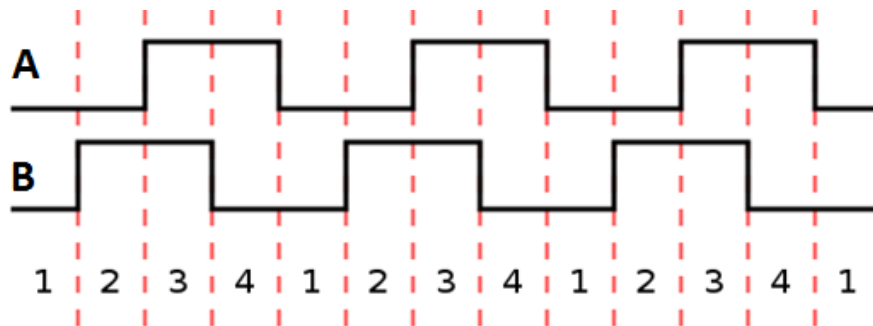


Figura 3.5: Forma de las señales A y B del *encoder*.

Analizando la forma en que se combinan las señales mostradas en la Figura 3.5, se puede notar que tanto la señal A, como la señal B se van intercalando en cada fase en cuanto equivalencia de los estados lógicos, por ejemplo, en la fase 1 y 3, se puede notar claramente que tanto la señal A y la B son idénticas en cuanto estado lógico Alto o Bajo, mientras que en la fase 2 y 4 son diferentes, por lo que tenemos que para el *encoder* de cuadratura de 2 señales solo hay dos opciones posibles:

$$A = B \text{ o } A \neq B$$

En otras palabras, podemos ver que en la fase 1, el valor de A y B es 0 (Bajo), seguido de la fase 2 cuyas valores de A es 0 y B es 1 (Bajo y Alto respectivamente), luego en la fase 3 para A y B es 1 (Alto) y finalmente en la fase 4, A es 1 y B es 0 (Alto y Bajo respectivamente), repitiéndose este ciclo ya sea en un sentido o en el otro, por lo que para propósitos de programación, tenemos que si la señal A no es igual a B, entonces al valor de lectura del *encoder* se le suma 1, en caso contrario se resta 1 y que si la señal A es igual a B, entonces al valor del *encoder* se le suma 1, en caso contrario se resta 1.

Con esto se tiene una parte fundamental para el control del robot, ya que esta información permite obtener otras variables como velocidad y aceleración.

3.4 CONVERTIDOR DIGITAL ANALÓGICO DE 10 BITS

El torque deseado y la dirección del movimiento del motor están controlados por las terminales 35 y 36 del driver, siendo el 35 la entrada de voltaje analógico el cual controla tanto el par como el sentido de movimiento dependiendo directamente de la señal de voltaje implementada que puede ir de $\pm 10V$ o $\pm 6V$ dependiendo de la configuración cargada al servo driver. Según la configuración, el máximo torque máximo incrementándose o decrementándose de forma lineal, dependiendo del nivel de la señal de voltaje analógico, por lo que si implementamos un voltaje positivo el movimiento del motor será en sentido de las manecillas del reloj y el voltaje negativo en sentido inverso a las manecillas del reloj.

Debido a que los DACs del Arduino Due no son lineales y adicionalmente en sus valores de cero sigue mandando un voltaje de aproximadamente 600mV, es necesario construir uno de tal forma que se permita obtener los valores de voltaje deseados a partir de una conversión digital-analógica, para esto se recurre a un circuito sumador con OpAmp's para generar un voltaje de salida analógica dependiendo del valor binario enviado por el bus de datos de la tarjeta Arduino. Este bus es creado utilizando 10 puertos digitales del Arduino, los cuales pueden dar hasta 1024 valores de voltaje. Tomando en cuenta que se requiere de voltaje negativo y positivo para hacer que el motor cambie de dirección, se fija un punto medio para que el valor binario de 512 del valor de 0V, siendo 0 el valor de -10V y 1023 el valor de 10V. En la Figura 3.6 se muestra el circuito para la construcción del DAC de 10 bits y en la Figura 3.7 su montaje en placa fenólica.

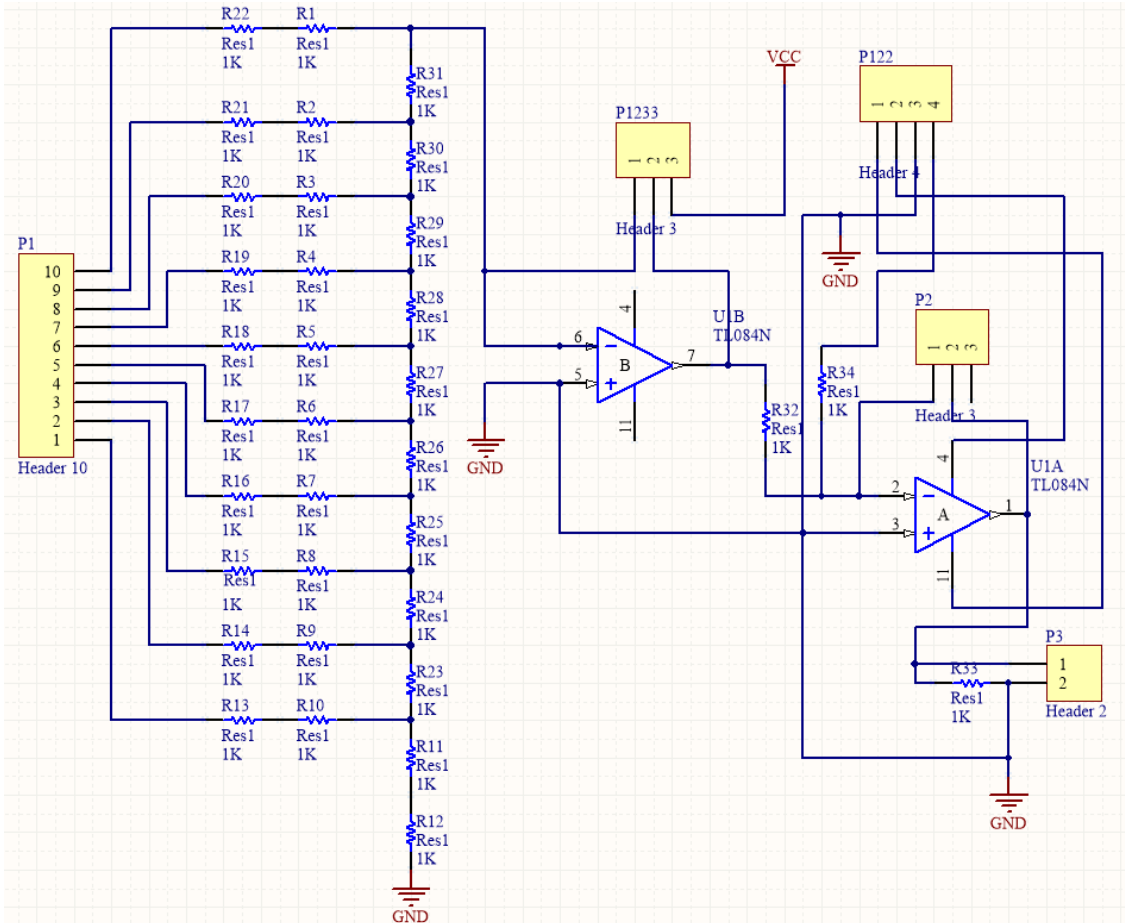


Figura 3.6: Circuito para un DAC de 10 bits.

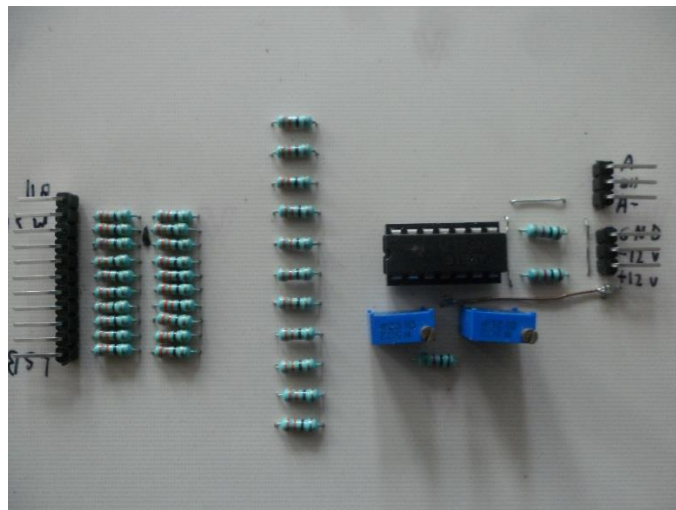


Figura 3.7: Tarjeta del circuito DAC diseñado para el ROTRADI VI

Ahora se realiza una serie de pruebas en donde se genera una tabla en base a los valores binarios y el nivel de voltaje de salida, en un inicio hay que considerar que para el circuito DAC se armó con resistencias comunes, por lo que es necesario calibrar las salidas finales a través de 2 potenciómetros trimpot para que la salida en sus valores extremos (0 y 1023)

sean los $\pm 5V$ que se proponen para el funcionamiento de prueba de cada motor al 50% del torque máximo del que se dispone en cada actuador, mostrado en la Figura 3.8.

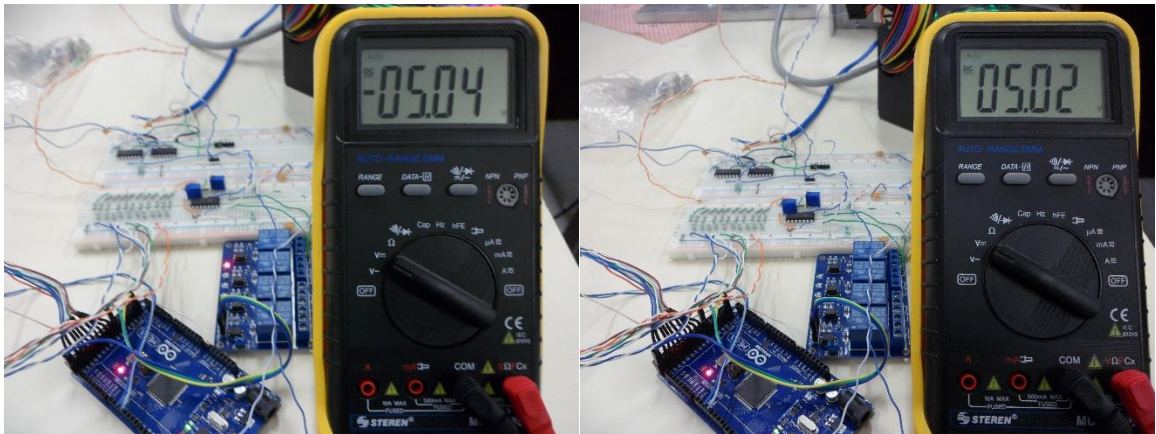


Figura 3.8: Valores entregados en los bits 0 y 1023, respectivamente.

En la Figura 3.9 se muestra la gráfica de funcionamiento del DAC en la cual se puede apreciar que tiene una linealidad casi perfecta².

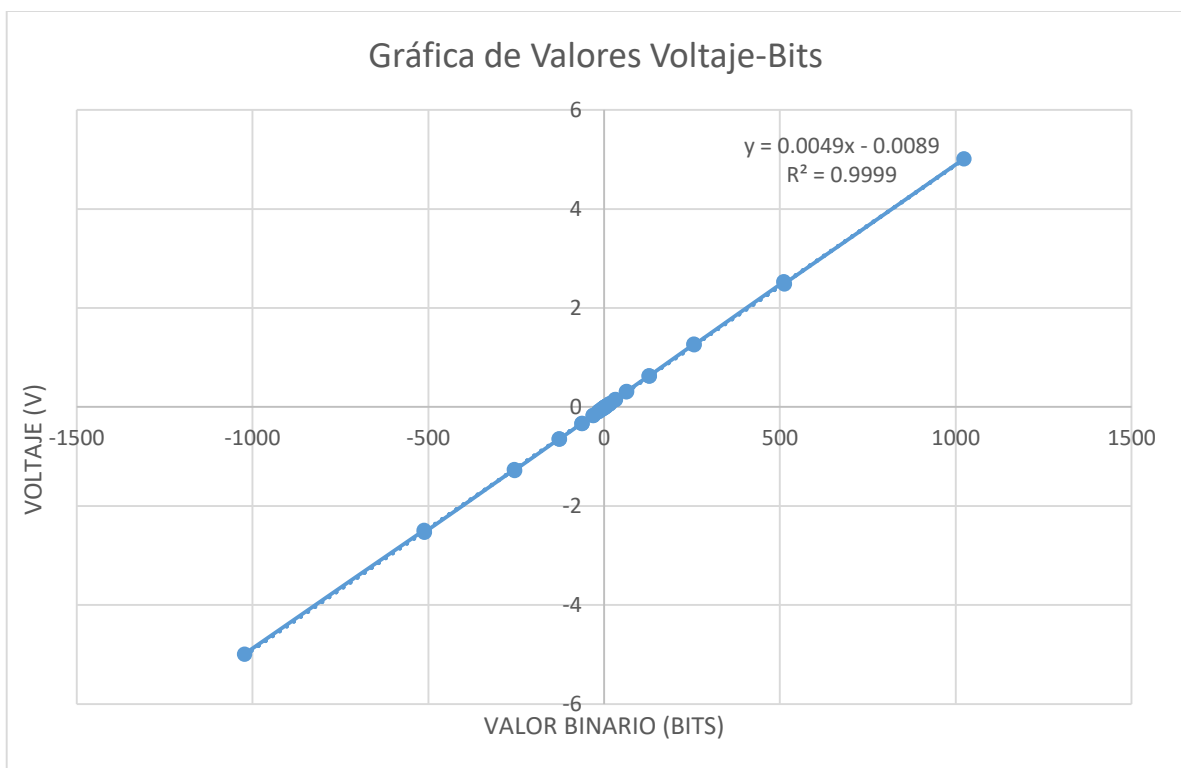


Figura 3.9: Gráfica de valores de Voltaje-Bits.

² Nota: Los valores de salida de cada pin que conforma el Bus de salida del Arduino, no dan valores exactos de 5 V, hay variaciones que van de 4.7 a 5V tras energizar el Arduino, por lo que es necesario esperar unos segundos para que el voltaje de salida sea constante.

3.4 SENSOR DE FUERZA

Un sensor de fuerza es un transductor del cual se mide una tensión inducida por fuerza o par aplicado a éste. Un componente básico de un sensor de fuerza es un indicador de tensión, que es una pequeña espira plana de alambre conductor que cambia su resistencia cuando se ejerce fuerza sobre ella. Este indicador se coloca sobre la placa y se mide el cambio en la resistencia para determinar la tensión, debido a que ésta se encuentra relacionada con la fuerza aplicada sobre la placa.

Una forma clásica de estos transductores es el arreglo por puente de Wheatstone, en donde a través de dicho arreglo las salidas de la celda de carga enviarán una señal de voltaje proporcional a la fuerza aplicada. Estas señales deben ser acondicionadas para así ser interpretadas por un microcontrolador. En la Figura 3.10 se muestra el arreglo clásico del puente de Wheatstone implementado en la celda de carga utilizado como sensor de fuerza.

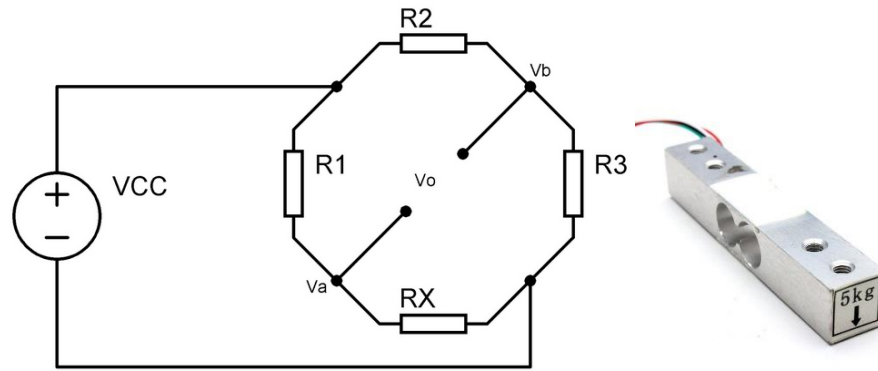


Figura 3.10: Arreglo del puente de Wheatstone y el sensor de fuerza (celda de carga diferencial).

$$v_o = \left(\frac{R_x}{R_1 + R_x} - \frac{R_3}{R_2 + R_3} \right) v_{cc}$$

El arreglo del puente de Wheatstone en la celda de carga consiste principalmente en la variación de las resistencias R_x y R_3 , las cuales cambian su valor de resistencia cuando se aplica una fuerza sobre la celda, haciendo que la resistencia ubicada en la parte superior varíe su valor con la expansión y que la resistencia en la parte inferior varíe su valor con la compresión. La variación de las resistencias genera una diferencia de la salida de voltaje la cual sirve para calcular la fuerza aplicada sobre el sensor a través de un dispositivo electrónico.

Dado que las salidas de la celda de carga entregan un voltaje diferencial en la escala de mili volts, es necesario realizar una amplificación de estas señales. Para la amplificación de las

señales de salida del sensor, se recurre al integrado INA114, cuyo circuito se muestra en la Figura 3.11. Es un amplificador de instrumentación cuya característica principal es el bajo nivel de ruido y gran versatilidad para el manejo de sensores de voltajes diferenciales como lo son celdas de carga o electrocardiogramas. El voltaje de salida del INA114 se obtiene a través de las siguientes expresiones:

$$V_o = G(V_{IN}^+ - V_{IN}^-) \quad y \quad G = 1 + \frac{50k\Omega}{R_G}$$

donde V_o es el voltaje de salida, G es el valor de ganancia, V_{IN}^+ y V_{IN}^- son los voltajes de salida de la celda de carga y R_G es la resistencia de ganancia que para propósitos de esta tesis se ajusta de tal manera que la resistencia se acerque a 50Ω para tener una ganancia $G = 1000$.

**Nota: los valores V_{IN}^+ y V_{IN}^- son voltajes positivos que están en función a la deformación de la celda causada por una fuerza aplicada, de ahí su nombre de celda de carga diferencial.

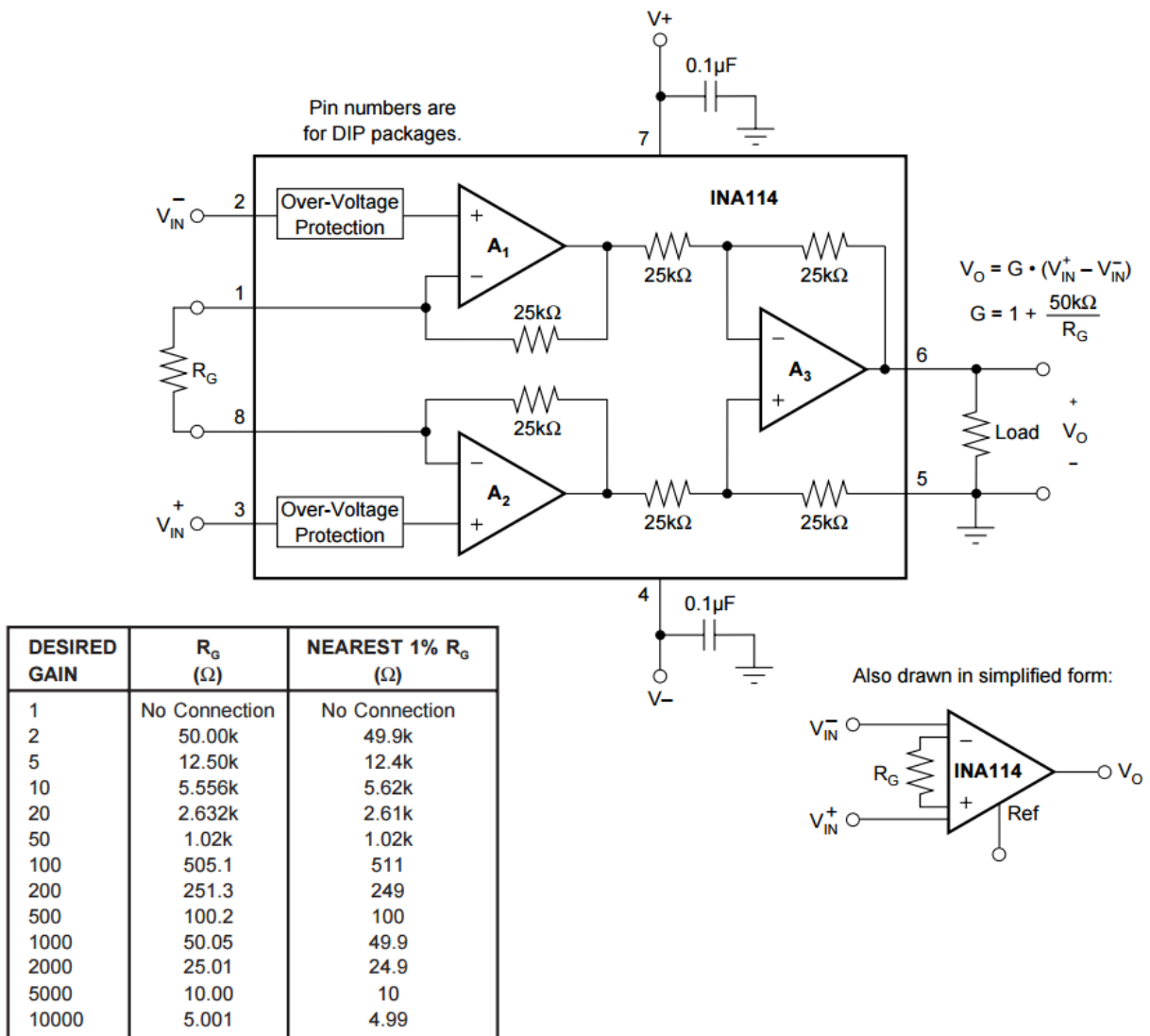


Figura 3.11: Diagrama de conexiones del INA114 y tabla de ganancias del fabricante.

3.4.1 IMPLEMENTACIÓN

Para la calibración del sensor de carga se recurre a la ley de Hooke, la que traducida para propósitos electrónicos se tiene $F = K(V_L - V_C)$, en donde F es la fuerza esperada, K es una constante de elasticidad, V_L es la lectura del voltaje entregada por el sensor y V_C es un voltaje de compensación.

Entre las características del sensor utilizado tenemos que la capacidad máxima nominal que se puede aplicar al sensor es de 5 kg, esto es, un peso de 49.05 N. El esquema de conexiones se muestra a continuación en la Figura 3.12.

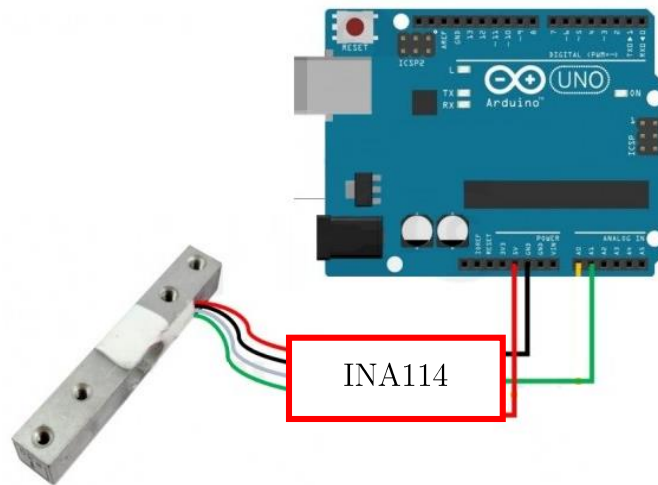


Figura 3.12: Esquema de conexión para Arduino-Celda de carga.

Para poder realizar la caracterización del sistema, se utilizan una serie de objetos de diferentes pesos conocidos para una posterior comparación. Para esto se hace un arreglo a modo de báscula, de tal forma que la celda de carga tenga el objeto a medir en un extremo del arreglo, mientras que del otro extremo este sujeto a una superficie fija como se muestra a continuación en la Figura 3.13.

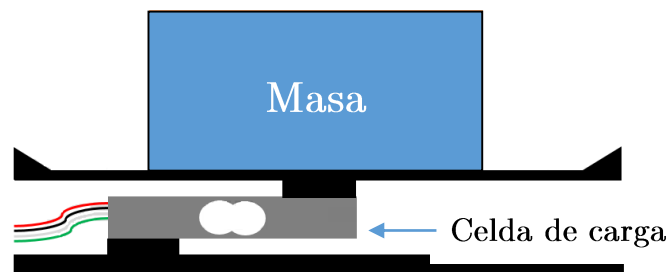


Figura 3.13: Implementación de la celda de carga para la medición del peso.

En la Figura 3.14 se muestra la implementación del sensor en conjunto con el ROTRADI VI.

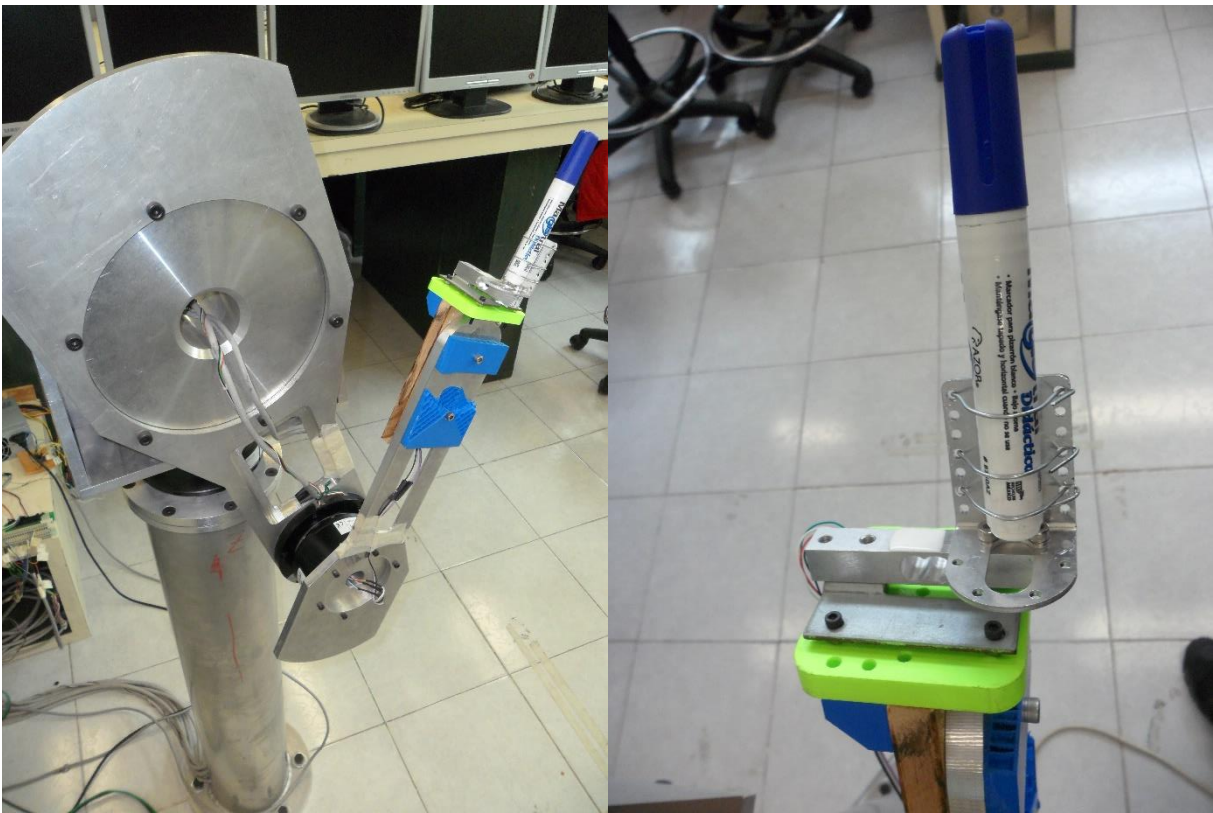


Figura 3.14: Implementación de la celda de carga en el efector final del ROTRADI VI.

Variando el valor de la resistencia del diagrama de conexiones del INA114, se ajusta de tal manera que el valor de salida sea de 1mV para una lectura de 1g, dando un voltaje proporcional de 1V para 1kg de masa, con esto se diseña un simple programa de lectura de voltaje analógico que permita el censado de peso, el comando es:

$$\text{Peso} = (\text{analogRead}(\text{PIN}) * 5.0 / 1023) * 9.8;$$

Nota** La resolución de cada puerto analógico es de 1023 valores de voltaje para un valor de lectura máximo de 5V, dando un valor de lectura de 0.0048V por bit.

Esto permitirá obtener el equivalente en fuerza en el programa de control tras multiplicar el valor obtenido por la constante gravedad obteniendo un valor de fuerza dado en Newtons-Metro. Para realizar la caracterización de este sistema, se mide una serie de objetos usando una báscula digital donde se registran los valores de pesos conocidos y a continuación se

registra el valor de bit medido por el programa del Arduino así como el valor de peso registrado. En la Figura 3.15 se muestra una gráfica donde se compara el valor esperado (azul) y el valor censado (rojo), los cuales casi se sobrepone.

TABLA 3.1: Tabla de datos de la caracterización de la celda de carga.

	Masa (g)	Voltaje (V)	Lectura en Bits	Masa Registrada (Kg)
Taza	320	0.321	65	0.31769306
Cilindro de aluminio	1000	1.031	210	1.02639296
Botella con pegamento	70	0.074	15	0.07331378
Cargador de Asus	620	0.622	127	0.62072336
Rollo de estaño	127	0.126	25	0.12218964
2 monedas de 1 peso	8	0.007	1	0.00488759

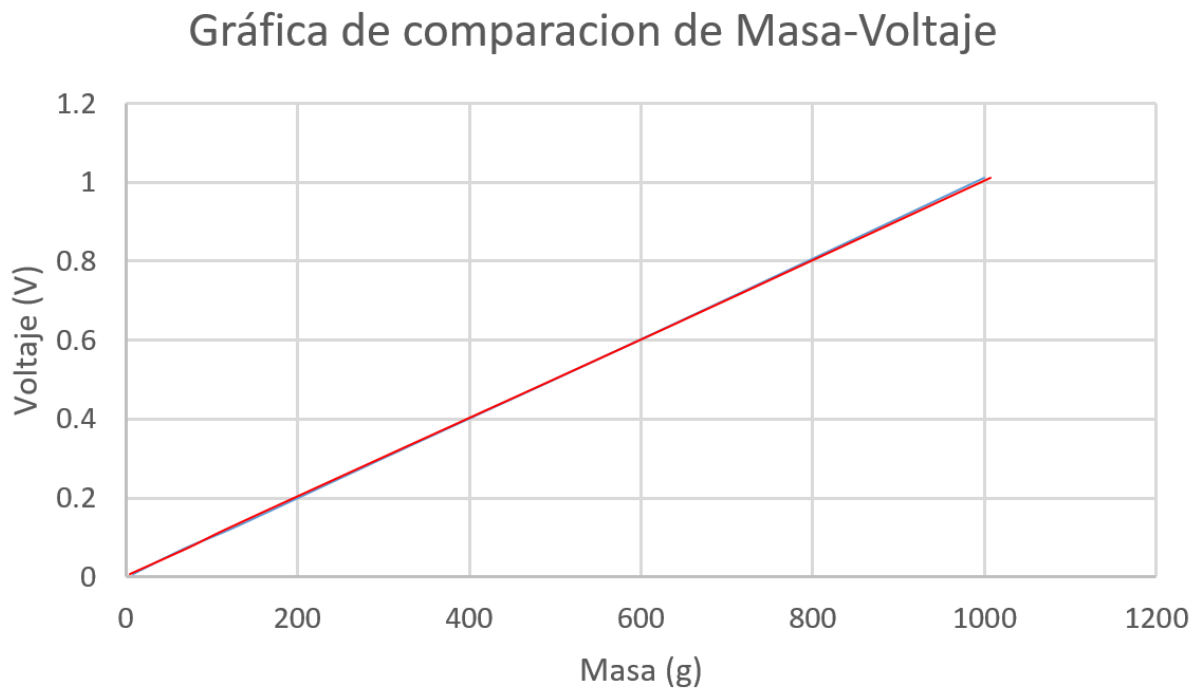


Figura 3.15: Gráfica de comparación de los valores reales y los censados.

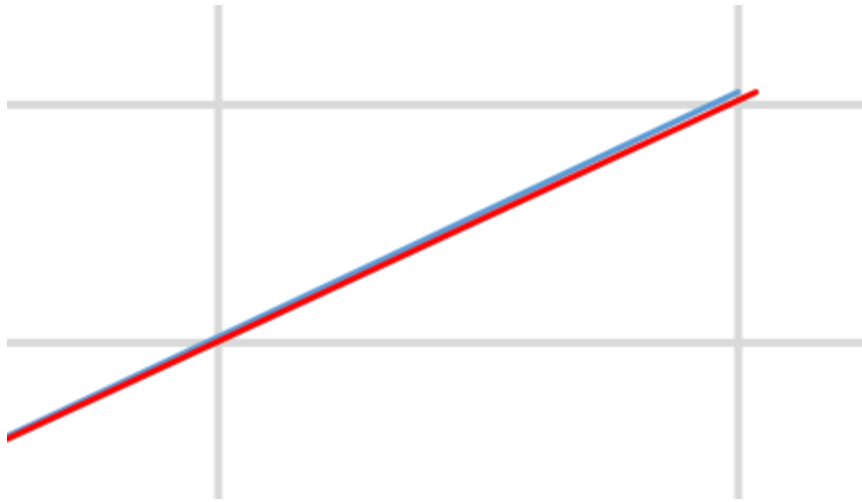


Figura 3.16: Zoom de gráfica de la figura 3.15, en donde se muestra las variaciones entre la medición real y la censada.

En la Figura 3.16 se muestra el INA114 montado en una placa fenólica y en la Figura 3.17 se muestra el funcionamiento del programa para calcular el peso en el ambiente de Arduino.

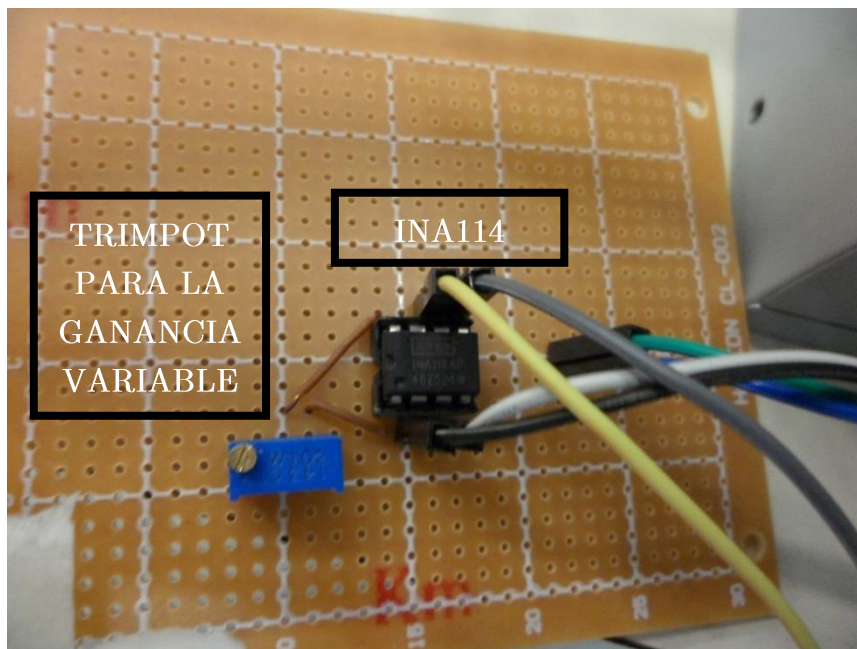


Figura 3.16: Integrado INA114 montado en una placa fenólica.

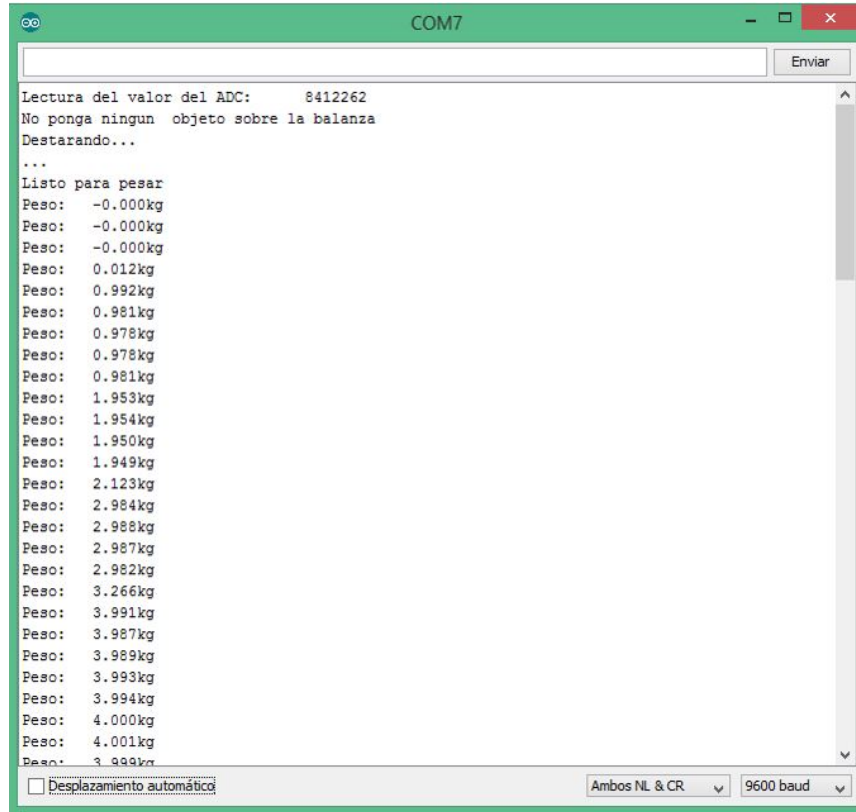


Figura 3.17: Programa en Arduino para obtener el peso.

Finalmente en la Figura 3.18 se presenta la “caja negra” en donde se contienen todos los componentes diseñados para la interface electrónica del ROTRADI VI.

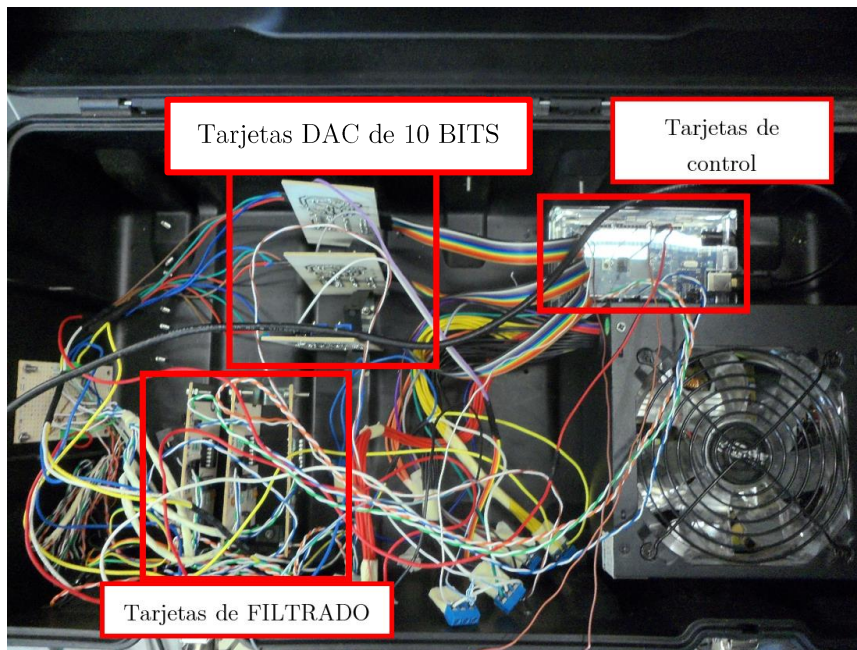


Figura 3.18 Caja negra en donde se incluyen las tarjetas de control de sistema.

En esta sección se presentó la electrónica necesaria para la puesta en operación de cada uno de los servomotores de transmisión directa, esto a su vez permitirá la aplicación de cada uno de los esquemas de control requerido para alcanzar el objetivo principal de esta tesis.

CAPÍTULO 4: ALGORITMOS DE CONTROL DE ARDUINO

En esta sección se revisa la estructura y parámetros de las diversas estrategias de control que permitirán su implementación en el microcontrolador elegido.

Para la implementación de las estrategias de control, se diseña un programa general en donde se consideran las posibles variables y constantes del sistema, dejando un bloque específico en donde se cambia dicha estrategia de control.

**Los vectores serán representados con letras minúsculas del castellano y del griego en negritas itálicas, por ejemplo $\mathbf{x}, \mathbf{y}, \boldsymbol{\omega}, \boldsymbol{\sigma} \in \mathbb{R}^n$; las componentes de los vectores son números reales $x_i \in \mathbb{R}$ para $i = 1, 2, \dots, n$ siendo $n \in \mathbb{N}$ la dimensión del espacio vectorial.

**A partir de este punto, por facilidad y abusando de la notación, una función aplicada a un vector tiene la siguiente forma:

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

** Entre las características que debe cumplir una función candidata de Lyapunov son:

$V(\mathbf{x})$ es una función definida positiva.

$\frac{\delta V(\mathbf{x})}{\delta \mathbf{x}}$ es una función continua con respecto a \mathbf{x} .

La derivada con respecto al tiempo $\frac{dV(\mathbf{x})}{dx}$ existe y es una función continua con respecto a \mathbf{x} .

4.1 CONTROL DEL POSICIÓN ARTICULAR

El problema de control de posición o regulación consiste en mover el extremo final del robot manipulador desde cualquier posición inicial hacia una posición deseada. Esto significa que

la i -ésima articulación del robot deberá moverse hacia la respectiva i -ésima posición deseada. Formalmente al objetivo del problema de control de posición está determinado por encontrar una ley de control $\boldsymbol{\tau}$ que proporcione los pares aplicados a las articulaciones o servomotores del robot, de tal forma que la posición actual del robot $\mathbf{q}(t)$ y la velocidad articular de movimiento $\dot{\mathbf{q}}(t)$ tiendan asintóticamente hacia la posición deseada \mathbf{q}_d y la velocidad cero, respectivamente, sin importar las condiciones iniciales [6]. Es decir,

$$\lim_{t \rightarrow \infty} \begin{bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{q}_d \\ 0 \end{bmatrix}.$$

El funcionamiento del programa de control se muestra en el diagrama de flujo de la Figura 4.1, en donde la ecuación de control elegida debe implementarse en el bloque $\boldsymbol{\tau}$.

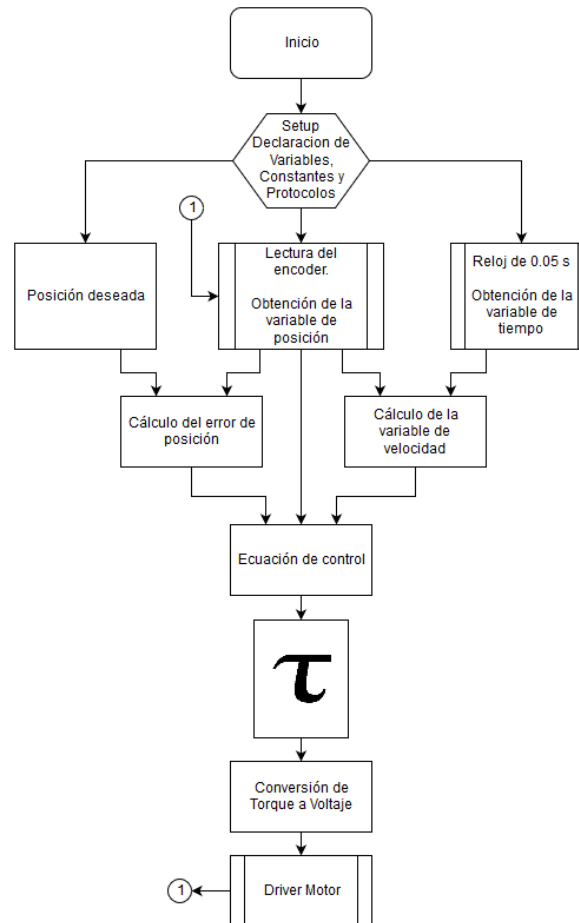


Figura 4.1 Propuesta del programa de control de posición para el ROTRADI VI.

Uno de los esquemas de control más simples y populares es el algoritmo proporcional-derivativo, al cual incluyendo la compensación de gravedad está dado por la siguiente ecuación:

$$\boldsymbol{\tau} = K_p \tilde{\mathbf{q}} - K_v \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (4.1)$$

donde $\tilde{\mathbf{q}}$ es el vector de error de posicionamiento que se define como la diferencia entre la posición deseada \mathbf{q}_d y la posición actual del robot \mathbf{q} , K_p es la ganancia proporcional la cual es una matriz definida positiva, K_v es la ganancia derivativa la cual es una matriz definida positiva[6].

4.1.1 CONTROLADORES PROPUESTOS

A continuación se muestra una serie de controladores propuestos para su implementación.

$$\text{-PD: } \tau = K_p \tilde{\mathbf{q}} - K_v \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})$$

$$\text{-ATAN-ATAN: } \tau = K_p \text{atan}(\tilde{\mathbf{q}}) - K_v \text{atan}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q})$$

$$\text{-TANH-TANH: } \tau = K_p \tanh(\tilde{\mathbf{q}}) - K_v \tanh(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q})$$

4.2 CONTROL DE POSICIÓN CARTESIANO

Con el control de posición articular del robot en funcionamiento, el siguiente paso consiste en el control cartesiano, el cual es el posicionamiento del efector final del robot en base a coordenadas cuadradas $\mathbf{x}, \mathbf{y}, \mathbf{z}$. Para la aplicación del control cartesiano, se deben considerar las siguientes hipótesis [6]:

Existe un vector de posición constante $\mathbf{q}_d \in \mathbb{R}^n$ en el espacio de configuración cartesiana para el cual el vector de error cartesiano se anula, $\boldsymbol{\chi}(\mathbf{q}) = \boldsymbol{\chi}(\mathbf{q}_d)$. Es importante mencionar que no es necesario conocer a \mathbf{q}_d . Esta hipótesis asegura la solución del problema.

El jacobiano del robot es una matriz continua y diferenciable con respecto a cada entrada $\mathbf{q} \in \mathbb{R}^n$ además satisface que es de rango completo, es decir $\text{rango}\{J(\mathbf{q})\} = n$.

Esta hipótesis es un requerimiento técnico en la existencia del punto de equilibrio y del análisis de estabilidad local.

La metodología del jacobiano transpuesto sugerido por Arimoto permite convertir una estructura del control cartesiana en par aplicado a las articulaciones del robot [6], por ejemplo:

$$\tau = J^T(\mathbf{q})\mathbf{f}_\chi, \quad (4.2)$$

donde \mathbf{f}_χ representa la fuerza en coordenadas cartesianas, $J(\mathbf{q})$ es el jacobiano del robot, τ es el par aplicado a las articulaciones del robot.

De esta forma, la técnica de moldeo de energía permite ampliar sus resultados de diseño de algoritmos de control en coordenadas articulares hacia el espacio de tareas, generando una familia extensa de esquemas de control en coordenadas cartesianas.

El problema de posición en coordenadas cartesianas consiste en determinar la ley de control $\tau = J^T(\mathbf{q})\mathbf{f}_x$ tal que suministre los pares aplicados a las articulaciones del robot con el objetivo de mover el extremo final del robot hacia la posición deseada \mathbf{x}_d . El error de posición cartesiano se define como $\tilde{\mathbf{x}} = \mathbf{x}_d - \mathbf{x}$, entonces el objetivo de control es [6]:

$$\lim_{t \rightarrow \infty} \begin{bmatrix} \tilde{\mathbf{x}} \\ \dot{\tilde{\mathbf{x}}} \end{bmatrix} = \mathbf{0} \in \mathbb{R}^{2n}.$$

De igual forma que el control angular de posición, se requiere de una ecuación de control, que traducido al control cartesiano tiene la siguiente forma:

$$\tau = J^T(\mathbf{q})\mathbf{f}_x, \quad (4.3)$$

en donde $\mathbf{f}_x = k_p(\tilde{\mathbf{x}}) - k_v(\dot{\tilde{\mathbf{x}}}) + \mathbf{g}_x$, el cual corresponde a la estructura simple del PD.

Ahora bien, para poder programar la ecuación de control, se requiere del cálculo del jacobiano transpuesto, que en este caso corresponde a:

$$J^T = \begin{pmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{pmatrix},$$

donde

$$J_{11} = -\text{sen}(q_1)(\text{sen}(q_2)l_2 + \text{sen}(q_2 + q_3)l_3) + \text{Cos}(q_1)(w_1 + w_2)$$

$$J_{12} = \text{cos}(q_1)(\text{sen}(q_2)l_2 + \text{sen}(q_2 + q_3)l_3) - \text{sen}(q_1)(w_1 + w_2)$$

$$J_{13} = 0$$

$$J_{21} = \text{cos}(q_1)(\text{cos}(q_2)l_2 + \text{cos}(q_2 + q_3)l_3)$$

$$J_{22} = \text{sen}(q_1)(\text{cos}(q_2)l_2 + \text{cos}(q_2 + q_3)l_3)$$

$$J_{23} = \text{sen}(q_2)l_2 + \text{sen}(q_2 + q_3)l_3$$

$$J_{31} = \text{cos}(q_1)\text{cos}(q_2 + q_3)l_3$$

$$J_{32} = \text{cos}(q_2 + q_3)\text{sen}(q_1)l_3$$

$$J_{33} = \text{sen}(q_2 + q_3)l_3$$

De igual forma los vectores $(\tilde{\mathbf{x}})$ y $(\dot{\tilde{\mathbf{x}}})$ corresponden a:

$$(\tilde{\mathbf{x}}) = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix}, \quad (\dot{\tilde{\mathbf{x}}}) = \begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{y}} \\ \dot{\tilde{z}} \end{bmatrix}. \quad (4.4)$$

4.2.1 CONTROLADORES PROPUESTOS

-PD: $\mathbf{f}_x = k_p(\tilde{\mathbf{x}}) - k_v(\dot{\tilde{\mathbf{x}}}) + \mathbf{g}_x$

-ATAN-ATAN: $\mathbf{f}_x = k_p \text{atan}(\tilde{\chi}) - k_v \text{atan}(\dot{\chi}) + \mathbf{g}_x$

-TANH-TANH: $\mathbf{f}_x = k_p \tanh(\tilde{\chi}) - k_v \tanh(\dot{\chi}) + \mathbf{g}_x$

4.3 CONTROL DE FUERZA

El control de fuerza consiste en lograr que el efector final de un manipulador aplique un nivel de fuerza deseado durante la tarea de interacción programada. Para esto se requiere de un transductor el cual sea capaz de devolver un valor analógico de voltaje proporcional a la fuerza con el cual el robot interacciona con su entorno.

La retroalimentación y el control de la fuerza es una condición obligatoria para lograr un comportamiento robusto y versátil de un sistema robótico en entornos poco o no estructurados, así como para lograr un funcionamiento seguro y confiable de un robot en presencia de seres humanos.

De manera similar al control cartesiano, la ecuación del controlador está dada por:

$$\tau = J^T(\mathbf{q})\mathbf{f}_x, \quad (4.5)$$

en donde $\mathbf{f}_x = k_p(\tilde{\mathbf{f}}) - k_v(\dot{\chi}) + \mathbf{g}_x$, $\tilde{\mathbf{f}} = \mathbf{f}_d - \mathbf{f}_e$,

en donde $\tilde{\mathbf{f}}$ corresponde al error de fuerza, \mathbf{f}_d es el valor de fuerza deseado y \mathbf{f}_e es el valor de fuerza censado en el entorno de interacción.

4.3.1 CONTROLADORES PROPUESTOS

-PD: $\mathbf{f}_x = k_p(\tilde{\mathbf{f}}) - k_v(\dot{\chi}) + \mathbf{g}_x$

-ATAN-ATAN: $\mathbf{f}_x = k_p \text{atan}(\tilde{\mathbf{f}}) - k_v \text{atan}(\dot{\chi}) + \mathbf{g}_x$

-TANH-TANH: $\mathbf{f}_x = k_p \tanh(\tilde{\mathbf{f}}) - k_v \tanh(\dot{\chi}) + \mathbf{g}_x$

4.4 CONTROL DE IMPEDANCIA

Para la tarea de interacción se requiere de la relación de los torques de las articulaciones y las correspondientes fuerzas externas aplicadas. En la representación del espacio articular, el vector \mathbf{F} consiste de componentes de fuerza que operan en el efector final con respecto a un sistema de coordenadas cartesianas. Basado en el principio del trabajo virtual, los torques

articulares $\boldsymbol{\tau} \in \mathbb{R}^n$, necesarios para generar una fuerza $\mathbf{F} \in \mathbb{R}^m$ en el efector final, está dado por la siguiente relación [29]:

$$\mathbf{F}_e = \mathbf{K}_d(\boldsymbol{\chi}_d - \boldsymbol{\chi}) + \mathbf{B}_d(\dot{\boldsymbol{\chi}}_d - \dot{\boldsymbol{\chi}}) + \mathbf{M}_d(\ddot{\boldsymbol{\chi}}_d - \ddot{\boldsymbol{\chi}}) \quad (4.6)$$

donde $\mathbf{M}_d \in \mathbb{R}^{m \times m}$ es la matriz diagonal de inercia, $\mathbf{B}_d \in \mathbb{R}^{m \times m}$ es la matriz diagonal de amortiguamiento y $\mathbf{K}_d \in \mathbb{R}^{m \times m}$ es la matriz diagonal de rigidez del sistema. Dado que $\mathbf{F}_e \equiv \mathbf{0}$ si $(\boldsymbol{\chi}_d - \boldsymbol{\chi}) = \mathbf{0}$, la trayectoria $\boldsymbol{\chi}_d$ es interpretada como una trayectoria sin contacto en el efector final. Cuando el efector final es perturbado por una fuerza externa \mathbf{F}_e (Ecuación 4.6) describe la dinámica de la trayectoria de perturbación.

De la ecuación 4.6, se puede obtener la impedancia mecánica \mathbf{Z}_d dada en el dominio de la frecuencia por

$$\mathbf{Z}_d = \mathbf{K}_d + \mathbf{B}_d + \mathbf{M}_d s^2. \quad (4.7)$$

El enfoque del control de impedancia corresponde a la generalización del control de movimiento en el espacio de trabajo eligiendo la trayectoria deseada $\boldsymbol{\chi}_d$. Basado en la ecuación 4.6 dicha trayectoria tratara de seguir la siguiente relación dinámica,

$$\boldsymbol{\chi}_d - \boldsymbol{\chi} = (\mathbf{K} + \mathbf{B}s + \mathbf{M}s^2)^{-1} \mathbf{F}_e \quad (4.8)$$

donde $s = \frac{d}{dt}$ denota el operador diferencial y \mathbf{K} , \mathbf{B} y $\mathbf{M} \in \mathbb{R}^{m \times m} > \mathbf{0}$ son diagonales matrices. Entonces $(\mathbf{K} + \mathbf{B}s + \mathbf{M}s^2)^{-1}$ es un filtro lineal estable de segundo orden [30].

Se puede definir el error de impedancia $\tilde{\boldsymbol{\xi}} \in \mathbb{R}^m$ como

$$\tilde{\boldsymbol{\xi}} = \tilde{\boldsymbol{\chi}} - \dot{\tilde{\boldsymbol{\chi}}} \quad (4.9)$$

donde

$$\tilde{\boldsymbol{\chi}} = \boldsymbol{\chi}_d - \boldsymbol{\chi} \quad (4.10)$$

$$\dot{\tilde{\boldsymbol{\chi}}} = (\mathbf{K}_d + \mathbf{B}_d s + \mathbf{M}_d s^2)^{-1} \mathbf{F}_e \quad (4.11)$$

Formalmente el objetivo de nuestro enfoque al control de impedancia consiste en seleccionar un $\boldsymbol{\tau}$ tal que:

$$\lim_{t \rightarrow \infty} \begin{bmatrix} \tilde{\boldsymbol{\xi}} \\ \dot{\tilde{\boldsymbol{\xi}}} \end{bmatrix} = \mathbf{0}.$$

Nótese que en ausencia del contacto con fuerzas exteriores $\mathbf{F}_e \equiv \mathbf{0}$, el objetivo de control de impedancia es equivalente al objetivo del control en un espacio de trabajo. Por lo tanto este enfoque de control de impedancia puede ser visto como un control de movimiento permitiendo una tolerancia al error en el seguimiento de la trayectoria deseada en presencia de fuerzas de interacción.

Considerando la definición del error de impedancia (Ecuación 4.9), tenemos que:

$$\dot{\xi} = \dot{\chi}_d - \dot{\chi} - s(\mathbf{K}_d + \mathbf{B}_d s + \mathbf{M}_d s^2)^{-1} \mathbf{F}_e \quad (4.12)$$

$$\ddot{\xi} = \ddot{\chi}_d - \ddot{\chi} - s^2(\mathbf{K}_d + \mathbf{B}_d s + \mathbf{M}_d s^2)^{-1} \mathbf{F}_e \quad (4.13)$$

que representan la primera y segunda derivada temporal del error de impedancia respectivamente.

Para propósitos de la tesis, es necesario remarcar el hecho de que no es posible calcular la impedancia de forma directa ya que no se cuenta con un sensor que haga dicha tarea, no obstante como se ha mencionado, en ausencia de fuerzas de contacto, la acción del control de impedancia equivale a un control de posicionamiento, así mismo se puede ver el control de fuerza como un control de impedancia sobre un punto de posicionamiento donde se ejerce determinada fuerza. Tomando esto en cuenta tenemos que para el control de impedancia se deben ejercer dos controladores simultáneos para cumplir con las condiciones del control de impedancia que corresponden al control de una trayectoria fuerza que se describe por la expresión 4.14.

$$I \cong F(t) / \chi(t) \quad (4.14)$$

en donde $F(t)$ corresponde al controlador de fuerza, mientras que $q(t)$ corresponde al controlador de posición. Y a su vez se tiene que $F = k\chi$ siendo k una constante de elasticidad y χ un vector de posición cartesiano.

Adicionalmente se tiene que

$$\tilde{I} = I_d - I \cong F_d(t) / q_d(t) - F(t) / q(t) \quad (4.15)$$

Tomando en cuenta las expresiones 4.14 y 4.15, tenemos que la estructura de la ecuación de control general para el programa de control de impedancia completa utilizando las variables disponibles del programa de control es:

$$f_\chi = k_{p1}(\chi_d - \chi) - k_{v1}\dot{\chi} + u(k_{p2}(\mathbf{F}_d - \mathbf{F}_e) - k_{v2}k\dot{\chi}) + \mathbf{g}_x$$

siendo u una matriz diagonal compuesta por 1's y 0's que indican las direcciones cartesianas hacia donde se controla la fuerza de contacto.

4.4.1 CONTROLADORES PROPUESTOS

-PD: $f_\chi = k_p(\tilde{I}) - k_v(\dot{\tilde{I}}) + \mathbf{g}_x$

-ATAN-ATAN: $f_\chi = k_p \text{atan}(\tilde{I}) - k_v \text{atan}(\dot{\tilde{I}}) + \mathbf{g}_x$

-TANH-TANH: $\mathbf{f}_x = k_p \tanh(\tilde{I}) - k_v \tanh(\dot{\tilde{I}}) + \mathbf{g}_x$

4.5 RUTINA DE CONTROL FISIOTERAPÉUTICA

Finalmente se implementa una rutina de fisioterapia el cual consiste en la programación de una trayectoria \mathbf{x}_d o \mathbf{q}_d a través del movimiento manual del brazo, el cual guarda una serie de posiciones $\mathbf{x}_d^T = [X_i, Y_i, Z_i]$ o $\mathbf{q}_d^T = [q_{1_i}, q_{2_i}, q_{3_i}]$ con respecto al tiempo, mismas que son utilizadas más tarde para generar una reproducción del movimiento previamente guardado.

Este programa está diseñado a partir del programa de impedancia y de un segundo programa en donde solo se implementa una salida de torque que corresponde al par gravitacional.

El funcionamiento del programa consiste en el movimiento del robot aplicando únicamente el par gravitacional en donde el movimiento manual de los eslabones del robot permitirá guardar una serie de datos de posicionamiento con respecto al tiempo, estos datos son guardados en una matriz e interpretada más tarde como una trayectoria conformada por posiciones deseadas en el tiempo, este funcionamiento se ilustra en la Figura 4.2.

La rutina de fisioterapia se propone principalmente como una forma práctica de la aplicación del control de robots manipuladores, y como una forma de relacionar la robótica como una posible opción para la aplicación de rutinas usadas en medicina.

El funcionamiento de este programa posee la cualidad de que el usuario no requiere de un conocimiento avanzado de programación, solo se requiere del movimiento manual por parte del médico según sea requerido en el ejercicio a realizar, acto seguido de la aplicación de la rutina con el brazo o extremidad deseada del paciente acoplada al robot.

**Para ver la ejecución del programa, visitar los siguientes enlaces.

<https://youtu.be/IHF2TAdgf0>

Buscar en YouTube bajo el nombre de “Programación manual del ROTRADI VI”

<https://youtu.be/RPTT7ObsIrY>

Buscar en YouTube bajo el nombre de “Ejecución del programa manual del ROTRADI VI”

**Para ver las pruebas preliminares aplicadas en un péndulo de transmisión directa, ver el Apéndice C.

**Para el programa de control en Arduino, ver el Apéndice D.

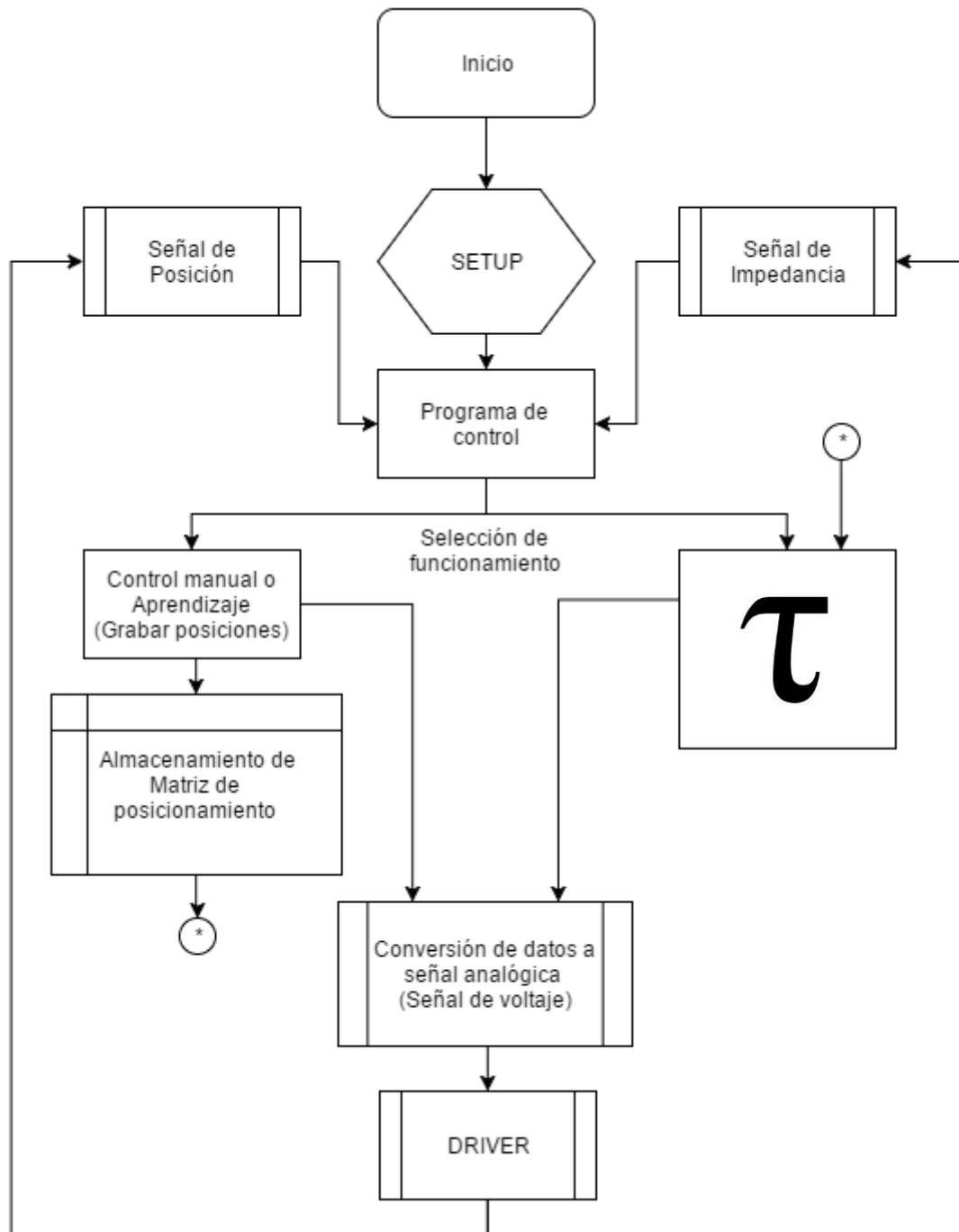


Figura 4.2: Diagrama de flujo del programa de control para fisioterapia.

CAPITULO 5: RESULTADOS

En este capítulo se exponen los resultados obtenidos para cada uno de los controladores implementados: control de posición angular, cartesiano, de fuerza y control de impedancia. Así, se muestran diversas variaciones de los controladores para cada estrategia de control y los resultados que estos arrojan. Para cada prueba se selecciona un valor deseado \mathbf{q}_d , \mathbf{x}_d y \mathbf{f}_d , se realizaron 30 repeticiones del mismo experimento en cada caso, según la norma MSA usada en industria y finalmente a partir de los datos obtenidos se calcula el promedio de \bar{q}_i , \bar{x} y \bar{f} para conocer el comportamiento de cada controlador y hacer una comparación de los controladores en general.

5.1 RESULTADOS DE CONTROL DE POSICIÓN ANGULAR

En esta sección se presentan los resultados obtenidos, las gráficas del par aplicado y de su posición angular con respecto al tiempo de cada uno de los controladores utilizados en las pruebas:

$$\text{-PD: } \tau = K_p \tilde{\mathbf{q}} - K_v \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}),$$

$$\text{-ATAN-ATAN: } \tau = K_p \text{atan}(\tilde{\mathbf{q}}) - K_v \text{atan}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}),$$

$$\text{-TANH-TANH: } \tau = K_p \tanh(\tilde{\mathbf{q}}) - K_v \tanh(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}).$$

En este caso $q_{d1}, q_{d2}, q_{d3} = 45^\circ$.

Controlador PD

Para este controlador los resultados se pueden apreciar en la Figura 5.1 y 5.2, en donde el error de posición promedio fue de $\bar{q}_1 = 1.94$, $\bar{q}_2 = 0.74$, $\bar{q}_3 = 0.56$ grados.

Este primer controlador es una prueba general para el control articular, esto es, para comparar el desempeño de otros controladores con respecto al PD.

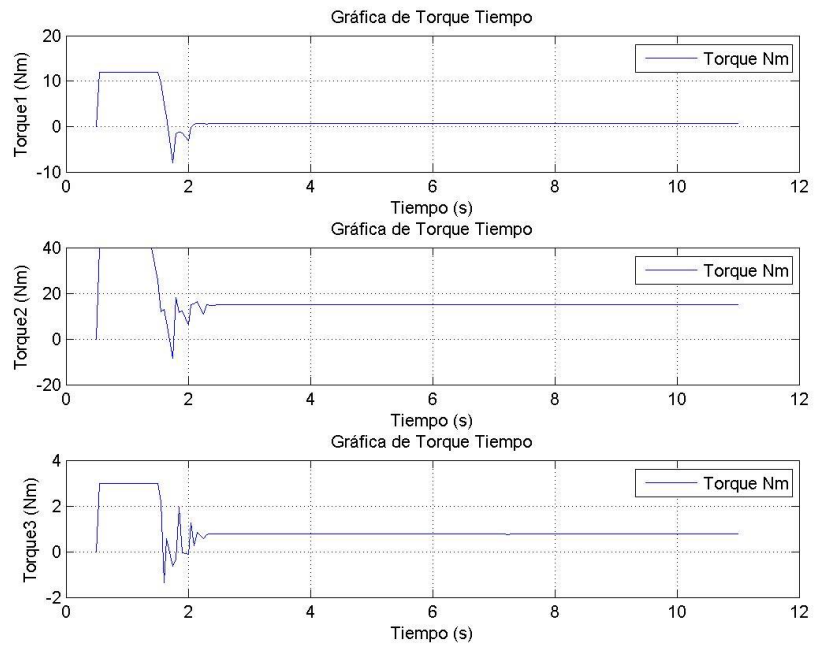


Figura 5.1: Gráfica de torque con respecto al tiempo para el controlador PD.

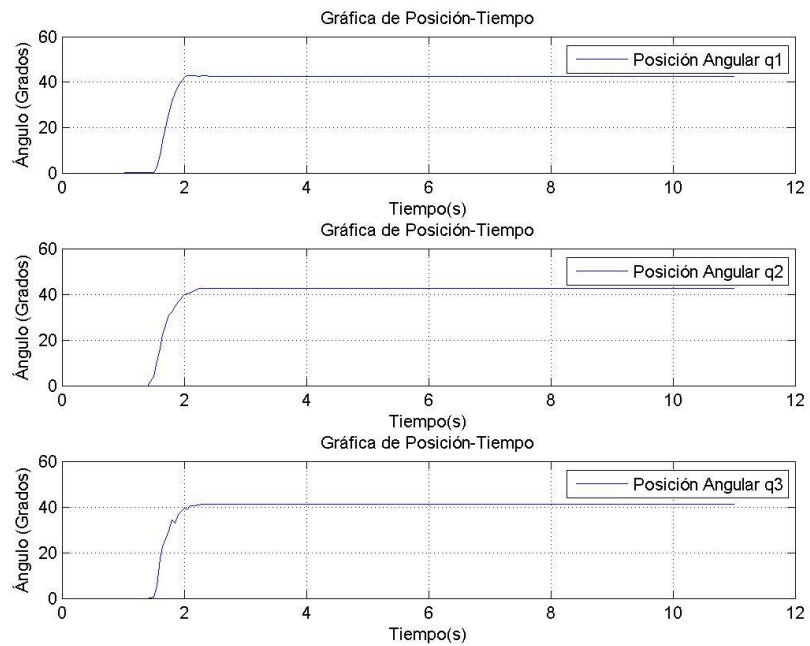


Figura 5.2: Gráfica de posición con respecto al tiempo para el controlador PD.

Controlador ATAN-ATAN

Para este controlador los resultados se pueden apreciar en la Figura 5.3 y 5.4, en donde el error de posición promedio fue de $\bar{q}_1 = -0.53$, $\bar{q}_2 = 0.06$, $\bar{q}_3 = -0.49$ grados.

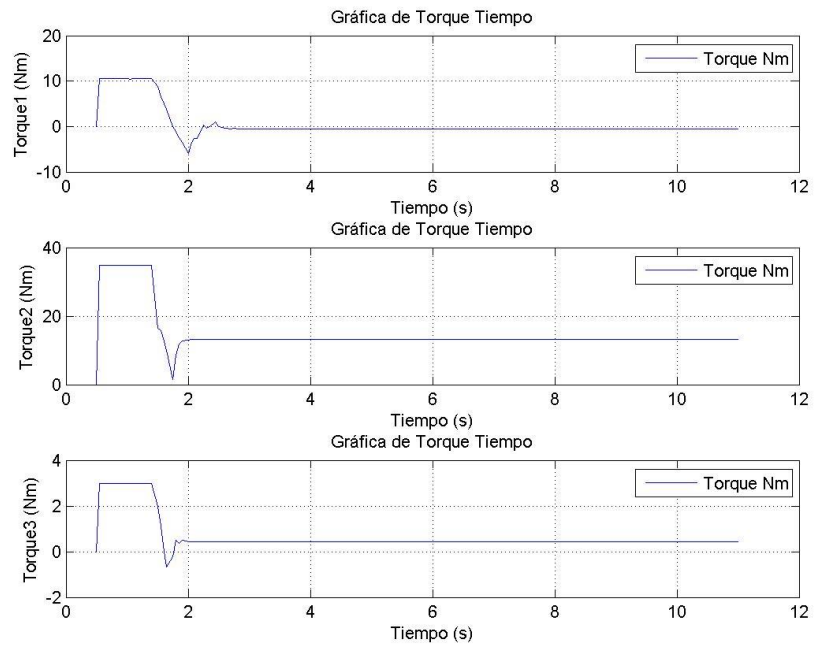


Figura 5.3: Gráfica de torque con respecto al tiempo para el controlador ATAN-ATAN.

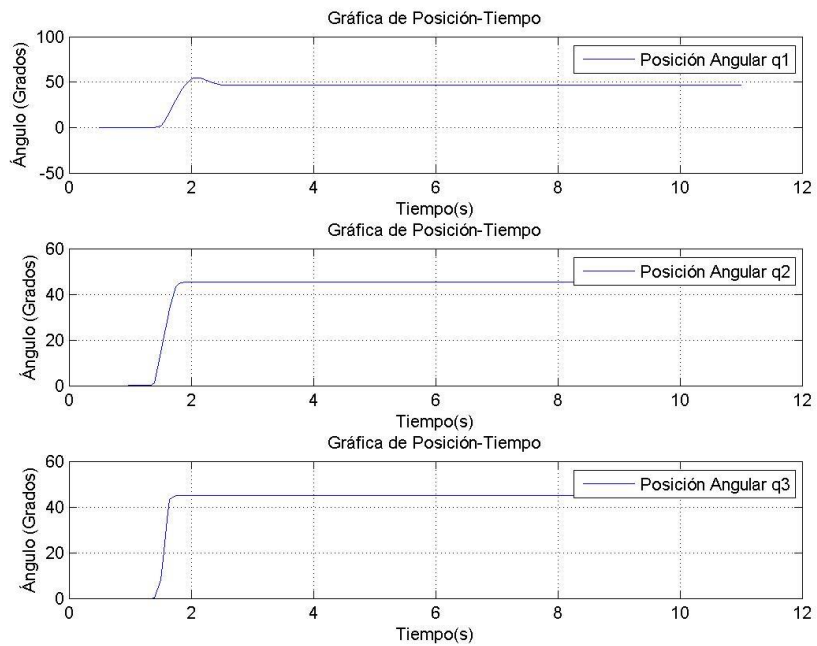


Figura 5.4: Gráfica de posición con respecto al tiempo para el controlador ATAN-ATAN.

Para este controlador se nota por las gráficas 5.3 y 5.4 una menor cantidad de picos en cuanto a cambios de torque, permitiendo un movimiento más directo y estable en comparación con el comportamiento del PD el cual mostro gran cantidad de vibración antes de encontrar una posición estable.

Controlador TANH-TANH

Para este controlador los resultados se pueden apreciar en la Figura 5.5 y 5.6, en donde el error de posición promedio fue de $\bar{q}_1 = -0.21$, $\bar{q}_2 = 0.18$, $\bar{q}_3 = -0.07$ grados.

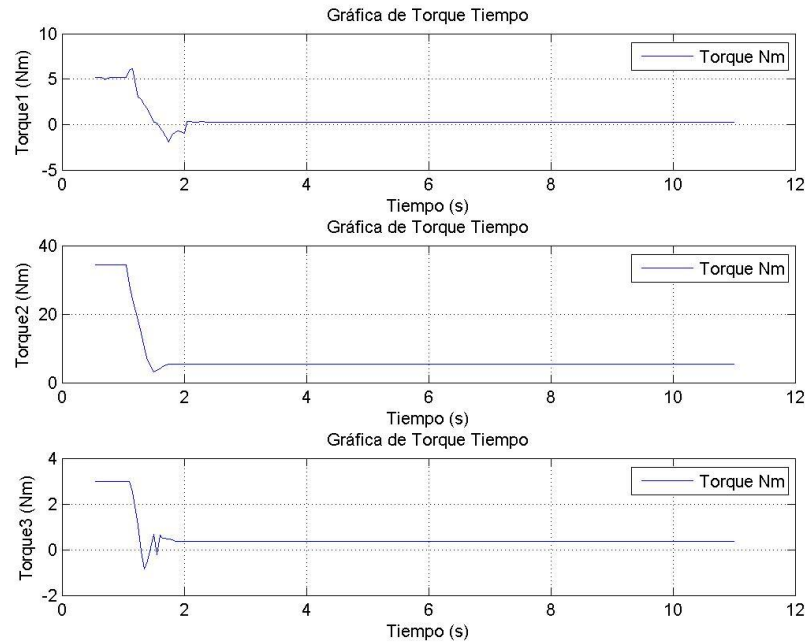


Figura 5.5: Gráfica de torque con respecto al tiempo para el controlador TANH-TANH.

Al igual que el controlador ATAN-ATAN, el controlador TANH-TANH presenta un comportamiento estable y rápido en comparación con el PD, nótese que a diferencia del controlador ATAN-ATAN, para este controlador se presentan sobre impulsos en los eslabones 2 y 3 antes de alcanzar el valor de posición deseado.

A continuación en las tablas 5.1-5.3 comparando el error de posición obtenida en cada uno de los controladores implementados y en la tabla 5.4 los valores k_p y k_v implementados.

TABLA 5.1 TABLA COMPARATIVA DEL PROMEDIO DEL ERROR.

Controlador	\bar{q} (grados)
PD	
q_1	1.94
q_2	0.74
q_3	0.56

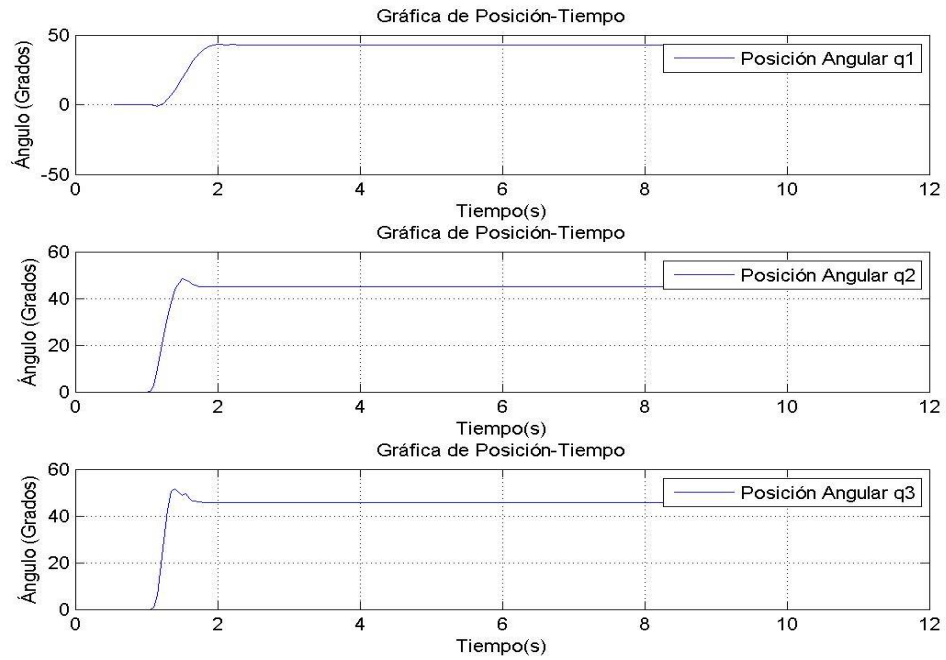


Figura 5.6: Gráfica de posición con respecto al tiempo para el controlador TANH-TANH.

TABLA 5.2 TABLA COMPARATIVA DEL PROMEDIO DEL ERROR.

Controlador ATAN-ATAN	\tilde{q} (grados)
q_1	-0.53
q_2	0.06
q_3	-0.49

TABLA 5.3 TABLA COMPARATIVA DEL PROMEDIO DEL ERROR.

Controlador TANH-TANH	\tilde{q} (grados)
q_1	-0.21
q_2	0.18
q_3	-0.07

TABLA 5.4 TABLA DE VALORES k_p Y k_v .

Variable	Valor
k_{p1}	$(0.80 * 10.95)1.88e^{-0.008*abs(qd1)}$
k_{v1}	$0.3k_{p1}$
k_{p2}	$(0.80 * 36.5)1.88e^{-0.008*abs(qd2)}$
k_{v2}	$0.2k_{p2}$
k_{p3}	$(0.80 * 2.92)1.88e^{-0.008*abs(qd3)}$
k_{v3}	$0.2k_{p3}$

El desempeño de los controladores con estructura atan y tanh a través del sistema propuesto es eficaz con errores de posicionamiento menores al de un grado, en comparación con la estructura PD cuyo valor de error puede llegar a sobrepasar el grado. Cabe mencionar que este error puede ser reducido a través de una buena sintonización de las constantes k_p y k_v .

5.2 RESULTADOS DE CONTROL CARTESIANO

En esta sección se presentan los resultados obtenidos del control cartesiano implementado en el ROTRADI VI. Las figuras 5.7-5.12 muestran las gráficas del par aplicado y de su posición cartesiana con respecto al tiempo, para cada uno de los controladores aplicados en las pruebas:

-PD: $\mathbf{f}_x = k_p(\tilde{\mathbf{x}}) - k_v(\dot{\mathbf{x}}) + \mathbf{g}_x$,

-ATAN-ATAN: $\mathbf{f}_x = k_p \text{atan}(\tilde{\mathbf{x}}) - k_v \text{atan}(\dot{\mathbf{x}}) + \mathbf{g}_x$,

-TANH-TANH: $\mathbf{f}_x = k_p \tanh(\tilde{\mathbf{x}}) - k_v \tanh(\dot{\mathbf{x}}) + \mathbf{g}_x$.

Para cada prueba se seleccionó un valor de posicionamiento deseado, en este caso ese valor corresponde a $\mathbf{x}_d^T = [0.6, 0, 1.0]$.

Controlador PD

Para este primer controlador como se muestra en las gráficas de las Figuras 5.7 y 5.8 el sistema tuvo un comportamiento aceptable, dando un error de $\tilde{\mathbf{x}}^T = [0.08, -0.01, 0.08]$.

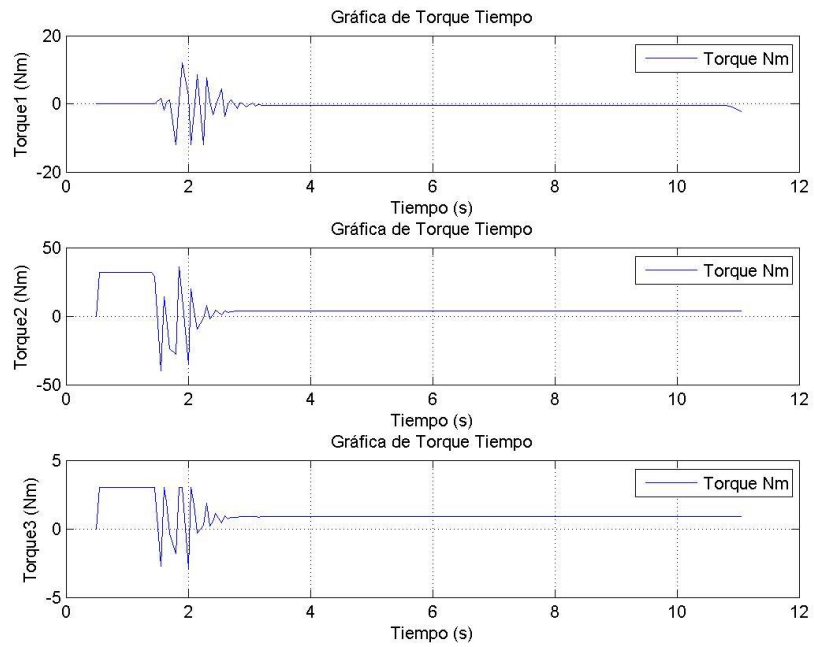


Figura 5.7: Grafica de torque con respecto al tiempo para el controlador PD cartesiano.

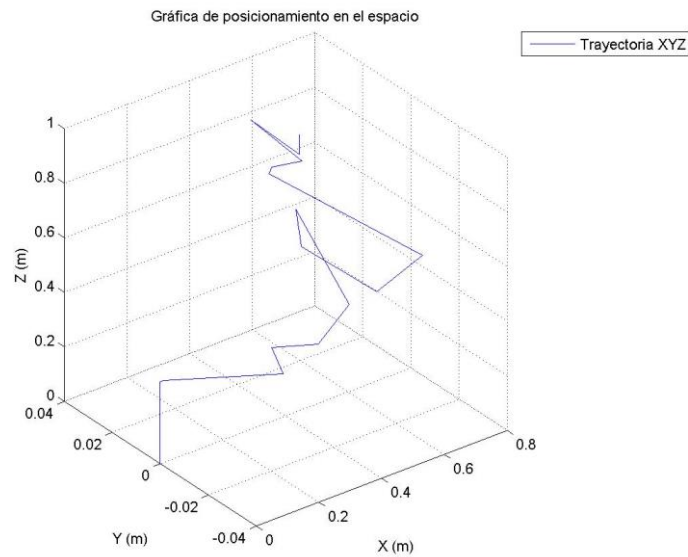


Figura 5.8: Grafica de la trayectoria del efector final en el espacio para el controlador PD cartesiano.

Para este controlador se pudo notar una mayor vibración por parte de los actuadores en comparación con los controladores de posición articular, también hay que mencionar para el caso de control cartesiano, los actuadores pueden tener más de una combinación articular para que el efector final alcance la zona apuntada.

Controlador ATAN-ATAN

Como se muestra en las Figuras 5.9 y 5.10, el sistema tuvo un comportamiento más estable en comparación con la estructura PD. Así, se presentó una trayectoria con $\bar{\chi}^T = [0.02, -0.01, 0.04]$. Además, se observa una clara respuesta con un error menor con respecto a la estructura PD, aunque en la implementación se observó una pequeña vibración en el momento final cuando se alcanza el valor de la posición deseada; esta vibración corresponde principalmente a la respuesta que este tiene por las ganancias k_p y k_v elegidas para ese actuador.

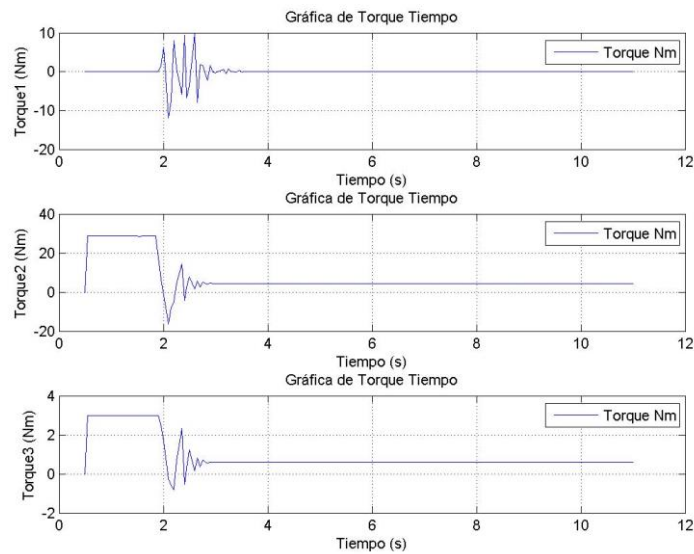


Figura 5.9: Gráfica de torque con respecto al tiempo para el controlador ATAN-ATAN cartesiano.

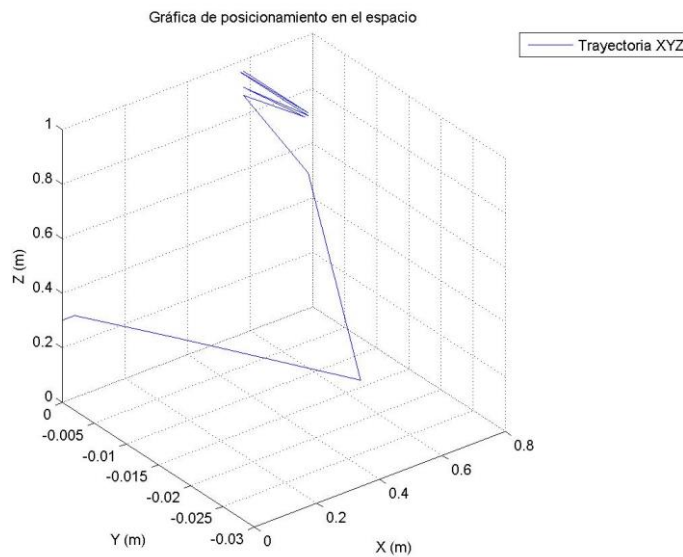


Figura 5.10: Gráfica de la trayectoria del efector final en el espacio para el controlador ATAN-ATAN cartesiano.

Controlador TANH-TANH

Para este controlador como se muestra en las Figuras 5.11 y 5.12 el sistema tuvo un comportamiento menos estable en comparación con la estructura ATAN-ATAN, así mismo se presentó una trayectoria con más vibración, sin embargo el error final fue el menor de los controladores propuestos dando un $\bar{\chi}^T = [0.00, 0.00, 0.03]$.

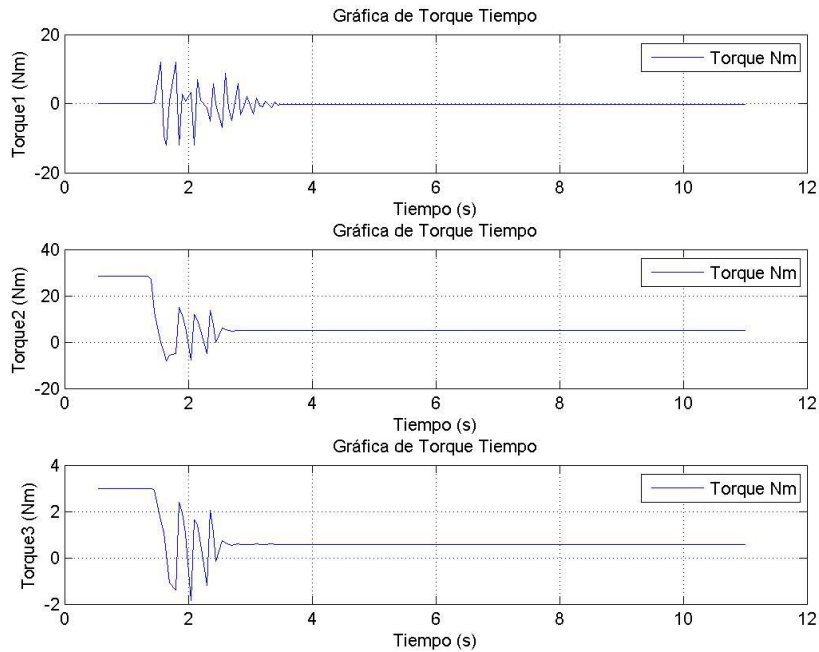


Figura 5.11: Gráfica de torque con respecto al tiempo para el controlador TANH-TANH cartesiano.

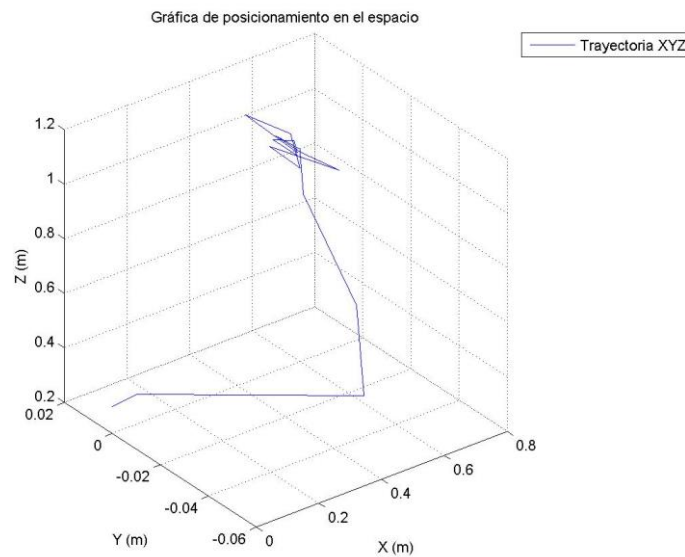


Figura 5.12: Gráfica de la trayectoria del efector final en el espacio para el controlador TANH-TANH cartesiano.

Para estos experimentos, el controlador TANH-TANH fue el controlador con la mejor respuesta de movimiento, siendo el que presento un menor error de posicionamiento cartesiano además del hecho de tener un comportamiento similar al ATAN-ATAN. Al igual que los controladores articulares, el comportamiento de los algoritmos de control puede mejorarse con una buena sintonización de las ganancias k_p y k_v . A continuación se muestran en las tablas 5.5-5.7 la comparación del error de posición obtenida en cada uno de los controladores y en la tabla 5.8 los valores k_p y k_v implementados.

TABLA 5.5 TABLA COMPARATIVA DEL ERROR.

Controlador PD	$\tilde{\chi}$ (metros)
X	0.08
Y	-0.01
Z	0.04

TABLA 5.6 TABLA COMPARATIVA DEL ERROR.

Controlador ATAN-ATAN	$\tilde{\chi}$ (metros)
X	0.02
Y	-0.01
Z	0.04

TABLA 5.7 TABLA COMPARATIVA DEL ERROR.

Controlador TANH-TANH	$\tilde{\chi}$ (metros)
X	0.0
Y	0.0
Z	0.03

TABLA 5.8 TABLA DE VALORES k_p Y k_v .

Variable	Valor
k_{p1}	100
k_{v1}	50
k_{p2}	60
k_{v2}	20
k_{p3}	15
k_{v3}	3

Algunas observaciones importantes de la estrategia de control cartesiano, es el hecho de que la posición deseada puede tener más de una combinación en coordenadas angulares que lleve el efector final a la posición deseada, esto debe tomarse en cuenta según la aplicación deseada, ya que aunque la posición deseada sea alcanzada, el movimiento de los actuadores podría no ser la deseada, ejemplo de esto es una rutina de fisioterapia en donde el brazo del paciente estuviera sujeto a los eslabones del robot, donde no convendría que doblara el brazo del paciente en ángulos que pudieran fracturar su brazo.

5.3 RESULTADOS DE CONTROL DE FUERZA

Para la aplicación de cualquier controlador de fuerza, hay que remarcar el hecho de que se requiere de un entorno con el cual interactuar, a modo de que el entorno de interacción sirva a modo de barrera para el robot, para esto se requiere que el efector final este cerca del área apuntada, en este caso un muro de acrílico, por lo que en primera instancia es necesario aplicar un control de posicionamiento el cual realice la primera tarea (control de movimiento) y acto seguido del contacto entre el sensor y la superficie de interacción (muro), los controladores aplicados son:

$$\text{-PD: } \mathbf{f}_x = k_p(\tilde{\mathbf{f}}) - k_v(\dot{\mathbf{x}}) + \mathbf{g}_x$$

$$\text{-ATAN-ATAN: } \mathbf{f}_x = k_p \text{atan}(\tilde{\mathbf{f}}) - k_v \text{atan}(\dot{\mathbf{x}}) + \mathbf{g}_x$$

$$\text{-TANH-TANH: } \mathbf{f}_x = k_p \tanh(\tilde{\mathbf{f}}) - k_v \tanh(\dot{\mathbf{x}}) + \mathbf{g}_x$$

Para cada caso el valor deseado para el control de fuerza que se eligió fue $\mathbf{f}_d^T = [6,0,0]$. A groso modo el objetivo de esta tarea es empujar la barrera de tal modo que la fuerza aplicada sobre la superficie de interacción sea cercana al valor deseado.

Controlador PD

Como prueba preliminar se utiliza la estructura PD, en donde tras ser posicionado el brazo, se procede a la aplicación de fuerza sobre una superficie como se muestra en la Figura 5.13.

Como se puede notar, el comportamiento de la fuerza aplicada no se estabiliza en ningún momento, esto puede deberse a que el sensor utilizado no es de precisión, sin embargo el promedio de la fuerza aplicada es de $\bar{\mathbf{f}}^T = [5.24,0,0]$.

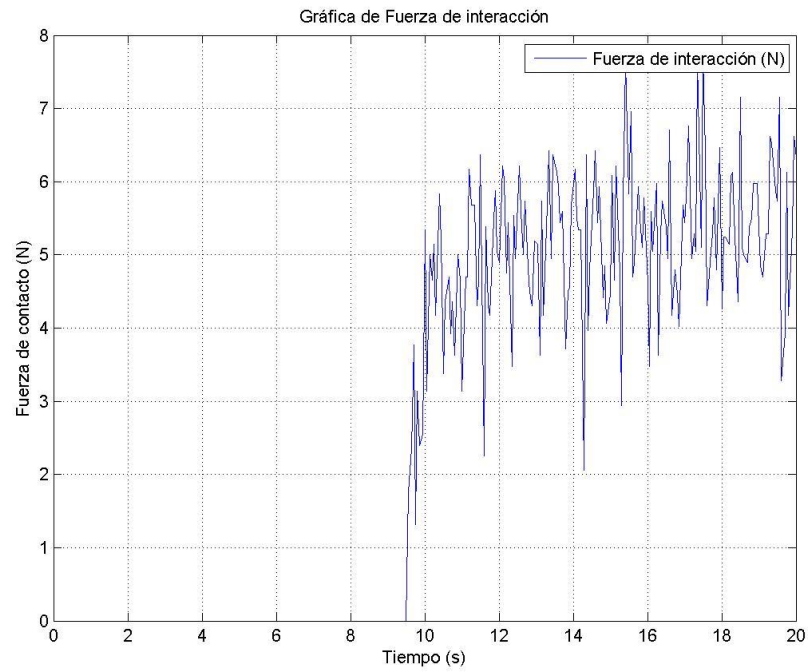


Figura 5.13: Grafica de la fuerza de interacción para el controlador PD.

Controlador ATAN-ATAN

Procediendo con siguiente controlador, se aplican las mismas condiciones y se obtienen los siguientes resultados denotados en la Figura 5.14.

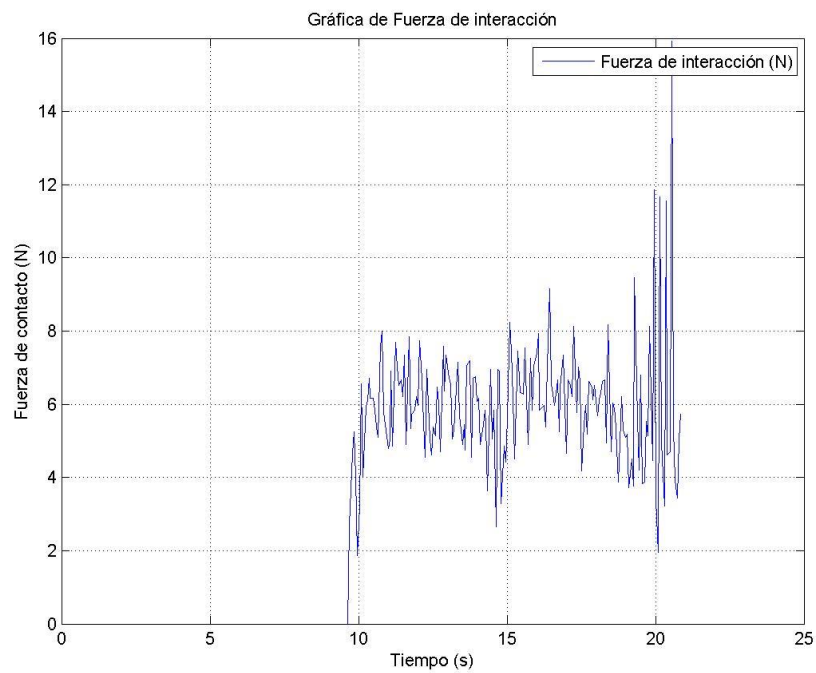


Figura 5.14: Grafica de la fuerza de interacción para el controlador ATAN-ATAN.

Al igual que en el controlador PD, el comportamiento de la fuerza aplicada no se estabiliza en ningún momento, sin embargo el promedio de la fuerza de interacción aplicada a la superficie es de $\bar{\mathbf{f}}^T = [6.21, 0, 0]$.

Controlador TANH-TANH

Finalmente se implementa el último controlador, se aplican las mismas condiciones y se obtienen los siguientes resultados denotados en la Figura 5.15.

Para este controlador, al igual que los otros dos controladores, no se estabiliza y su promedio de fuerza de interacción es de $\bar{\mathbf{f}}^T = [6.62, 0, 0]$.

A continuación se muestra una tabla con los valores k_p y k_v utilizados para el control de fuerza.

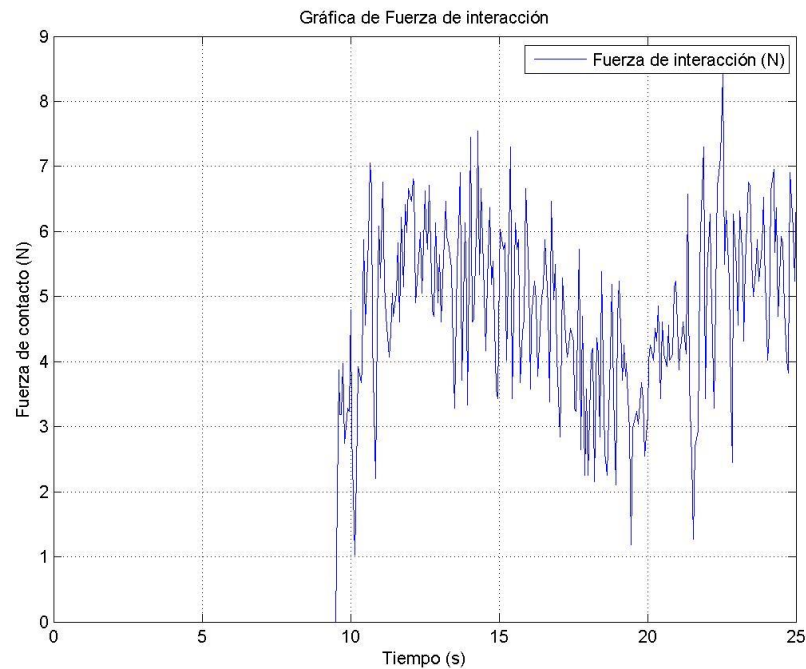


Figura 5.15: Grafica de la fuerza de interacción para el controlador TANH-TANH.

En el control de fuerza se pudo observar un comportamiento poco estable en comparación con los controladores de posición articular y control cartesiano esto principalmente a las características del sensor aplicado, que si bien no comparables con la robustez de los componentes del ROTRADI VI, sirve como punto de referencia para futuras aplicaciones con sensores más especializados. Finalmente hay que reiterar que la respuesta de los controladores puede ser mejorada con una buena sintonización de los valores k_p y k_v .

A continuación en la tabla 5.9 se muestran los valores k_p y k_v implementados

TABLA 5.9 TABLA DE VALORES k_p Y k_v .

Variable	Valor
k_{p1}	25
k_{v1}	4.5
k_{p2}	25
k_{v2}	4.5
k_{p3}	9
k_{v3}	5

5.4 RESULTADOS DE CONTROL DE IMPEDANCIA

Para los controladores de impedancia en las Figuras 5.16-5.24 se presentan graficas en donde se aplica el control de impedancia, involucrando una trayectoria de forma rectangular obtenida mediante la programación “manual” o de aprendizaje aplicada sobre una superficie cuya fuerza deseada de contacto corresponde a $f_d^T = [4,0,0]$. La superficie de contacto es la misma que se utilizó en los experimentos de fuerza. Los controladores propuestos son:

$$\text{-PD: } \mathbf{f}_x = k_p(\tilde{I}) - k_v(\dot{\tilde{I}}) + \mathbf{g}_x$$

$$\text{-ATAN-ATAN: } \mathbf{f}_x = k_p \text{atan}(\tilde{I}) - k_v \text{atan}(\dot{\tilde{I}}) + \mathbf{g}_x$$

$$\text{-TANH-TANH: } \mathbf{f}_x = k_p \tanh(\tilde{I}) - k_v \tanh(\dot{\tilde{I}}) + \mathbf{g}_x$$

Controlador PD

En primera instancia se aplica el controlador PD, en donde se muestra el comportamiento del mismo en las gráficas de las Figuras 5.16 a la 5.18, en donde se puede apreciar la trayectoria del efector final, la fuerza de interacción y el comportamiento del torque con respecto al tiempo de cada uno de los actuadores.

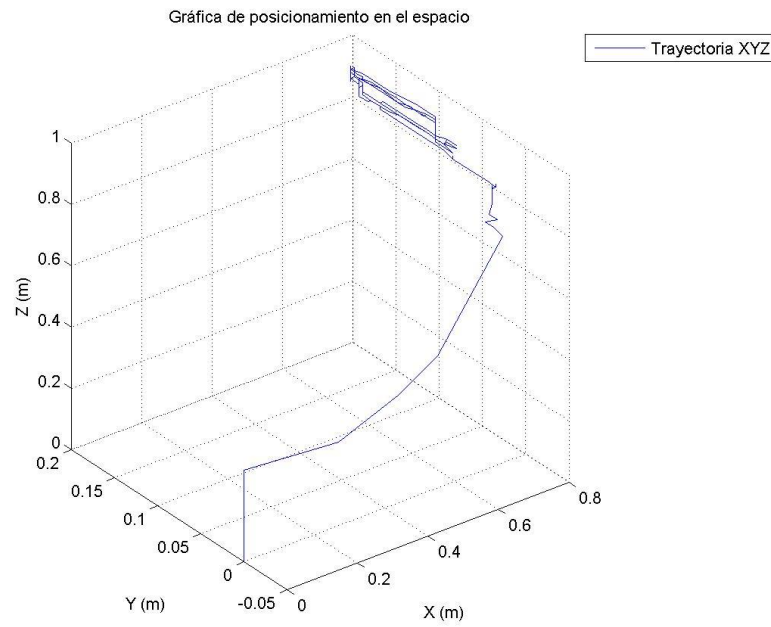


Figura 5.16: Gráfica de la trayectoria de interacción para el controlador PD de impedancia.

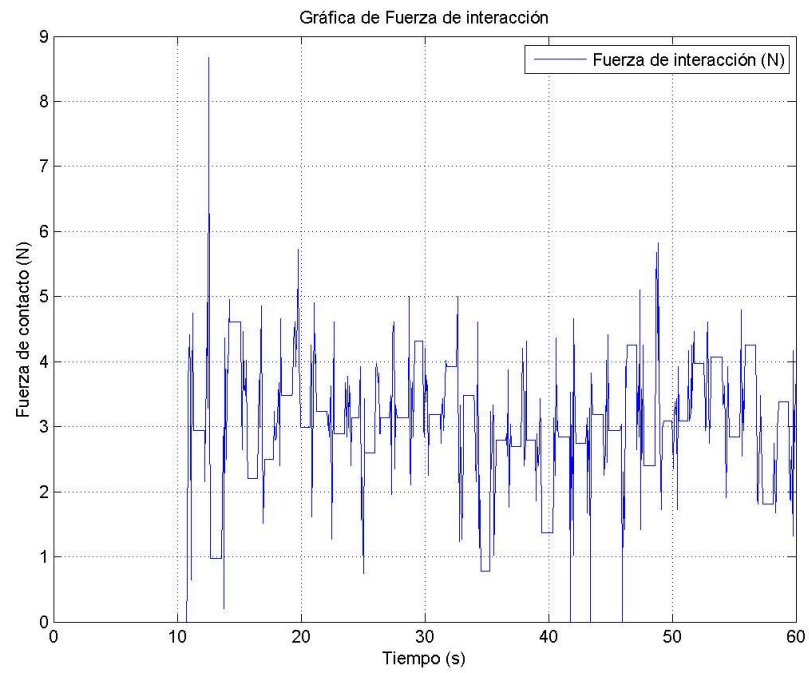


Figura 5.17: Gráfica de la fuerza de interacción para el controlador PD de impedancia.

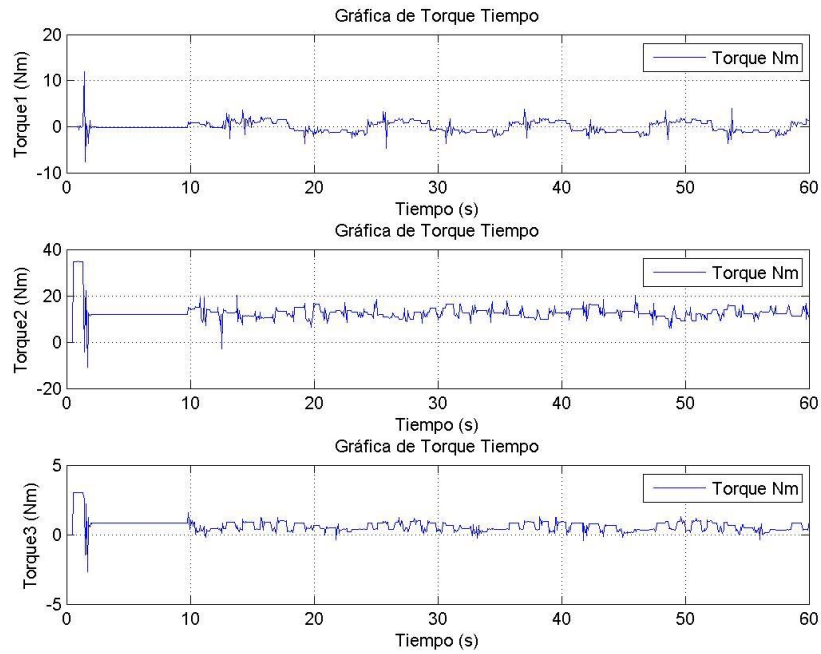


Figura 5.18: Grafica de torque con respecto al tiempo para el controlador PD de impedancia.

Controlador ATAN-ATAN

A continuación se presenta el segundo controlador de impedancia, en donde se aplicaron las mismas condiciones del controlador PD y cuyo comportamiento se muestra en las Figuras 5.19 a la 5.21.

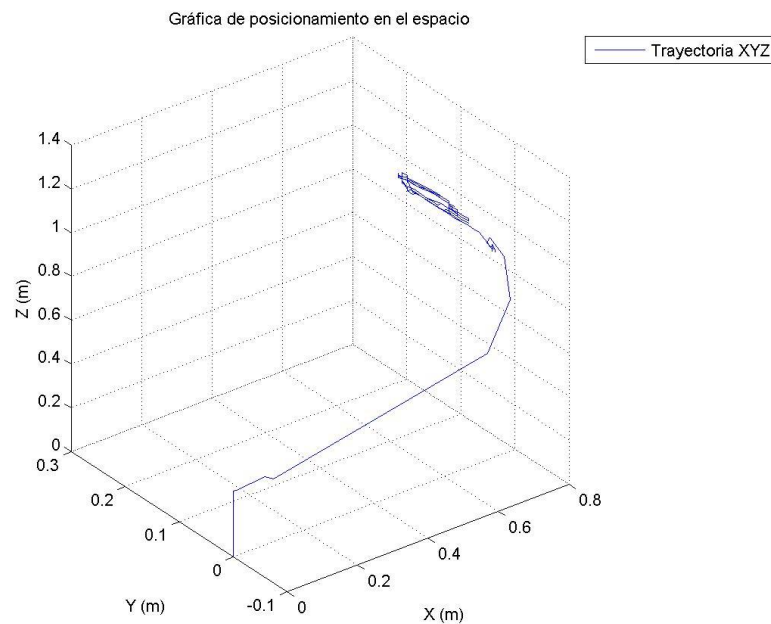


Figura 5.19: Grafica de la trayectoria para el controlador ATAN-ATAN de impedancia.

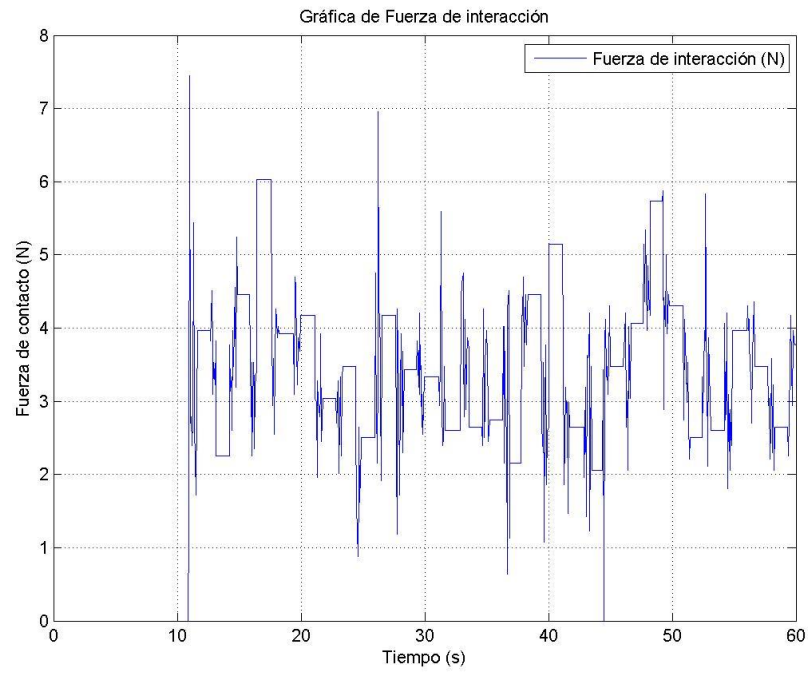


Figura 5.20: Gráfica de la fuerza de interacción para el controlador ATAN-ATAN de impedancia.

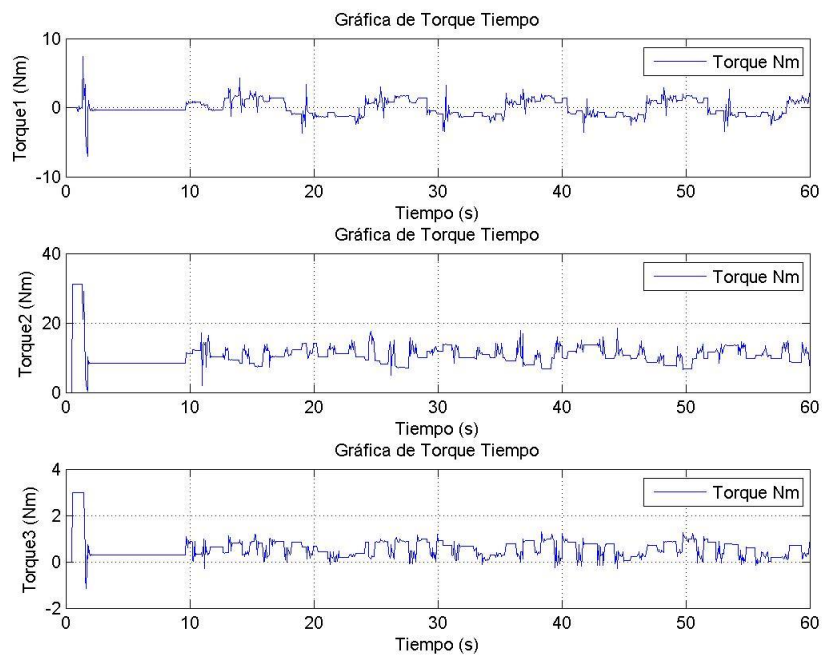


Figura 5.21: Gráfica de torque con respecto al tiempo para el controlador ATAN-ATAN de impedancia.

Controlador TANH-TANH

Finalmente se implementa el último controlador bajo las mismas condiciones cuya respuesta se muestra a continuación en las Figuras 5.22 a la 5.24.

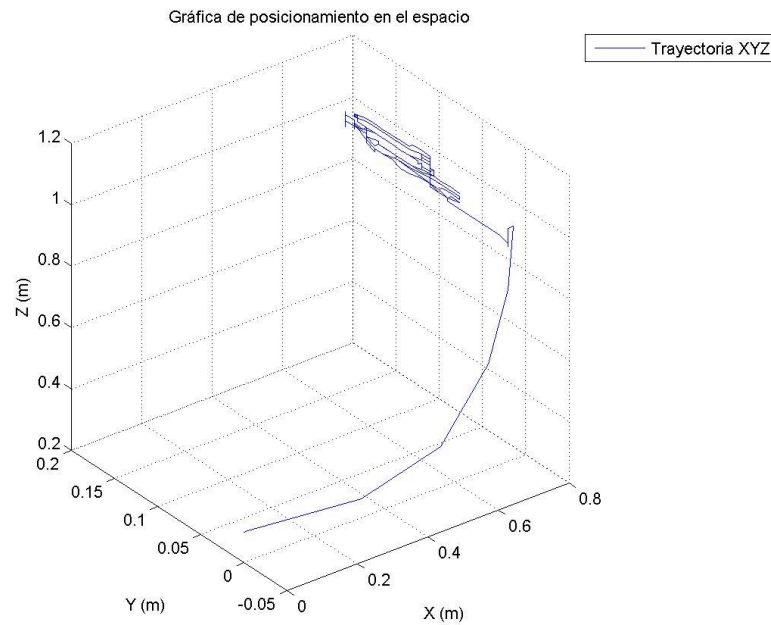


Figura 5.22: Gráfica de la trayectoria para el controlador TANH-TANH de impedancia.

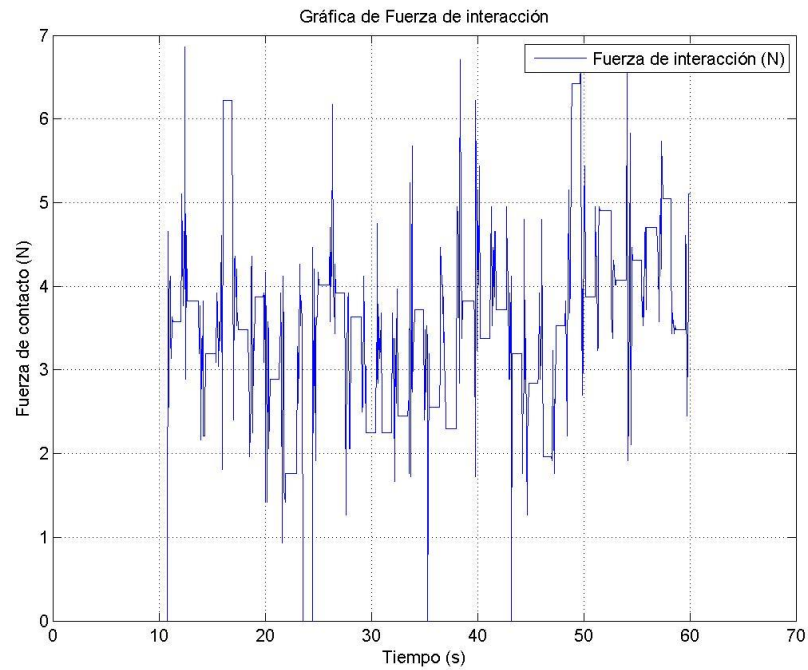


Figura 5.23: Gráfica de la fuerza de interacción para el controlador ATAN-ATAN de impedancia.

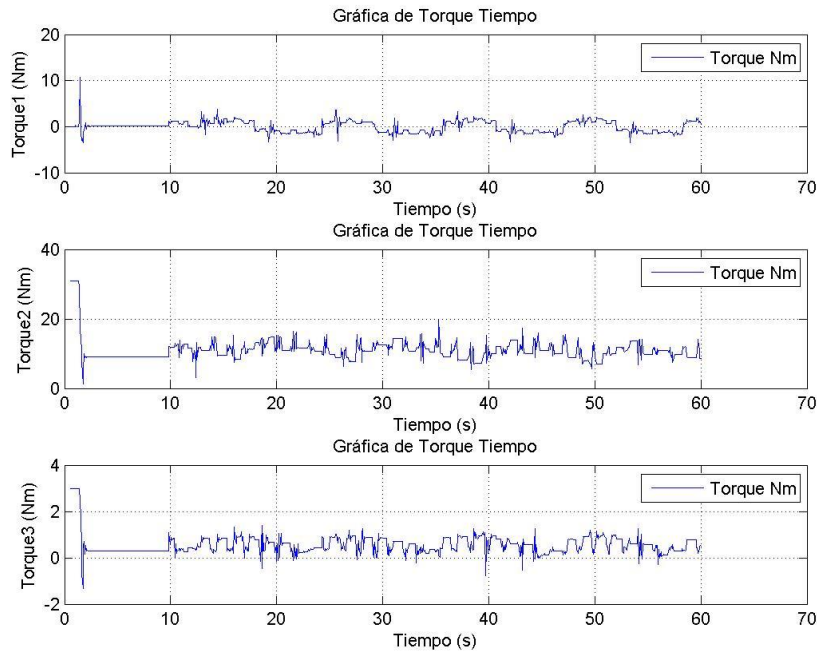


Figura 5.24: Gráfica de torque con respecto al tiempo para el controlador TANH-TANH de impedancia.

Como se pudo observar, el controlador PD fue el que mejor trazó el rectángulo deseado, manteniendo una fuerza de contacto menor a 4 N, a diferencia de las estructuras atan y tanh que no pudieron mantener correctamente la trayectoria. Esto se debe principalmente al hecho de que en el movimiento en el espacio de trabajo cartesiano solo era posible pensar la fuerza aplicada en el eje X, lo cual claramente afectó el desempeño de los controladores. En la tabla 5.10 se muestran los valores de las ganancias k_p y k_v utilizadas para los controladores de impedancia.

TABLA 5.10 TABLA DE VALORES k_p Y k_v PARA EL CONTROL DE IMPEDANCIA

Variable	Valor
k_{p1}	100
k_{v1}	50
k_{p2}	60
k_{v2}	20
k_{p3}	15
k_{v3}	3

CONCLUSIONES

Como se mostró en este trabajo de tesis, se cumplieron con los objetivos propuestos, principalmente con la implementación algoritmos de control de impedancia en un robot de transmisión directa de 3 grados de libertad; estos son el PD, el TANH-TANH y el ATAN-ATAN. En donde el hardware diseñado para el ROTRADI VI resulto ser eficaz y funcional para propósitos de la tesis.

Se revisó la bibliografía y se describieron diferentes sistemas dinámicos de robots manipuladores, desde un robot de 1 grado de libertad, hasta el modelo de 3 grados de libertad a través de la metodología de Euler-Lagrange de donde se obtuvieron los parámetros necesarios para obtener el modelo dinámico de cada robot y aplicar estos parámetros a cada estrategia de control en donde se requirieron dichos parámetros: control articular, cartesiano, de fuerza y de impedancia.

El estudio de la estrategia de control de impedancia y el sistema dinámico del ROTRADI VI, permitió la implementación de los diferentes esquemas y algoritmos de control propuestos en esta tesis.

Tras el estudio del sistema, el diseño y construcción del hardware y software fue eficaz, simple de implementar, pero sobre todo escalable, lo que significa que es posible copiar cualquier número de tarjetas electrónicas y aplicarlo a un robot de n grados de libertad.

El estudio del funcionamiento de los *encoders* en el modo diferencial de señales A-B (*encoder* de cuadratura) permitió obtener un algoritmo de lectura de posición altamente eficaz, que permite una posible aplicación general de cualquier *encoder* de cuadratura, no solo para la lectura de una tarjeta Arduino, si no para cualquier microcontrolador en general.

El Arduino Mega es suficiente para poder generar un programa de control completo y en muchos sentidos eficaz para la implementación de diversas estrategias de control, no obstante hay que remarcar que las limitaciones tecnológicas en cuanto a velocidad de procesamiento y cálculo no permite utilizar al ROTRADI a su máxima capacidad en cuanto respecta a resolución, pues todos sus actuadores tuvieron que ser escalados de 4,000,000 de

pulsos por vuelta a tan solo 10,000, es decir maneja una resolución de 0.036 grados por pulso que suficiente para el trabajo realizado.

El diseño del programa en Arduino permitió la implementación de múltiples estrategias de control modificando un único bloque correspondiente a la ecuación de control experimental. Los algoritmos de control implementados permitieron el diseño de un programa con la capacidad de mimetizar el movimiento de un fisioterapeuta el cual podría ser utilizado para propósitos de rehabilitación para algún paciente.

Finalmente los resultados de esta tesis sirven como base para la actualización de no solo el ROTRADI VI, si no de todos los robots de transmisión directa ubicados en el laboratorio, pues se ha demostrado el hecho de que los ROTRADIs ya no necesitan estar confinados al control exclusivo de una única pc y que a través de la interfaz diseñada, es posible usar una computadora portátil para la implementación experimental de cualquier estrategia de control que en su momento se desee conocer su comportamiento a través de un ROTRADI.

APÉNDICE A: MODELO DINÁMICO DE UN ROBOT ANTROPOMÓRFICO DE 3 GRADOS DE LIBERTAD

En esta sección se presenta la deducción del modelo dinámico del robot antropomorfo de tres grados de libertad, esto usando el método de Lagrange tomando en cuenta el ancho de los motores.

Parámetros del sistema robot antropomorfo de 3 GDL:

m_1 : Masa del eslabón 1.

m_2 : Masa del eslabón 2.

m_3 : Masa del eslabón 3.

l_1 : Longitud del primer eslabón.

l_2 : Longitud del segundo eslabón.

l_3 : Longitud del tercer eslabón.

l_{1c} : Longitud del centro de masa del primer eslabón.

l_{c2} : Longitud del centro de masa del segundo eslabón.

l_{c3} : Longitud del centro de masa del tercer eslabón.

I_1 : Inercia del primer eslabón.

I_2 : Inercia del segundo eslabón.

I_3 : Inercia del tercer eslabón.

g : Aceleración de la gravedad.

b_1 : Coeficiente de fricción viscosa.

b_2 : Coeficiente de fricción viscosa.

b_3 : Coeficiente de fricción viscosa.

Cinemática directa

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ l_1 \end{bmatrix},$$

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} w_1 \text{sen}(q_1) + l_{c2} \text{sen}(q_2) \cos(q_1) \\ w_1 \cos(q_1) + l_{c2} \text{sen}(q_2) \text{sen}(q_1) \\ l_1 - l_{c2} \text{sen}(q_2) \end{bmatrix},$$

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} (w_1 + w_2) \text{sen}(q_1) + \cos(q_1) (l_2 \text{sen}(q_2) + l_{c3} \text{sen}(q_2 + q_3)) \\ (w_1 + w_2) \cos(q_1) + \text{sen}(q_1) (l_2 \text{sen}(q_2) + l_{c3} \text{sen}(q_2 + q_3)) \\ l_1 - l_2 \text{sen}(q_2) - l_{c3} \text{sen}(q_2 + q_3) \end{bmatrix}.$$

Cinemática diferencial

$$\mathbf{v}_1 = \frac{d}{dt} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = 0,$$

$$\mathbf{v}_2 = \frac{d}{dt} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} w_1 \cos(q_1) \dot{q}_1 - l_{c2} \text{sen}(q_1) \text{sen}(q_2) \dot{q}_1 + l_{c2} \cos(q_2) \cos(q_1) \dot{q}_2 \\ w_1 \text{sen}(q_1) \dot{q}_1 + l_{c2} \cos(q_1) \text{sen}(q_2) \dot{q}_1 + l_{c2} \cos(q_2) \text{sen}(q_1) \dot{q}_2 \\ -l_{c2} \cos(q_2) \dot{q}_2 \end{bmatrix},$$

$$\mathbf{v}_3 = \frac{d}{dt} \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix},$$

$$\begin{aligned} \frac{dx_3}{dt} &= (w_1 + w_2) \cos(q_1) \dot{q}_1 + l_2 [\cos(q_1) \cos(q_2) \dot{q}_2 - \text{sen}(q_1) \text{sen}(q_2) \dot{q}_1] \\ &\quad + l_{c3} [\cos(q_1) \cos(q_2 + q_3) (\dot{q}_2 + \dot{q}_3) - \text{sen}(q_1) \text{sen}(q_2 + q_3) \dot{q}_1], \end{aligned}$$

$$\begin{aligned} \frac{dy_3}{dt} &= (w_1 + w_2) \text{sen}(q_1) \dot{q}_1 + l_2 [\text{sen}(q_1) \cos(q_2) \dot{q}_2 + \cos(q_1) \text{sen}(q_2) \dot{q}_1] \\ &\quad + l_{c3} [\text{sen}(q_1) \cos(q_2 + q_3) (\dot{q}_2 + \dot{q}_3) + \cos(q_1) \text{sen}(q_2 + q_3) \dot{q}_1], \end{aligned}$$

$$\frac{dz_3}{dt} = -l_2 \cos(q_2) \dot{q}_2 + l_{c3} \cos(q_2 + q_3) (\dot{q}_2 + \dot{q}_3),$$

Finalmente,

$$\mathbf{v}_2^T \mathbf{v}_2 = w_1^2 \dot{q}_1^2 + w_1 \dot{q}_1 l_{c2} (\text{sen}(q_2) \dot{q}_1 + 2 \cos(q_2) \dot{q}_2) + 2l_{c2}^2 \cos^2(q_2) \dot{q}_2^2 + l_{c2}^2 \text{sen}^2(q_2) \dot{q}_1^2,$$

$$\begin{aligned} \mathbf{v}_3^T \mathbf{v}_3 &= W^2 \dot{q}_1^2 + 2W \dot{q}_1 l_{c3} \cos(q_2 + q_3) (\dot{q}_2 + \dot{q}_3) + 2W \dot{q}_1 l_2 \cos(q_2) \dot{q}_2 + \text{sen}^2(q_2) \dot{q}_1^2 \\ &\quad + l_2^2 \cos^2(q_2) \dot{q}_2^2 + l_{c3}^2 \cos^2(q_2 + q_3) (\dot{q}_2 + \dot{q}_3)^2 \\ &\quad + 2l_2 \text{sen}(q_2) \dot{q}_1^2 l_{c3} \text{sen}(q_2 + q_3). \end{aligned}$$

Modelo de energía

$$K_1(q, \dot{q}) = \frac{1}{2} I_1 \dot{q}_1^2$$

$$\begin{aligned} K_2(q, \dot{q}) &= \frac{1}{2} I_2 (\dot{q}_1 + \dot{q}_2)^2 + \frac{1}{2} m_2 ((w_1^2 + \text{sen}(q_2) l_{c2} (-2w_1 \text{sen}(2q_1) + \text{sen}(q_2) l_{c2})) \dot{q}_1^2 \\ &\quad + 2w_1 \cos(2q_1) \cos(q_2) l_{c2} \dot{q}_1 \dot{q}_2 + l_{c2}^2 \dot{q}_2^2) \end{aligned}$$

$$K_3(q, \dot{q}) = \frac{1}{2} I_3 (\dot{q}_1 + \dot{q}_2 + \dot{q}_3)^2 + \frac{1}{2} m_3 ((w_1^2 + w_2^2 + (\sin(\dot{q}_2)l_2 + \sin(q_2 + q_3)l_{c3})(-2(w_1 + w_2^2)\sin(2q_1) + \sin(q_2)l_2 + \sin(q_2 + q_3)l_{c3}))\dot{q}_1^2 + l_2^2 \dot{q}_2^2 + 2\cos(q_3)l_2 l_{c3} \dot{q}_2 (\dot{q}_2 + \dot{q}_3) + l_{c3}^2 (\dot{q}_2 + \dot{q}_3)^2 + 2(w_1 + w_2)\cos(2q_1)\dot{q}_1 (\cos(q_2)l_2 \dot{q}_2 + \cos(q_2 + q_3)l_{c3}(\dot{q}_2 + \dot{q}_3)))$$

$$U_1(q) = m_1 g l_{c1}$$

$$U_2(q) = m_2 g [l_1 - l_{c2} \cos(q_2)]$$

$$U_3(q) = m_3 g [l_1 - l_2 \cos(q_2) - l_{c3} \cos(q_2 + q_3)]$$

$$U(q) = m_1 g l_{c1} + m_2 g [l_1 - l_{c2} \cos(q_2)] + m_3 g [l_1 - l_2 \cos(q_2) - l_{c3} \cos(q_2 + q_3)]$$

$$\begin{aligned} \mathcal{L} = & -g l_1 l_{c1} m_1 - g l_{c2} (l_1 - \cos(q_2) l_{c2}) m_2 - g l_{c3} (l_1 - \cos(q_2) l_2 - \cos(q_2 + q_3) l_{c3}) m_3 \\ & + \frac{1}{2} I_1 \dot{q}_1^2 + \frac{1}{2} I_2 (\dot{q}_1 + \dot{q}_2)^2 + \frac{1}{2} m_2 ((\sin(q_2)^2 l_{c2}^2 - 2\sin(2q_1)\sin(q_2)l_{c2}w_1 + w_1^2)\dot{q}_1^2 + 2\cos(2q_1)\cos(q_2)l_{c2}w_1\dot{q}_1\dot{q}_2 + l_{c2}^2 \dot{q}_2^2) + \frac{1}{2} I_3 (\dot{q}_1 + \dot{q}_2 + \dot{q}_3)^2 \\ & + \frac{1}{2} m_3 ((w_1 + w_2)^2 \dot{q}_1^2 + l_2^2 (\sin(q_2)^2 \dot{q}_1^2 + \dot{q}_1^2) + l_{c3} (w_1 + w_2)\dot{q}_1 (-2\sin(2q_1)\sin(q_2 + q_3)\dot{q}_1 + 2\cos(2q_1)\cos(q_2 + q_3)(\dot{q}_2 + \dot{q}_3)) \\ & + l_{c3}^2 (\sin(q_2 + q_3)^2 \dot{q}_1^2 + (\dot{q}_2 + \dot{q}_3)^2) + 2l_2 (-w_1 + w_2)\dot{q}_1 (\sin(2q_1)\sin(q_2)\dot{q}_1 - \cos(2q_1)\cos(q_2)\dot{q}_2) + l_{c3} (\sin(q_2)\sin(q_2 + q_3)\dot{q}_1^2 + \cos(q_3)\dot{q}_2(\dot{q}_2 + \dot{q}_3))) \end{aligned}$$

Ecuaciones de movimiento

$$\begin{aligned}
\tau_1 = & -\frac{1}{2}m_2(-4\cos(2q_1)\text{sen}(q_2)l_{c2}w_1\dot{q}_1^2 - 4\cos(q_2)\text{sen}(2q_1)l_{c2}w_1\dot{q}_1\dot{q}_2) \\
& -\frac{1}{2}m_3(-2l_2(w_1 + w_2)\dot{q}_1(2\cos(2q_1)\text{sen}(q_2)\dot{q}_1 + 2\cos(q_2)\text{sen}(2q_1)\dot{q}_2) \\
& + l_{c3}(w_1 + w_2)\dot{q}_1(-4\cos(2q_1)\text{sen}(q_2 + q_3)\dot{q}_1 - 4\cos(q_2 + q_3)\text{sen}(2q_1)(\dot{q}_2 \\
& + \dot{q}_3))) + I_1\ddot{q}_1 + I_2(\ddot{q}_1 + \ddot{q}_2) + \frac{1}{2}m_2(-4\cos(q_2)\text{sen}(2q_1)l_{c2}w_1\dot{q}_1\dot{q}_2 \\
& - 2\cos(2q_1)\text{sen}(q_2)l_{c2}w_1\dot{q}_2^2 + 2\dot{q}_1(-4\cos(2q_1)\text{sen}(q_2)l_{c2}w_1\dot{q}_1 \\
& + 2\cos(q_2)\text{sen}(q_2)l_{c2}^2\dot{q}_2 - 2\cos(q_2)\text{sen}(2q_1)l_{c2}w_1\dot{q}_2) \\
& + 2(\text{sen}(q_2)^2l_{c2}^2 - 2\text{sen}(2q_1)\text{sen}(q_2)l_{c2}w_1 + w_1^2)\dot{q}_1 \\
& + 2\cos(2q_1)\cos(q_2)l_{c2}w_1\ddot{q}_2) + I_3(\ddot{q}_1 + \ddot{q}_2 + \ddot{q}_3) \\
& + \frac{1}{2}m_3(-4\cos(2q_1)\text{sen}(q_2) + q_3)l_{c3}(w_1 + w_2)\dot{q}_1^2 + 4\cos(q_2)\text{sen}(q_2)l_2^2\dot{q}_1\dot{q}_2 \\
& + 4\cos(q_2 + q_3)\text{sen}(q_2 + q_3)l_{c3}^2\dot{q}_1(\dot{q}_2 + \dot{q}_3) \\
& - 2\cos(q_2 + q_3)\text{sen}(2q_1)l_{c3}(w_1 + w_2)\dot{q}_1(\dot{q}_3 + \dot{q}_3) + 2\text{sen}(q_2)^2l_2^2\dot{q}_1 \\
& + 2\text{sen}(q_2 + q_3)^2l_{c3}^2\dot{q}_1 - 2\text{sen}(2q_1)\text{sen}(q_2 + q_3)l_{c3}(w_1 + w_2)\dot{q}_1 \\
& + 2(w_1 + w_2)^2\dot{q}_1 + 2l_2(-2\cos(2q_1)\text{sen}(q_2)(w_1 + w_2)\dot{q}_1^2 + 2\cos(q_2)\text{sen}(q_2 \\
& + q_3)l_{c3}\dot{q}_1\dot{q}_2 - \cos(q_2)\text{sen}(2q_1[t])(w_1 + w_2)\dot{q}_1\dot{q}_2 \\
& + 2\cos(q_2 + q_3)\text{sen}(q_2)l_{c3}\dot{q}_1(\dot{q}_2 + \dot{q}_3) + 2\text{sen}(q_2)\text{sen}(q_2 + q_3)l_{c3}\ddot{q}_1) \\
& - \text{sen}(2q_1)\text{sen}(q_2)(w_1 + w_2)\ddot{q}_1 - (w_1 + w_2)(2\cos(2q_1)\text{sen}(q_2)\dot{q}_1^2 \\
& + 3\cos(q_2)\text{sen}(2q_1)\dot{q}_1\dot{q}_2 + \cos(2q_1)\text{sen}(q_2)\dot{q}_2^2 + \text{sen}(2q_1)\text{sen}(q_2)\ddot{q}_1 \\
& - \cos(2q_1)\cos(q_2)\ddot{q}_2)) + l_{c3}(w_1 + w_2)(-4\cos(2q_1)\text{sen}(q_2 + q_3)\dot{q}_1^2 \\
& - 6\cos(q_2 + q_3)\text{sen}(2q_1)\dot{q}_1(\dot{q}_2 + \dot{q}_3) - 2\cos(2q_1)\text{sen}(q_2 + q_3)(\dot{q}_2 + \dot{q}_3)^2 \\
& - 2\text{sen}(2q_1)\text{sen}(q_2 + q_3)\ddot{q}_1 + 2\cos(2q_1)\cos(q_2 + q_3)(\ddot{q}_2 + \ddot{q}_3))) \\
& + \text{Sign}(\dot{q}_1)f_{c1} + (1 - \text{Abs}[\text{Sign}[\dot{q}_1]])f_{e1} + b_1\dot{q}_1
\end{aligned}$$

$$\begin{aligned}
\tau_2 = & g\text{sen}(q_2)l_{c2}^2m_2 + gl_{c3}(\text{sen}(q_2)l_2 + \text{sen}(q_2 + q_3)l_{c3})m_3 - \frac{1}{2}m_2((2\cos(q_2)\text{sen}(q_2)l_{c2}^2 \\
& - 2\cos(q_2)\text{sen}(2q_1)l_{c2}w_1)\dot{q}_1^2 - 2\cos(2q_1)\text{sen}(q_2)l_{c2}w_1\dot{q}_1\dot{q}_2) \\
& - \frac{1}{2}m_3(2\cos(q_2)\text{sen}(q_2)l_2^2\dot{q}_1^2 + 2\cos(q_2 + q_3)\text{sen}(q_2 + q_3)l_{c3}^2\dot{q}_1^2 \\
& + 2l_2(l_{c3}(\cos(q_2 + q_3)\text{sen}(q_2)\dot{q}_1^2 + \cos(q_2)\text{sen}(q_2 + q_3)\dot{q}_1^2) - (w_1 \\
& + w_2)\dot{q}_1(\cos(q_2)\text{sen}(2q_1)\dot{q}_1 + \cos(2q_1)\text{sen}(q_2)\dot{q}_2)) + l_{c3}(w_1 \\
& + w_2)\dot{q}_1(-2\cos(q_2 + q_3)\text{sen}(2q_1)\dot{q}_1 - 2\cos(2q_1)\text{sen}(q_2 + q_3)(\dot{q}_2 + \dot{q}_3))) \\
& + I_3(\ddot{q}_1 + \ddot{q}_2) + \frac{1}{2}m_2(-4\cos(q_2)\text{sen}(2q_1)l_{c2}w_1\dot{q}_1^2 \\
& - 2\cos(2q_1)\text{sen}(q_2)l_{c2}w_1\dot{q}_1\dot{q}_2 + 2\cos(2q_1)\cos(q_2)l_{c2}w_1\dot{q}_1 + 2l_{c2}^2\ddot{q}_2) + I_3(\ddot{q}_1 \\
& + \ddot{q}_2 + \ddot{q}_3) + \frac{1}{2}m_3(-4\cos(q_2 + q_3)\text{sen}(2q_1)l_{c3}(w_1 + w_2)\dot{q}_1^2 \\
& - 2\cos(2q_1)\text{sen}(q_2 + q_3)l_{c3}(w_1 + w_2)\dot{q}_1(\dot{q}_2 + \dot{q}_3) \\
& + 2\cos(2q_1)\cos(q_2 + q_3)l_{c3}(w_1 + w_2)\dot{q}_1 + 2l_2^2\ddot{q}_2 + 2l_{c3}^2(\ddot{q}_2 + \ddot{q}_3) \\
& + 2l_2(-2\cos(q_2)\text{sen}(2q_1)(w_1 + w_2)\dot{q}_1^2 - \cos(2q_1)\text{sen}(q_2)(w_1 + w_2)\dot{q}_1\dot{q}_2 \\
& + \cos(2q_1)\cos(q_2)(w_1 + w_2)\dot{q}_1 + l_{c3}(-\text{sen}(q_3)\dot{q}_2\dot{q}_3 - \text{sen}(q_3)\dot{q}_3(\dot{q}_2 + \dot{q}_3) \\
& + \cos(q_3)\ddot{q}_2 + \cos(q_3)(\ddot{q}_2 + \ddot{q}_3)))) + \text{Sign}(\dot{q}_2)f_{c2} + (1 - \text{Abs}[\text{Sign}[\dot{q}_2]])f_{e2} \\
& + b_2\dot{q}_2
\end{aligned}$$

$$\begin{aligned}
\tau_3 = & g\text{sen}(q_2 + q_3)l_{c3}^2m_3 - \frac{1}{2}m_3(2\cos(q_2 + q_3)\text{sen}(q_2 + q_3)l_{c3}^2\dot{q}_1^2 + l_{c3}(w_1 \\
& + w_2)\dot{q}_1(-2\cos(q_2 + q_3)\text{sen}(2q_1)\dot{q}_1 - 2\cos(2q_1)\text{sen}(q_2 + q_3)(\dot{q}_2 + \dot{q}_3)) \\
& + 2l_2l_{c3}(\cos(q_2 + q_3)\text{sen}(q_2)\dot{q}_1^2 - \text{sen}(q_3)\dot{q}_2(\dot{q}_2 + \dot{q}_3))) + I_3(\ddot{q}_1 + \ddot{q}_2 + \ddot{q}_3) \\
& + \frac{1}{2}m_3(-4\cos(q_2 + q_3)\text{sen}(2q_1)l_{c3}(w_1 + w_2)\dot{q}_1^2 - 2\text{sen}(q_3)l_2l_{c3}\dot{q}_2\dot{q}_3 \\
& - 2\cos(2q_1)\text{sen}(q_2 + q_3)l_{c3}(w_1 + w_2)\dot{q}_1(\dot{q}_2 + \dot{q}_3) \\
& + 2\cos(2q_1)\cos(q_2 + q_3)l_{c3}(w_1 + w_2)\dot{q}_1 + 2\cos(q_3)l_2l_{c3}\ddot{q}_2 + 2l_{c3}^2(\ddot{q}_2 + \ddot{q}_3)) \\
& + \text{Sign}(\dot{q}_3)f_{c3} + (1 - \text{Abs}[\text{Sign}[\dot{q}_3]])f_{e3} + b_3\dot{q}_3
\end{aligned}$$

APÉNDICE B: CONFIGURACIÓN DE DRIVERS

B.1. PUESTA EN OPERACIÓN DE SERVO DRIVERS

El ROTRADI es un brazo robot experimental de una composición mucho más refinada que la mayoría de los brazos robots convencionales, cada motor debe ser controlado por Drivers especializados los cuales se deben configurarse apropiadamente para poder controlar el motor según la variables deseada (Torque, Velocidad, Posición entre otros).

B.1.1. PREPARACIÓN DE SERVO DRIVERS

Pasando a la parte de puesta en marcha del ROTRADI, se debe conocer la forma en la que funcionan no sólo los motores, si no los manejadores de estos (Driver). Para cada motor de transmisión directa tenemos un Servo Driver similares al mostrado en la Figura 2.1, cada uno de estos tiene diferentes características dependiendo del motor a utilizar.

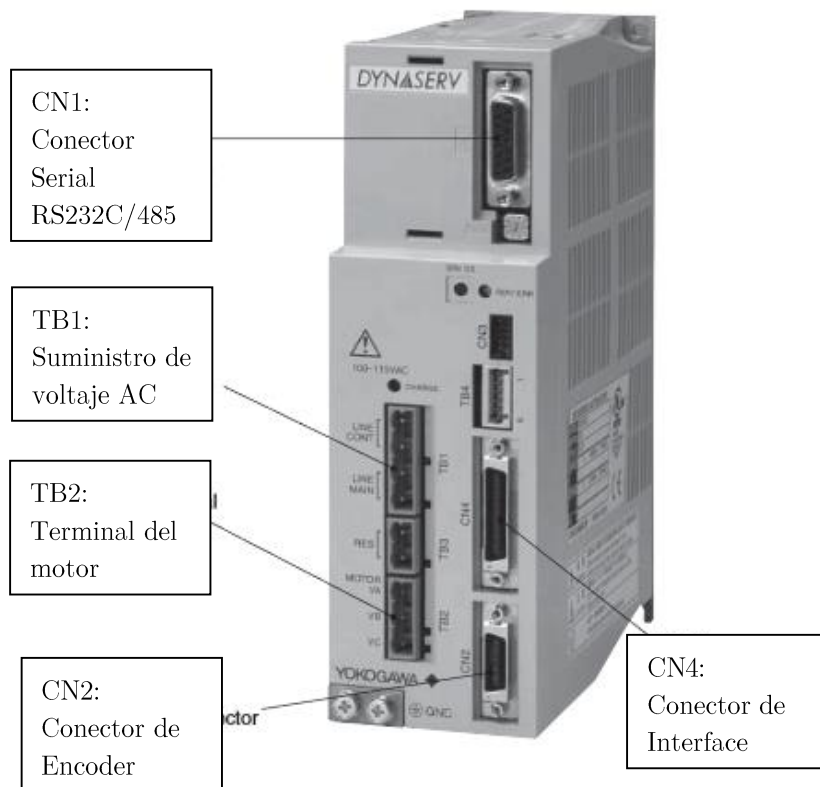


Figura 2.1: Ejemplo de Servo Driver ocupados por un motor de transmisión directa [24].

Lo primero es verificar el tipo de motor y sus requerimientos para su puesta en operación, para esto se debe verificar el número de serie impreso en cada motor, por lo general marcado en una etiqueta rectangular en el motor (Figura 2.2).

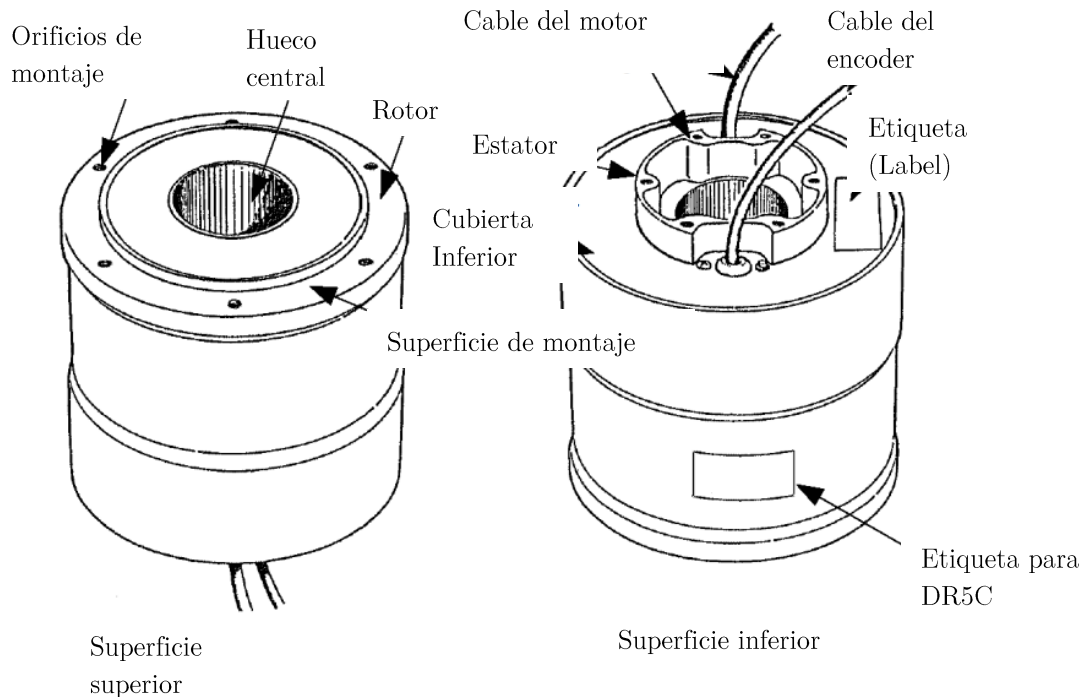


Figura 2.2: Esquema general de los motores de transmisión directa, donde lo que se debe notar es la presencia de la etiqueta donde se encuentra el número de serie (Label) [24].

Según sea el número de serie del motor, se va requerir alguna especificación mínima para el Driver, de tal manera que sea capaz de controlar al motor sin que tenga fallas por falta de carga eléctrica, o de frecuencia de trabajo. Para el caso específico del robot de pruebas actual (Péndulo) es el DMG3-1015B, tenemos que sus características principales son:

Torque Máximo - 15Nm.

Velocidad nominal - 2 PRM.

Tipo de encoder – Incremental / Pulso.

Frecuencia máxima del encoder – 3 MHz.

Tomando estas características, se utilizó el Driver G3 Dynaserv, fabricado por la compañía Parker Automation. Este Driver cuenta con una lista de servomotores compatibles, entre los cuales se encuentra el motor utilizado.

Para este driver existen varios métodos para programarlo, ya sea vía PC u “*Operation Display Pendant*”, para nuestro caso se programara vía PC a través del conector CN1, cuyo mapeo de pines se muestra en la Tabla 2.1.

Pin No.	Nombre de la Señal	Definición
1	FG	Terminal de Tierra
2	RxD	Terminal RxD comunicación unilateral
3	TxD	Terminal TXD comunicación unilateral
4	A	Rx (+) Terminal multicomunicación
5	Y	Tx (+) Terminal multicomunicación
6	485SW	Condición "busy"
7	TRMP	Terminador de comunicación
8	CN1SW	Condición "busy"
9	+5V	Voltaje positivo
10	SG/LG	Señal de tierra
11	B	Rx (-) Terminal multicomunicación
12	Z	Tx (-) Terminal multicomunicación
13	SG/LG	Señal de tierra
14	TRHN	Terminador de comunicación
15	SG/LG	Señal de tierra

Tabla 2.1: Tabla con el mapeo de pines del conector CN1.

Ahora bien la conexión hacia la computadora se realiza mediante una conexión RS232 como se muestra en la Figura 2.3.

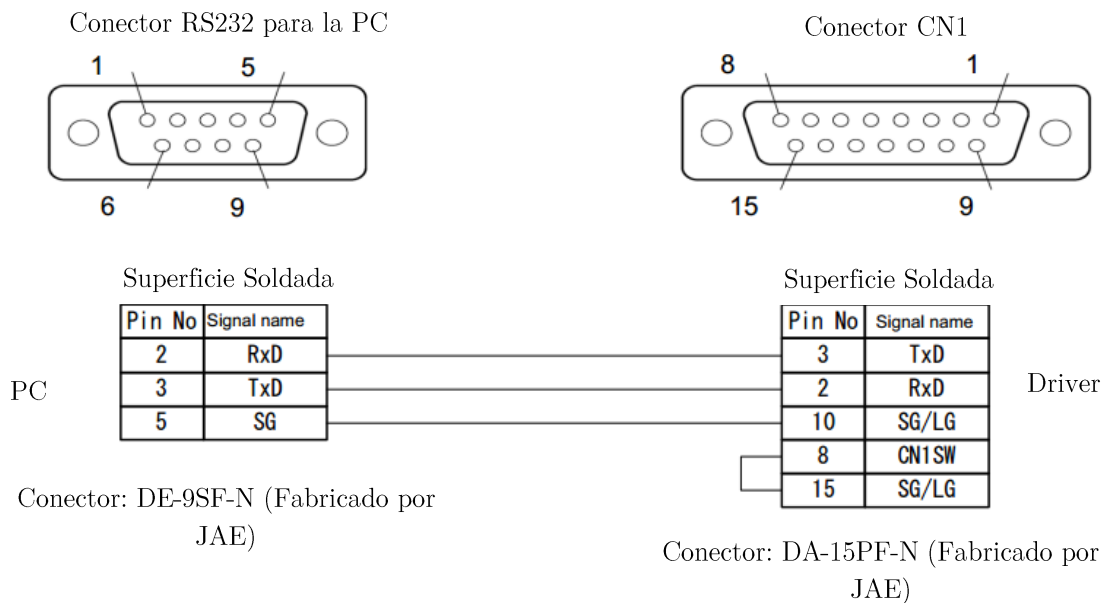


Figura 2.3: Conexiones PC-CN1.

Pin No.	Nombre de la señal	Color	Marca	Definición	Comentario
1	COMP1	BLANCO	AZUL-NEGRO 3	Voltaje +	Entrada de voltaje
2	DO_0	NARANJA	AZUL-NEGRO 1	Voltaje -	
3	DO_1	NARANJA	ROJO 1	I/O	Salidas de funciones
4	DO_2	GRIS	ROJO 1	I/O	
5	DO_3	BLANCO	ROJO 1	I/O	
6	DO_3	AMARILLO	ROJO 1	I/O	
7	DO_4	ROSA	ROJO 1	I/O	
8	DO_5	NARANJA	ROJO 2	I/O	
9	UA_OUT+	GRIS	ROJO 2	Pulso de posición +	
10	UA_OUT-	GRIS	AZUL 2	Pulso de posición -	
11	DB_OUT+	BLANCO	ROJO 2	Pulso de posición +	Seleccionar tipo de pulso "UP-DOWN" o "A-B"
12	DB_OUT-	BLANCO	AZUL 2	Pulso de posición -	
13	Z_OUT+	AMARILLO	ROJO 2	Pulso Z+	Salidas de pulso z
14	Z_OUT-	AMARILLO	AZUL O NEGRO 2	Pulso Z-	
15	PUA_IN+	ROSA	ROJO 2	Comando de pulso +	Seleccionar tipo de pulso "PLS-Sing" o "A-B"
16	PUA_IN-	ROSA	AZUL 2	Comando de pulso -	
17	SDB_IN+	NARANJA	RED 3	Comando de pulso +	
18	SDB_IN-	NARANJA	BLU3 3	Comando de pulso -	
19	DI_0	GRIS	ROJO 3	I/O	Entradas de funciones
20	DI_1	BLANCO	ROJO 3	I/O	
21	DI_2	AMARILLO	ROJO 3	I/O	
22	DI_3	ROSA	ROJO 3	I/O	
23	DI_4	NARANJA	ROJO 4	I/O	
24	DI_5	GRIS	ROJO 4	I/O	
25	DI_6	BLANCO	ROJO 4	I/O	
26	DI_7	AMARILLO	ROJO 4	I/O	
27	DI_8	ROSA	ROJO 4	I/O	
28	DI_9	NARANJA	LINEA CONTINUA	I/O	
29	DI_10	GRIS	LINEA CONTINUA	I/O	
30	DI_11	BLANCO	LINEA CONTINUA	I/O	
33	ASUB_IN+	AMARILLO	LINEA CONTINUA	Entrada Analógica +	Entrada de ejecución de fuerza/torque limite
34	ASUB_IN-	AMARILLO	LINEA CONTINUA	Entrada Analógica -	
35	ACMD_IN+	ROSA	LINEA CONTINUA	Entrada Analógica comando	Entrada de velocidad o torque/fuerza
36	ACMD_IN-	ROSA	LINEA CONTINUA	Entrada Analógica comando	

Tabla 2.2: Tabla de pines del cable CN4.

El control del motor sin la necesidad de una computadora es posible con el uso del cable de control CN4. Nótese que según se requiera hay diferentes conexiones según la variable a

controlar, ya sea en modo torque, velocidad o posición, se utilizaran diferentes pines, por lo que se recomienda soldar únicamente los pines correspondientes al uso del sistema y dejar los demás sin soltar para evitar posibles confusiones de configuración, a continuación se muestra un mapeo de pines del cable de control CN4. En la Figura 2.4 se muestran las terminales de entrada y salida del modelo de driver GIII.

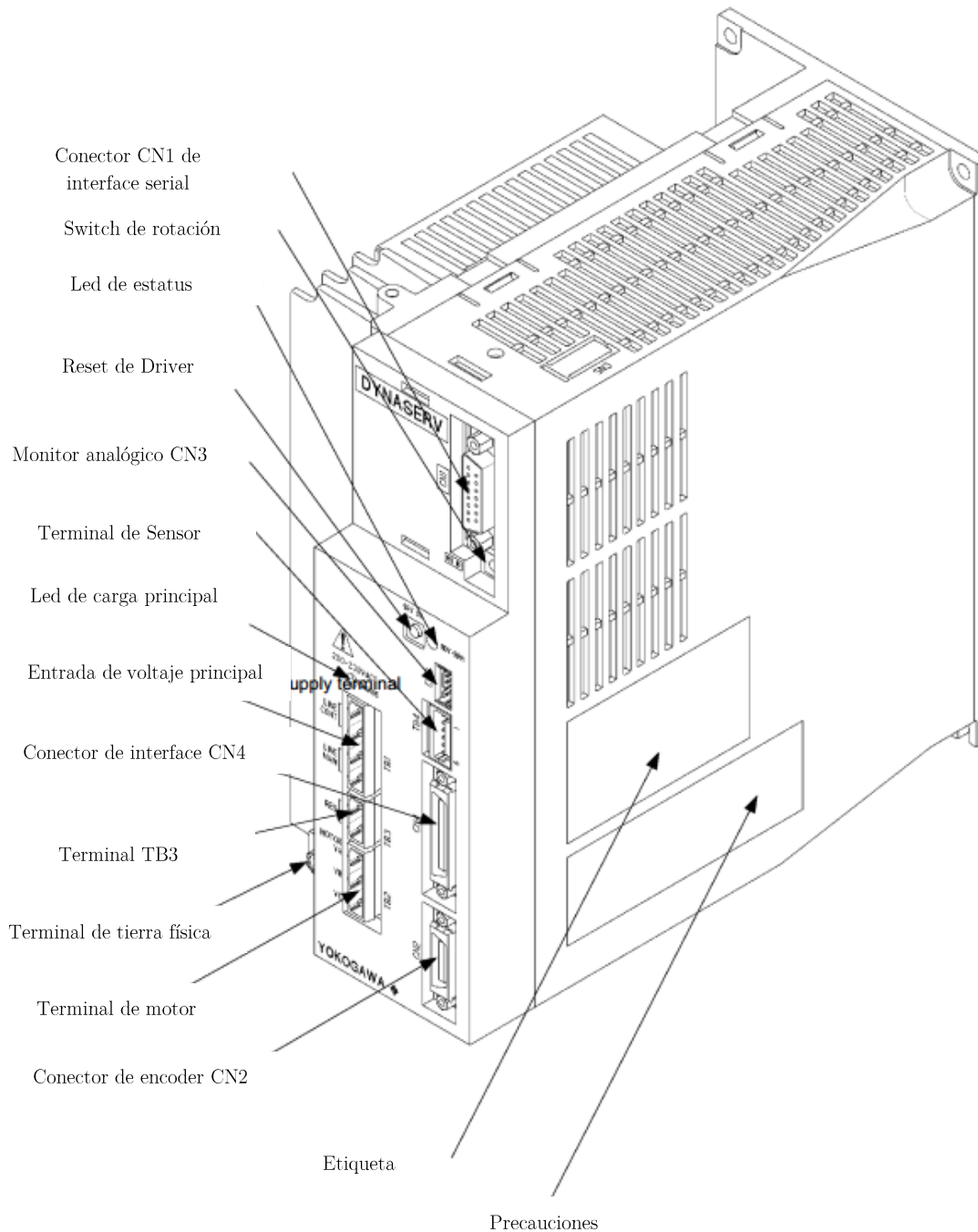


Figura 2.4: Vista general del Servo Driver en donde se muestran todas las terminales de entrada y salida [24].

Procediendo a las pruebas de funcionamiento del motor, se recomienda la configuración mostrada a continuación en la Figura 2.5, en donde la computadora va a realizar la programación necesaria en base a comandos “Demo” para conocer el funcionamiento general tanto del Driver como del motor.

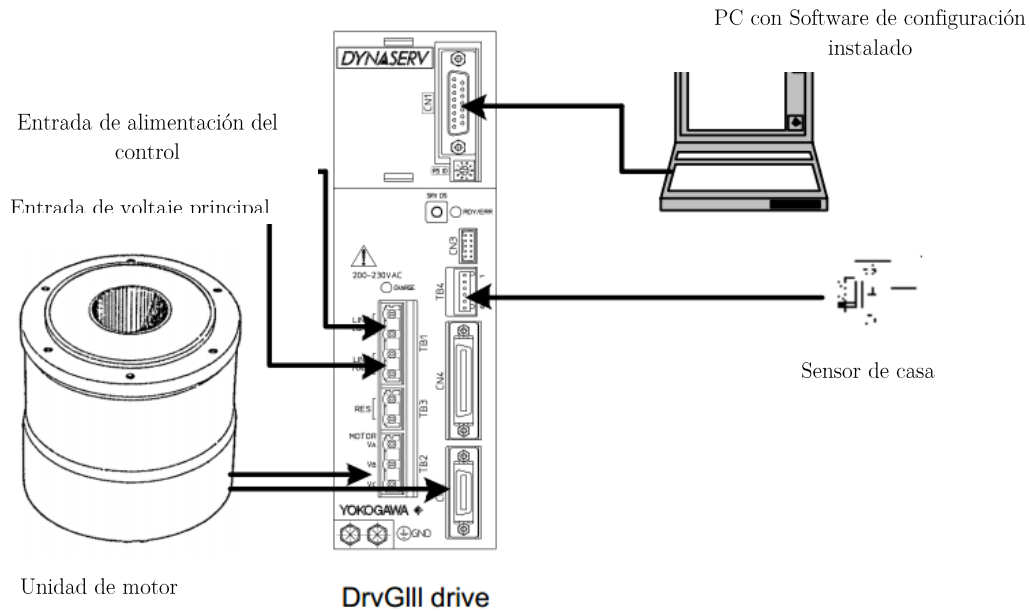


Figura 2.5: Configuración recomendada para propósitos de pruebas [24].

Una vez que se activa el driver, si las conexiones son correctas, el indicador que se encuentra al reverso deberá ponerse verde, en caso contrario se debe revisar las conexiones realizadas¹.

¹ Nota personal: Al desconectar la alimentación del Driver seguirá encendida una luz roja de “Power”, bajo ninguna circunstancia se deben tocar los pines de alimentación, pues la carga eléctrica residual puede generar descargas sobre el usuario.



Figura 2.6: Indicador de no errores del servo [24].

Una vez que el Driver no muestre error alguno, es posible proceder a las pruebas de funcionamiento, para esto requerimos el uso de un software especialmente diseñado para estos drivers los cuales son proporcionados por el fabricante.

B.1.2. CONFIGURACIÓN PRELIMINAR DEL DRIVER

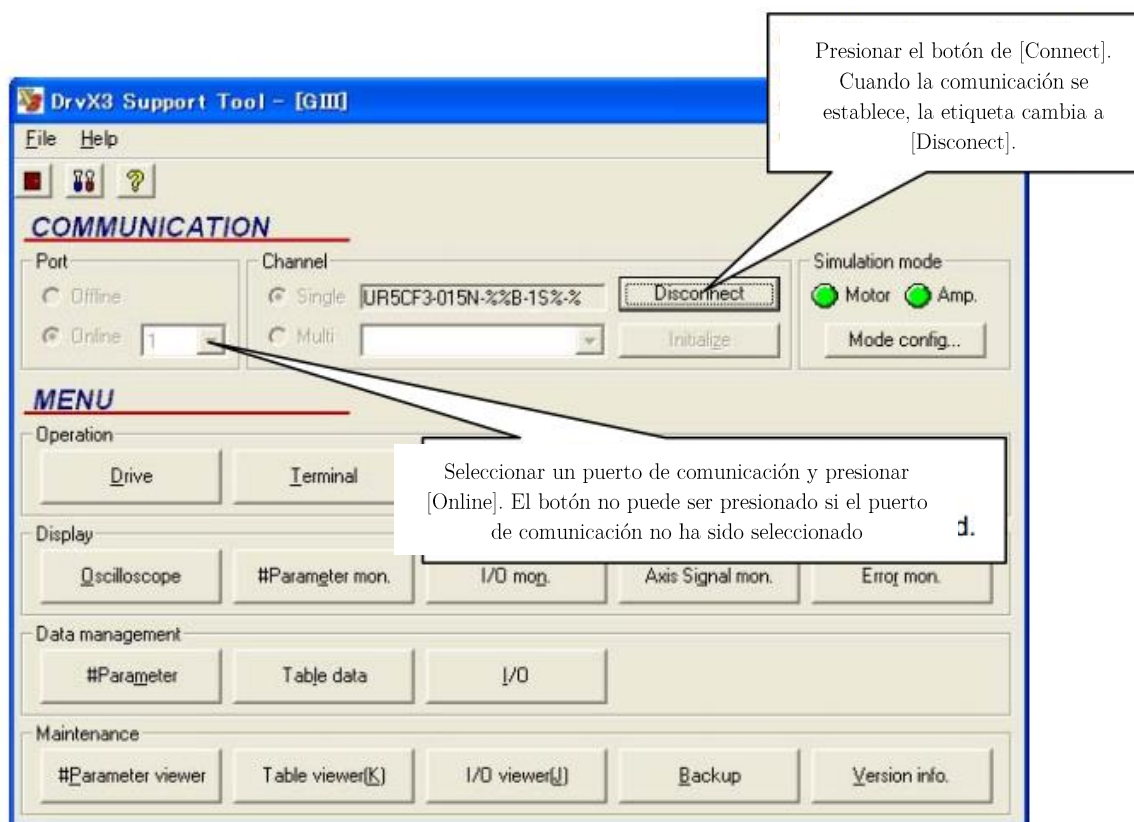


Figura 2.7: Vista general del software para programar el Driver [24].

Lo primero que se requiere es realizar la conexión con el servo con el que se quiere trabajar, basta con dar clic en el botón de “*Connect*” una vez hecho esto, el siguiente paso consiste en dar clic en el botón de “*Drive*”.

Esto nos abrirá dos ventanas, de las cuales una nos preguntara si deseamos conceder el control a puerto serial. Una vez hecho esto, se requiere presionar la pestaña de “*Auto tuning Operation*” y después dar clic en el botón “*Servo-On*”, el cual activará el motor, finalmente se requiere dar clic en el botón “*Drive*”. Cuando se active “*Drive*”, el motor comenzará a moverse de un lado a otro, por lo que se recomienda mantener una distancia prudente en caso de que el motor tenga alguna clase de eslabón conectado. Esta acción le dará información al sistema de los parámetros, como inercia, velocidad etc. facilitando la programación general del Driver.

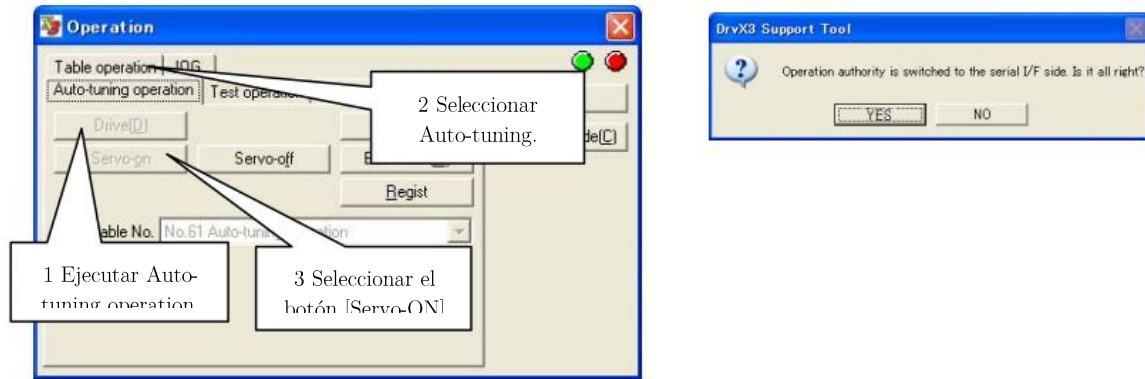


Figura 2.8: Pasos para la operación “Auto Tuning Operation”[24].

Para verificar que la operación anterior fue exitosa, basta con seleccionar la pestaña “JOG”, y de igual manera se activa el servo y se selecciona una de las direcciones a la cual se quiere mover el motor, por defecto si damos clic en el botón “+ direction” el motor se moverá en sentido de las manecillas del reloj. En caso de que llegue a marcar error ya sea por no haberlo desactivado tras realizar la operación de “Auto-tuning” o por presionar el botón de “Stop” sin haber detenido el motor, se puede reiniciar el motor dando clic en “Error Reset”.

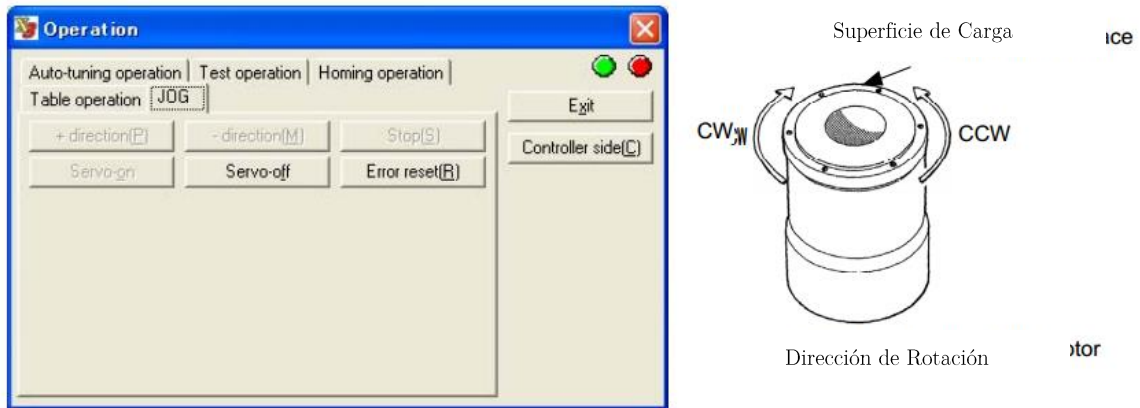


Figura 2.9: Operación Jog para probar la funcionalidad del motor[24].

El motor debe moverse suavemente en la dirección requerida, en caso contrario se debe realizar de nuevo la operación *Auto-tuning*, o en su defecto usar la Terminal de parámetros para reiniciar el motor a su estado de fábrica y volver a iniciar con el procedimiento [24]. En este punto el Servo Driver está listo para ser configurado a la modalidad deseada, ya sea control de Posición, Torque o Velocidad.

B.2. CONFIGURACIÓN ELECTRÓNICA

B.2.1. ANÁLISIS DEL ALGORITMO DE LECTURA DEL ENCODER

Como parte fundamental del trabajo de tesis, radica el hecho de usar dispositivos comerciales y de fácil acceso para el control de otros dispositivos más avanzados y de un grado tecnológico mucho más refinado, en este caso el micro controlador Arduino y el control de los motores de transmisión directa.

Por una parte tenemos que los dispositivos Arduino son tarjetas programables que en los últimos años se han vuelto populares para el manejo de sistemas domóticos[25] y robóticos[25] entre otros, esto por su fácil entorno de programación y su inmensa cantidad de soporte tanto en hardware (“Shields”) y software (Librerías) que va creciendo año tras año.

Por otro lado tenemos a los motores de transmisión directa, los cuales son controlados por servo drivers. Estos motores tienen la particularidad de tener un comportamiento lineal con respecto a un voltaje de control aplicado, convirtiéndolos en un punto importante para el estudio de algoritmos de control en robots manipuladores construidos con esta clase de motores. Además de su característica principal ya mencionada, otra de las características que tienen estos motores es una enorme resolución en su encoder, por lo general superior al millón de pulsos por vuelta, lo que nos da escalas de micro grados por pulso.

Ahora bien, tomando todo esto en cuenta se tiene que enfrentar al hecho de que las características del Arduino podrían no ser las suficientes para usar estos motores a toda su capacidad de no tener en cuenta las siguientes consideraciones:

La forma de trabajo del Arduino es secuencial, lo cual significa que se activan los comandos uno tras otro, por tanto se requiere un determinado tiempo para ejecutar los comandos, este tiempo medido en ciclos de reloj debe considerarse con respecto a la frecuencia máxima de la señal de entrada para conocer la frecuencia de trabajo del micro controlador mínima requerida para poder realizar la tarea requerida sin fallas en la lectura, esta frecuencia puede ser calculada a través de la expresión (2.1).

$$f_{Lectura} = f_{señal_max} * C_{clock}, \quad (2.1)$$

donde $f_{señal_max}$ es la frecuencia máxima de la señal de posición del encoder de cuadratura, debe tomarse en cuenta que este valor es obtenido de la frecuencia máxima de una de las señales multiplicada por el número de estados o combinaciones que se forman con ambas

señales en cuadratura. C_{clock} es el número de ciclos de reloj que tarda nuestro micro controlador en realizar las tareas deseadas.

Tomando en cuenta la frecuencia máxima de las señales del encoder del motor, en este caso de 750 kHz para cada señal A y B [24], y tomando en cuenta el desfase de 90° entre las señales, se tiene que la frecuencia para los estados generados por las señales son 4, por lo que la frecuencia máxima de la señal en cuadratura completa es de 3 MHz.

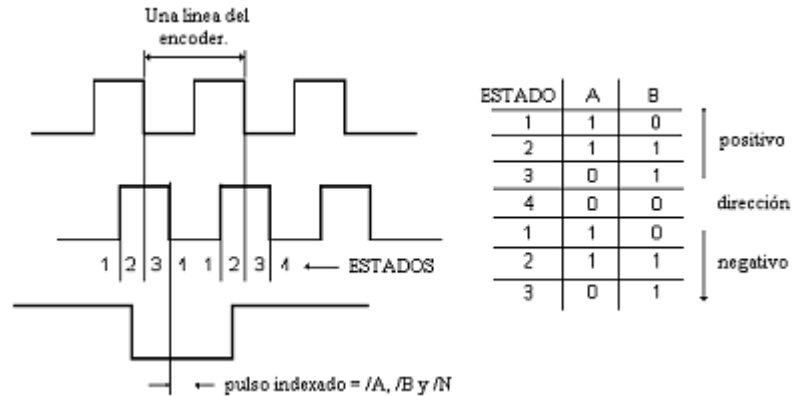


Figura 2.10: Estados encontrados en el encoder de cuadratura.

Ahora, tomando en cuenta el tiempo de lectura y almacenaje del programa en Arduino que es en total de **120 ciclos de reloj** con los comandos estándar, tenemos que la velocidad de reloj del microcontrolador debe ser mínimo de **360 MHz** para usar el motor a la máxima resolución del encoder, por lo que las únicas versiones de Arduino que podrían ejecutar la tarea sería el Arduino Galileo o el Arduino Edison, cuyas frecuencias de reloj son de 400 MHz y 500 MHz, respectivamente. Adicionalmente hay que reiterar que esto es solo para la lectura de un solo encoder, por lo que se requeriría un dispositivo de no menos de 1080 MHz para realizar la lectura de los tres encoders que conforman el ROTRADI completo o el uso de 3 microcontroladores separados que permitan la lectura individual de los encoder. Adicionalmente, hay que considerar que se requeriría de ciclos de reloj adicionales para llevar a cabo todas las conversiones, cálculos y demás transformaciones requeridos para el programa de control.

B.2.2 CONFIGURACIÓN Y ESCALAMIENTO DE ENCODER DEL ROTRADI

Dado que la frecuencia de trabajo de las versiones de Arduino Uno, Mega y Due es muy inferior a la requerida para generar una lectura eficiente de todo el encoder, se recurre al re-escalamiento del encoder del motor. Esta opción es permitida a través de un software

proporcionado por el fabricante del servo driver el cual permite realizar una serie de ajustes dependiendo de las características que se requieran, tales como funcionalidad en base al tipo de control para lo que se busque controlar, ya sea torque/par, posición, velocidad, entre otros, para nuestra aplicación se desea controlar el parámetro de torque, esto se programa desde el registro de parámetros (Parameter setting) en el menú del programa de configuración del fabricante como se muestra en la figura 2.11.

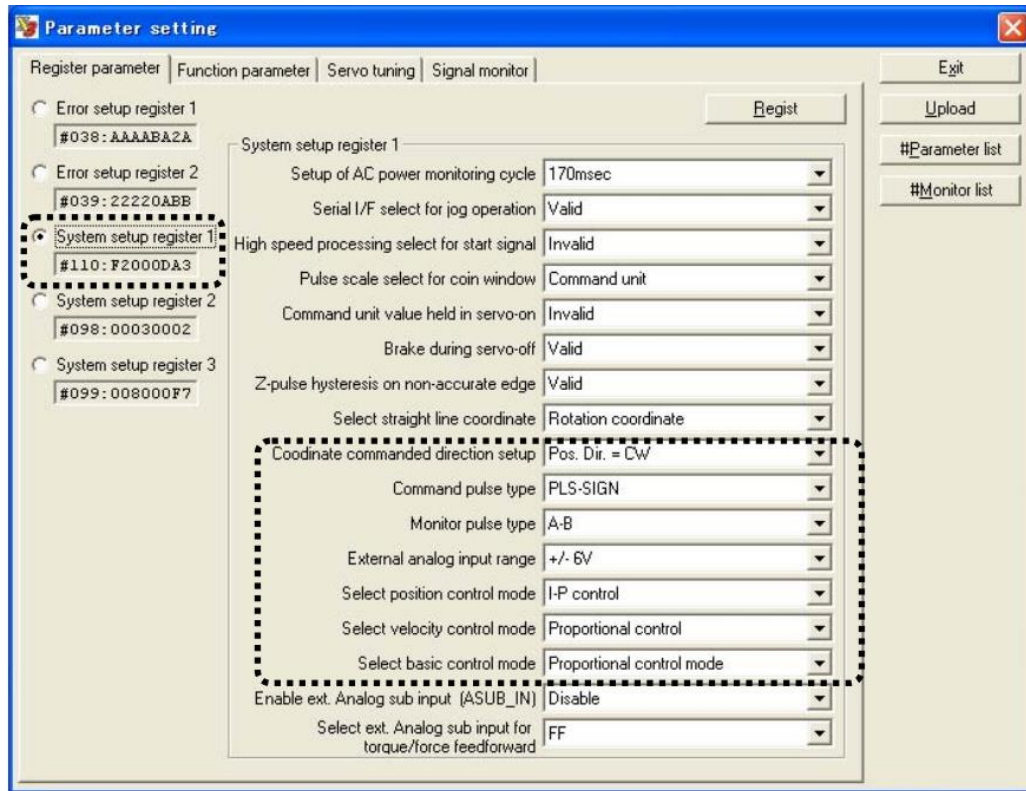


Figura 2.11: Ventana de configuración de parámetros.

En esta ventana se elige el rango de voltaje de control, modo de funcionamiento del encoder (cuadratura o pulsos) y la variable a controlar (velocidad, posición, torque). Ahora bien cada configuración de pines varía dependiendo de la variable que se quiere controlar, para la variable torque, su configuración es la más reducida en cuanto se refiere a pines de entrada y salida siendo el siguiente su diagrama de conexiones mostrado en la figura 2.12.

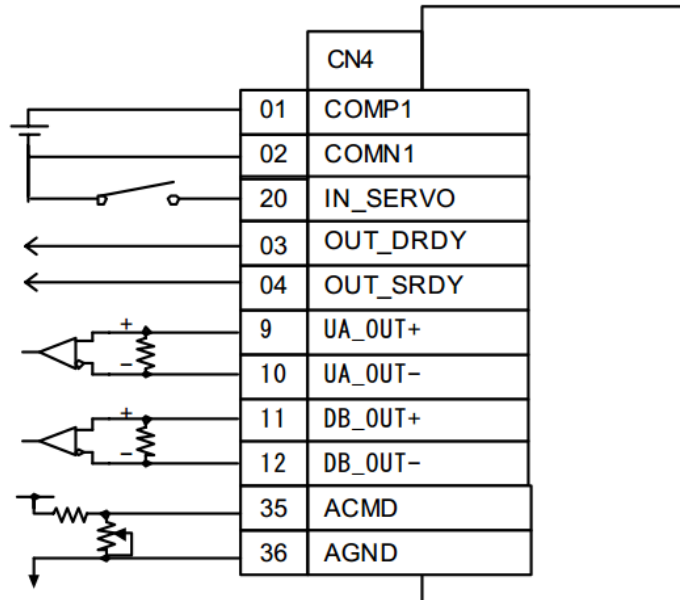


Figura 2.12: Diagrama general de conexiones para el puerto CN4 del servo driver [24].

En el diagrama de conexiones podemos apreciar que el driver devuelve 4 tipos de señales; en las terminales 9 y 10 se tienen las señales A y $-A$, y en las terminales 11 y 12 las señales B y $-B$, respectivamente. Para nuestros propósitos, sólo se requiere la lectura de un par de señales para obtener tanto el desplazamiento como el sentido de movimiento, ya sea A y B o de $-A$ y $-B$. Los pines 1 y 2 son los correspondientes a la alimentación del circuito DC del driver, el pin 20 es el pin de activación del motor, estando encendido cuando está conectado a GND, los pines 3 y 4 son sólo indicadores los cuales están incluidos en el mismo driver los cuales nos indican si hay algún error en el servo y de ser así el servo no funcionará. El pin 35 es la entrada de voltaje analógico con el cual se controla el motor y finalmente el pin 36 es GND para el encoder.

Procediendo a la reducción de la resolución del encoder, para este modelo de driver y motor se utiliza una reducción de 64 veces, cambiando de los 2621440 pulsos por vuelta una de 40960 pulsos por vuelta, lo que nos da en total una resolución de 0.00878 grados por pulso, esto se obtiene a través de dos comandos de escalamiento que se usan en la terminal de puerto RC232 del driver, esto se hace a través de los siguientes pasos.

-Se selecciona la opción Terminal en la ventana principal.

-Una vez abierta se introduce el comando `#302`, el cual nos permite conocer la resolución máxima del encoder.

-Se introduce el comando `#112=****`, donde `****` es el valor devuelto por `#302`.

APÉNDICE C: PRUEBAS DE PLATAFORMA EXPERIMENTAL DE UN PÉNDULO DE TRANSMISIÓN DIRECTA.

C.1. OBTENCIÓN DE PARÁMETROS DE UN ROBOT DE 1 GRADO DE LIBERTAD (PÉNDULO)

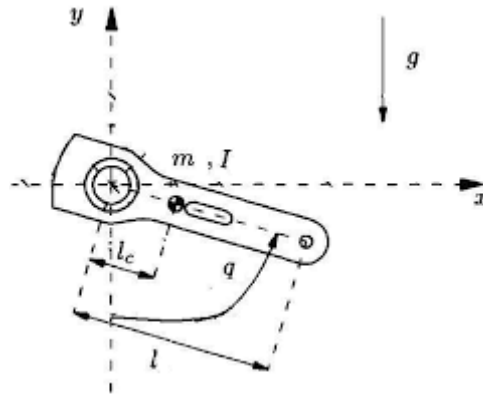


Figura C.1: Robot de 1 grado de libertad (Péndulo).

Parte fundamental del correcto funcionamiento del sistema radica en la caracterización del sistema diseñado en conjunto con el robot, es necesario entender el hecho de el torque o par, va a estar dado por el voltaje que se implemente en el sistema, el cual va a corresponder a una conversión generada por el driver, lo que se puede expresar de la siguiente forma:

$$\tau = kV, \quad (C.1)$$

donde V es el voltaje analógico suministrado al driver, k es una constante obtenida a través de una caracterización descrita más adelante y τ es el par de salida del motor.

Para la obtención del parámetro k , es necesario realizar una relación de Torque-Voltaje, a través de las lecturas del ángulo se obtendrá una relación de Ángulo-Torque, con la cual se generará una serie de posibles valores para k .

Sustituyendo en la ecuación (C.1) únicamente la ecuación de par gravitacional (C.2), tenemos:

$$\tau = mgl_c \sin(q) \quad (C.2)$$

$$mgl_c \sin(q) = kV \quad (C.3)$$

$$\frac{mgl_c \sin(q)}{V} = k \quad (C.4)$$

donde m es la masa, g es la aceleración de la gravedad, l_c es la distancia al centro de masa del eslabón y q es la posición angular del eslabón (Figura C.1).

Hay que tener presente que el valor máximo entregado por el DAC es de $\pm 5V$, lo cual implica que el torque máximo que nos va a entregar el motor es de $\pm 7.5 Nm$. Adicionalmente se está dando por hecho que el torque entregado por el motor es lineal y proporcional al voltaje de entrada, como lo garantiza el fabricante.

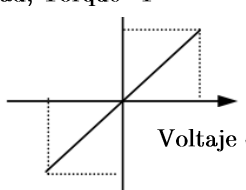
Señal	Descripción	I/O	
ACMD_IN+	Comando analógico + (Control de Torque/Par)	CN4-35	<p>Velocidad, Torque *1</p>  <p>La referencia de torque/par y voltaje se obtiene de las siguientes formular.</p> <p>Referencia de torque/par = máximo torque x #122 *0.0001</p> <p>Voltaje de referencia = 6 o 10V editables en el registro de sistema 1.</p>
ACMD_IN-	Comando analógico - (Tierra analógica)	CN4-36	

Figura C.2: Gráfica de referencia de parámetros de salida respecto al voltaje de entrada.

Continuando con la caracterización del sistema, se procede a generar una señal analógica con nuestro circuito experimental (revisar capítulo 2), en donde se va incrementando tanto el voltaje positivo como el negativo hasta obtener un mapeo completo del ángulo que alcanza con cada cambio de voltaje, con lo cual se obtiene a su vez el par y la posible constante k .

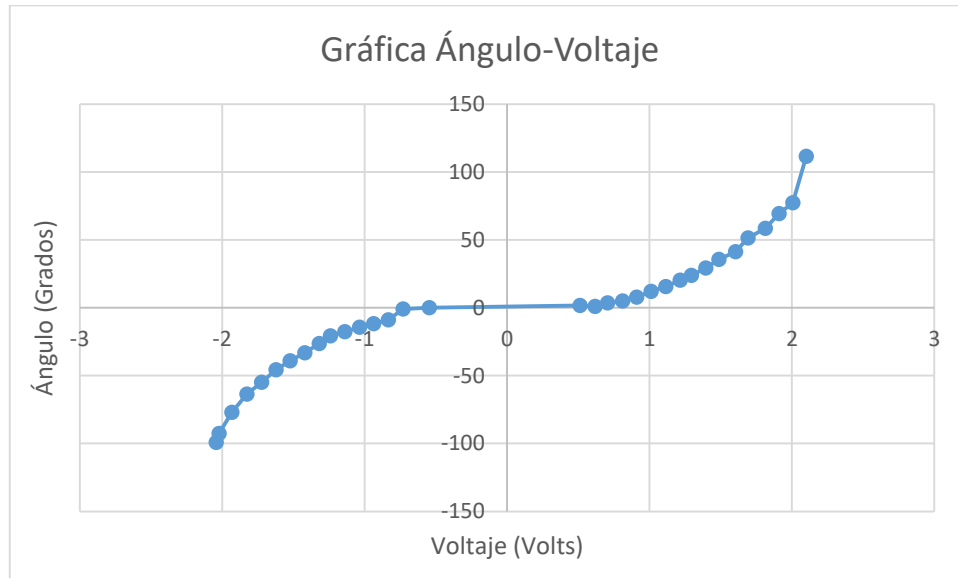


Figura C.3: Gráfica del ángulo alcanzado por el robot con respecto al voltaje analógico implementado.

Sustituyendo los valores de voltaje y ángulo en la ecuación (C.4) obtenemos una tabla con una serie de valores. De la tabla obtenida se eligen algunos valores no extremos, por lo que se define que los posibles valores de k están entre 1.8 y 2, por lo que se elige el valor de $k = 1.9$.

C.2. PROGRAMA DE CONTROL

Como punto de referencia se procederá a la implementación de un PD para poder probar más adelante otros esquemas de control. El algoritmo de control proporcional derivativo (PD) es el más conocido y empleado en el área de control automático, este algoritmo tiene un funcionamiento variable, depende de la calibración que se le de a dos valores constantes:

$$\tau = K_p \ddot{q} - K_v \dot{q} + g(q). \quad (\text{C.5})$$

Este esquema de control es implementado en un programa diseñado en el IDE de Arduino, cuyo funcionamiento y ejecución general se ilustra en la Figura C.4.

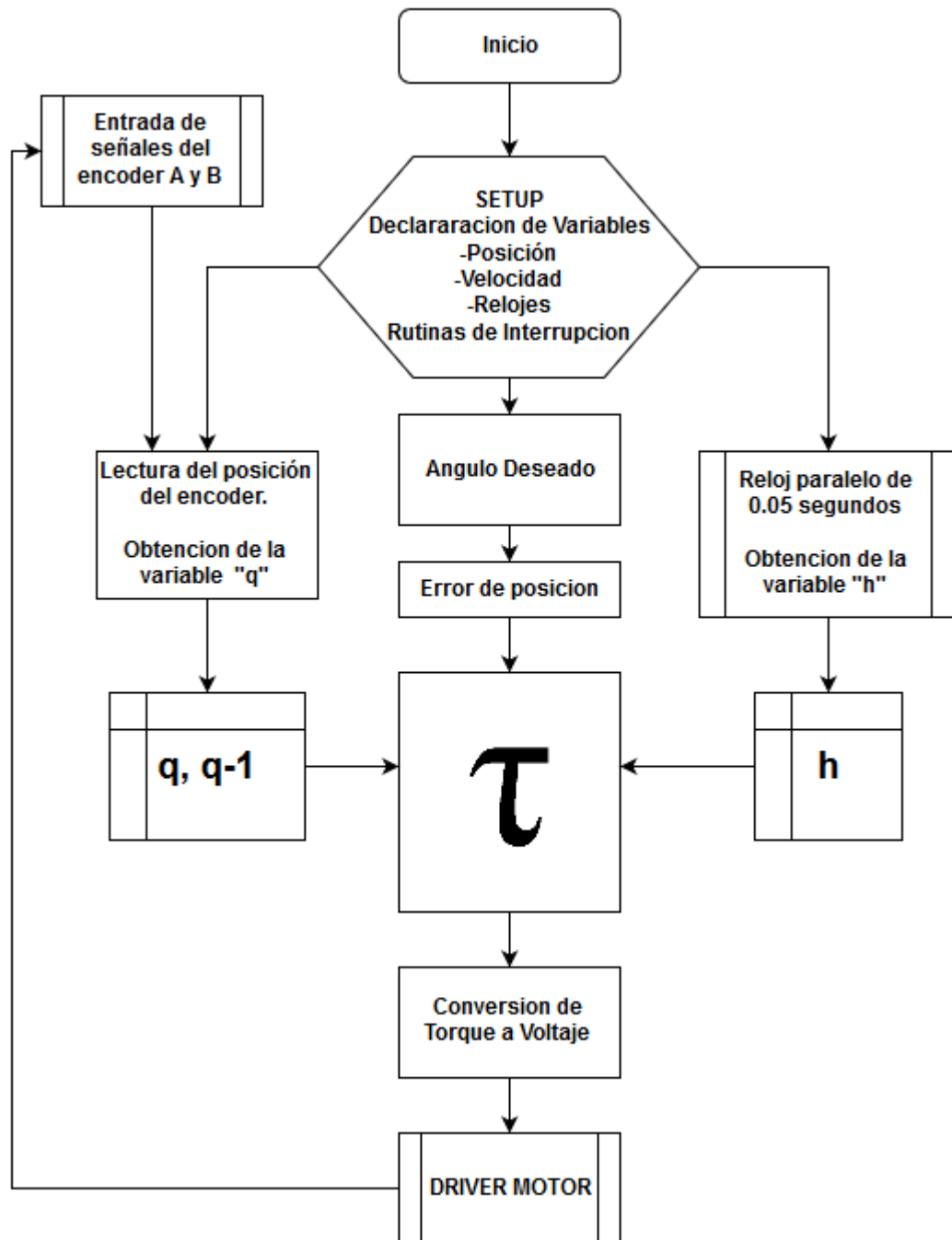


Figura C.4: Diagrama de flujo del funcionamiento del programa.

En un inicio tras encender el micro controlador, se inician los preparativos para el funcionamiento del programa, en donde se declaran todas las variables a ocupar, se inician las rutinas de interrupción y se activa el reloj Timer One.

Una vez listo el sistema se inicia un reloj independiente, el cual va aumentando cada 0.05 segundos y nos devuelve la variable “h”, la cual sirve para realizar cálculos de velocidad y aceleración. De igual forma, se inician los servicios de interrupción que generan la lectura del encoder, con esto se obtiene la variable q que nos da la posición actual del robot.

Se hace la lectura del ángulo deseado y esto genera un error de posición dado por la diferencia entre la posición deseada y la actual:

$$\tilde{q} = q_{des} - q, \quad (C.6)$$

donde \tilde{q} es el error de posición, q_{des} es la posición deseada y q es la posición actual del robot.

Los datos de tiempo y posición se guardan internamente en una base de datos temporal la cual almacena únicamente los dos últimos valores de posición y sus respectivos valores de tiempo con los cuales se genera la velocidad actual del sistema:

$$\dot{q} = \frac{q - q_{anterior}}{0.05}, \quad (C.7)$$

donde \dot{q} es la velocidad angular y $q_{anterior}$ es el valor de q anterior.

Ahora todos los datos obtenidos entran en un bloque donde se calcula el torque a través de una ecuación de control, en este caso como medio de referencia para futuros esquemas de control se ocupa el PD, esto es:

$$\tau = \frac{mgl_c \text{Sen}(q)}{k} + k_p \tilde{q} - k_v \dot{q} \quad (C.8)$$

A continuación se usa una rutina de conversión de torque a voltaje en el cual el voltaje deseado es convertido en un valor binario y este es enviado a través del bus de datos de 10 bits hacia el circuito del DAC para una conversión analógica, finalmente esta señal entra en el Driver el cual genera el par deseado. Esto se retroalimenta a través del encoder generando así un ciclo de repetición indefinido el cual lleva al robot a la posición deseada.

C.3. PRUEBAS

Con el sistema en operación se procede a realizar una serie de experimentos en donde el sistema se pone a prueba con una serie de ángulos deseados fácilmente reconocibles en cualquier cuadrante, en este caso los ángulos elegidos son $\pm 135^\circ$, $\pm 90^\circ$ y $\pm 45^\circ$, ahora bien se inician las primeras pruebas con los ángulos deseados, y obtenemos los siguientes resultados.



Figura C.5: Péndulo ubicado a 135° .



Figura C.6: Péndulo ubicado a 90° .

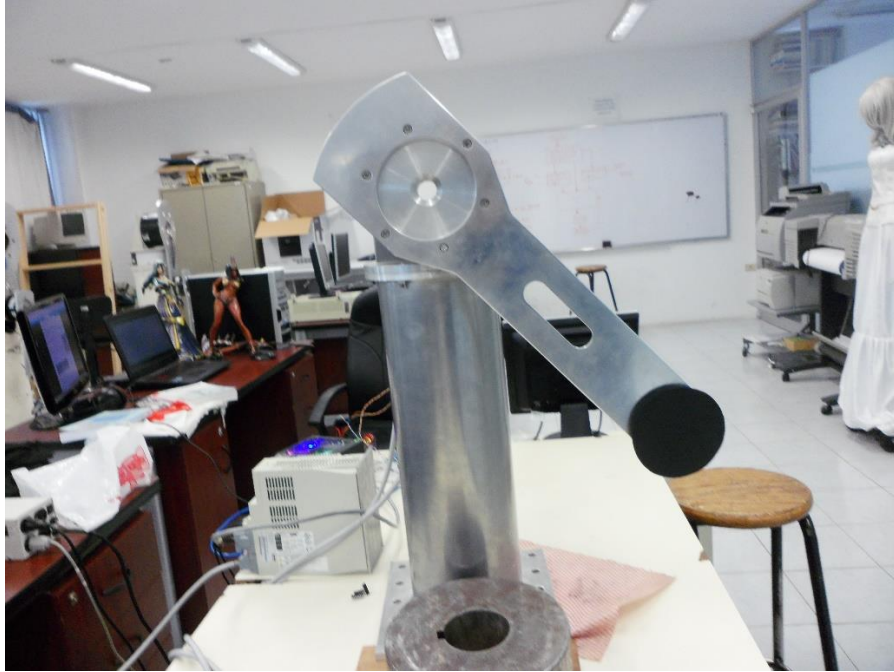


Figura C.7: Péndulo ubicado a 45° .

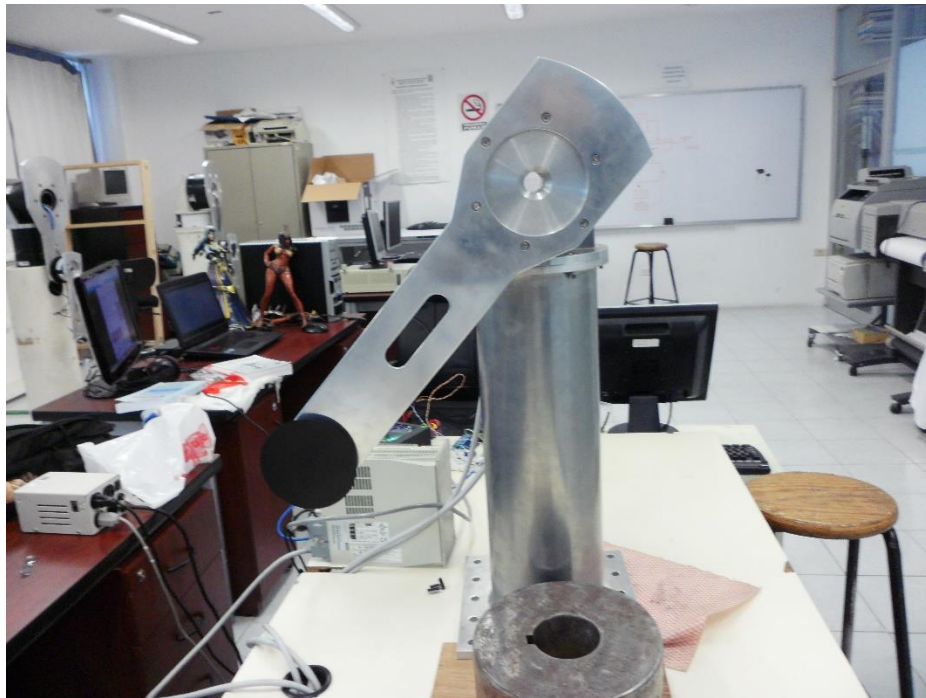


Figura C.8: Péndulo ubicado a -45° .



Figura C.9: Péndulo ubicado a -90° .



Figura C.10: Péndulo ubicado a -135° .

Como se puede ver, los ángulos son aparentemente los correctos, pero esto se puede apreciar de una mejor manera si se grafica el ángulo alcanzado así como la evolución del torque implementado para cada caso.

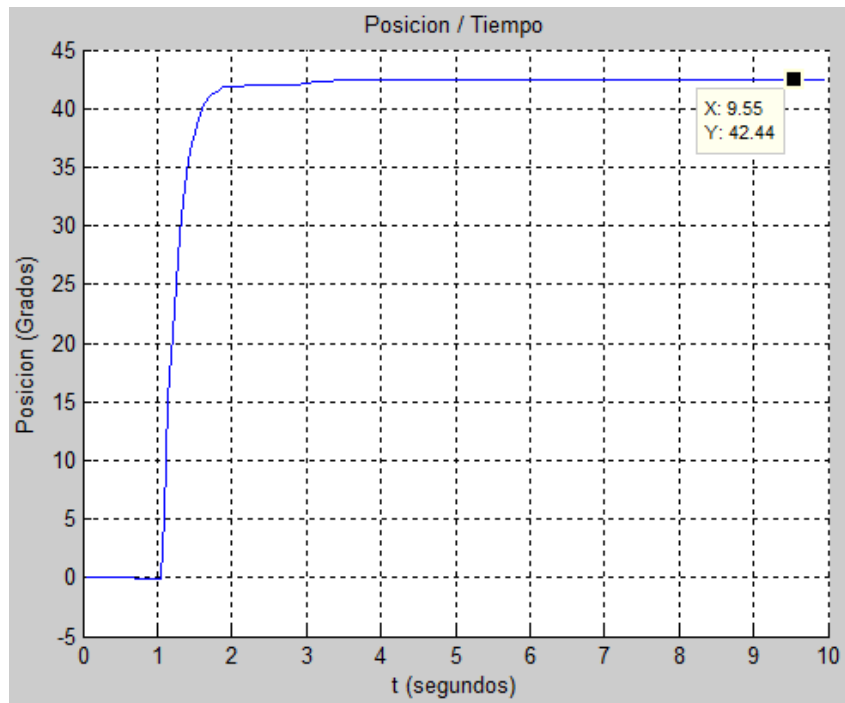


Figura C.11: Gráfica de posición tiempo para $q = 45^\circ$, con un error de 2.56° .

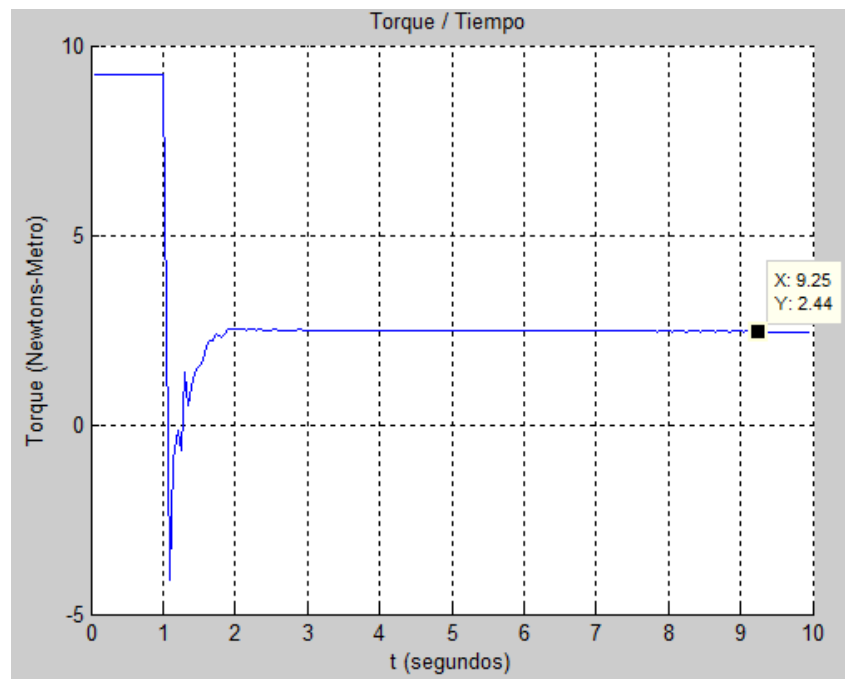


Figura C.12: Gráfica de torque para $q = 45^\circ$.

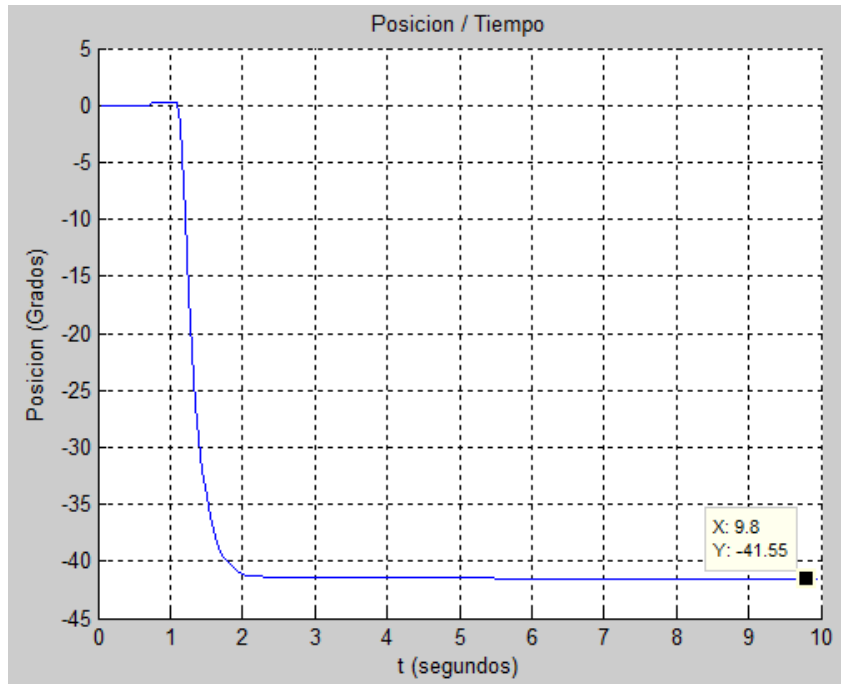


Figura C.13: Gráfica de posición tiempo para $q = -45^\circ$, con un error de -3.45° .

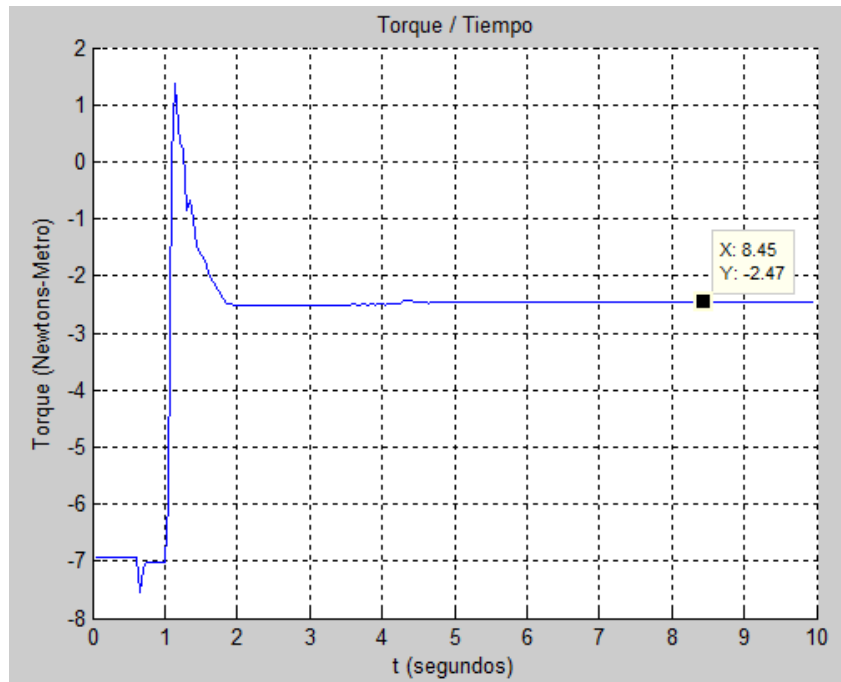


Figura C.14: Gráfica de torque para $q = -45^\circ$.

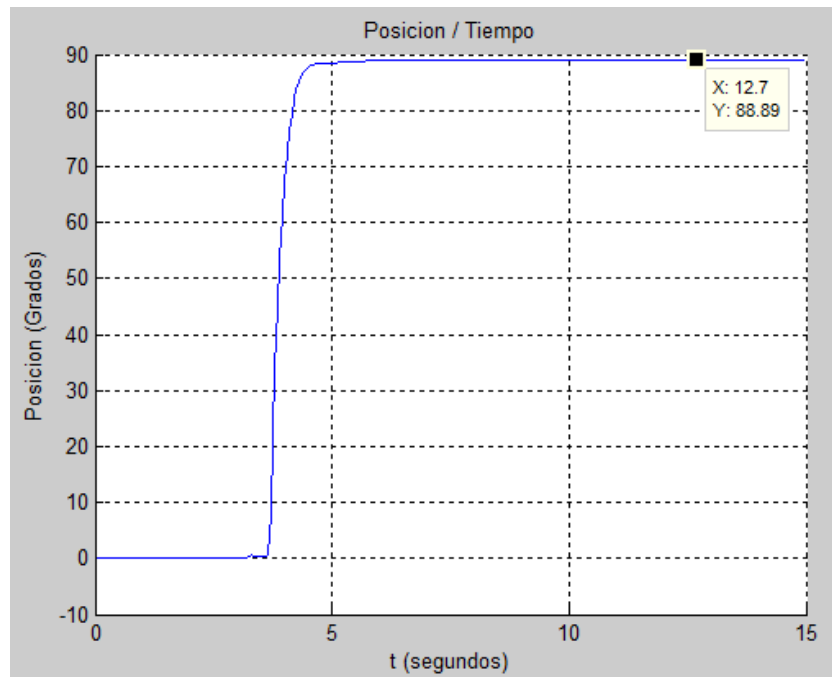


Figura C.15: Gráfica de posición tiempo para $q = 90^\circ$, con un error de 1.11° .

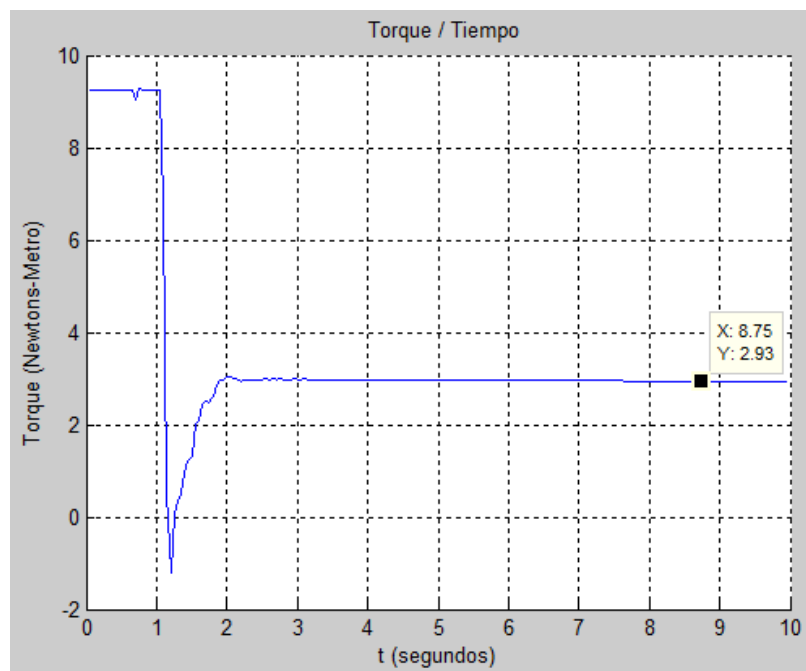


Figura C.16: Gráfica de torque para $q = 90^\circ$.

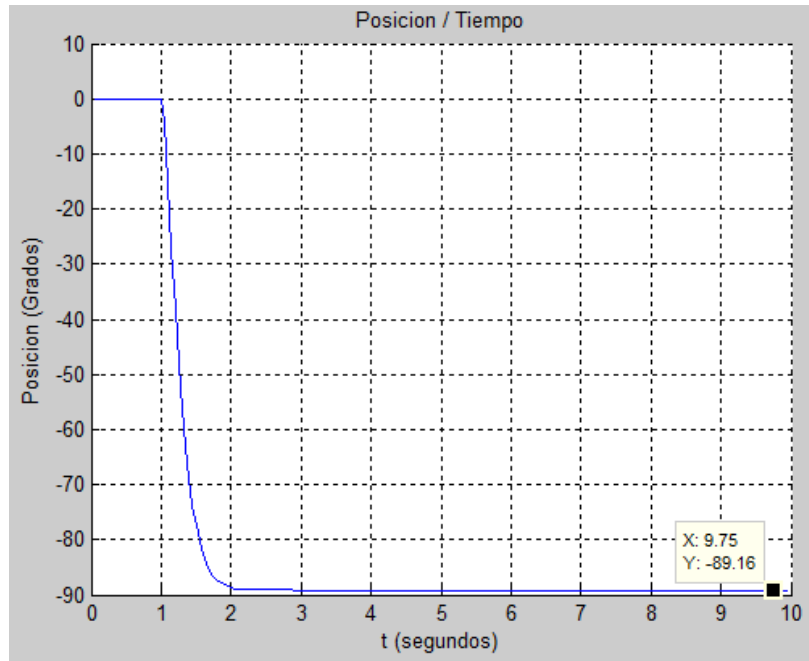


Figura C.17: Gráfica de posición tiempo para $q = -90^\circ$, con un error de -0.84° .

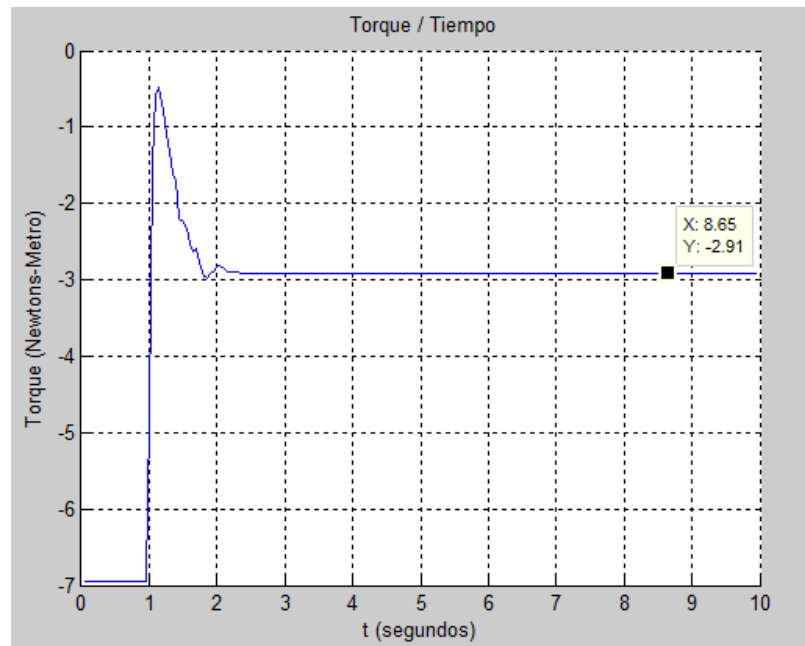


Figura C.18: Gráfica de torque para $q = -90^\circ$.

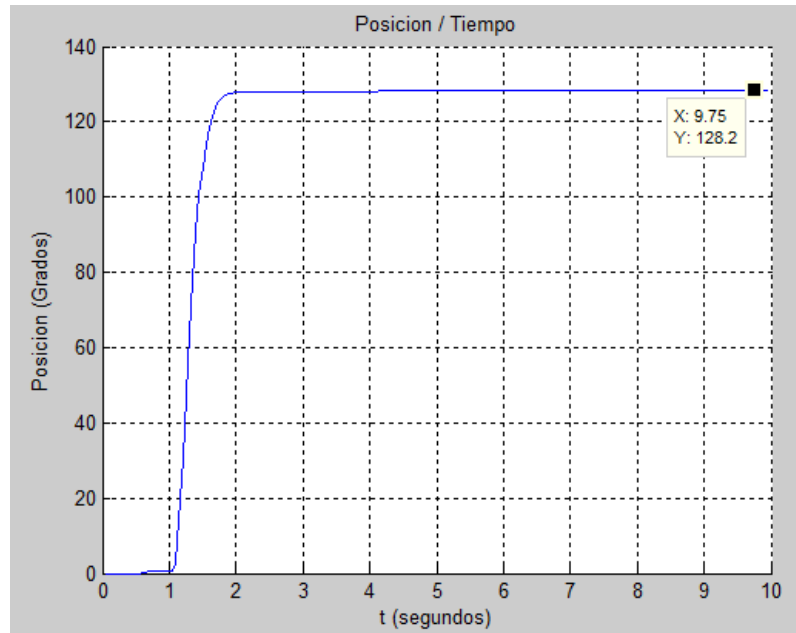


Figura C.19: Gráfica de posición tiempo para $q = 135^\circ$, con un error de 6.8° .

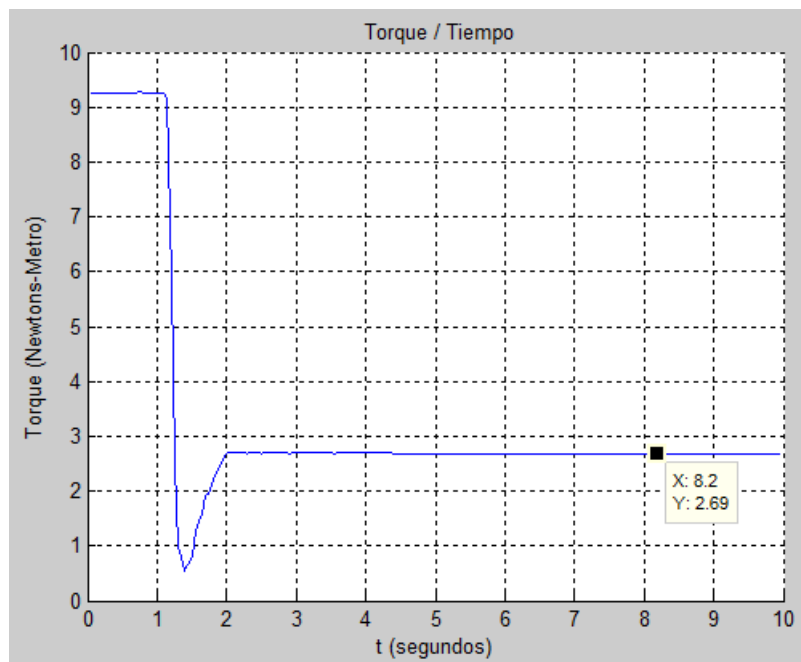


Figura C.20: Gráfica de torque para $q = 135^\circ$.

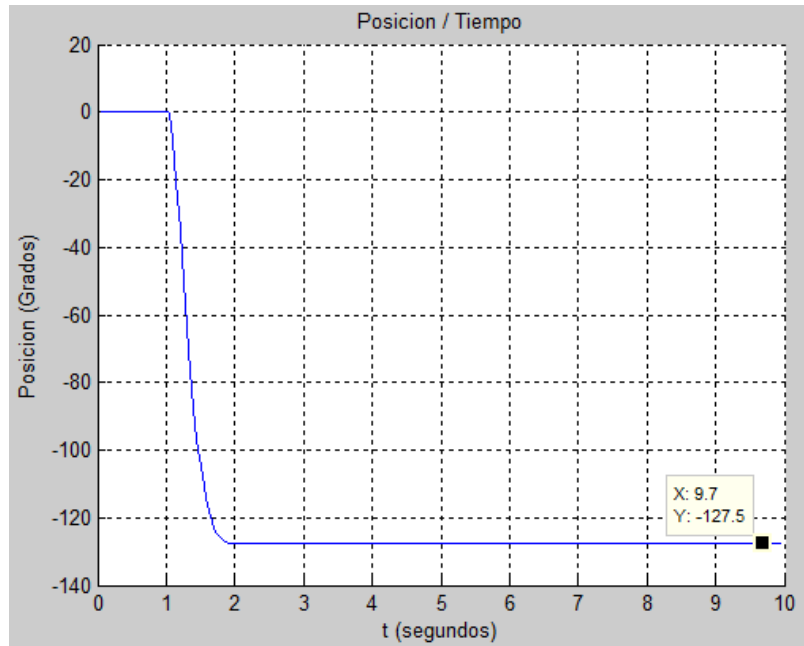


Figura C.21: Gráfica de posición tiempo para $q = -135^\circ$, con un error de -8.5° .

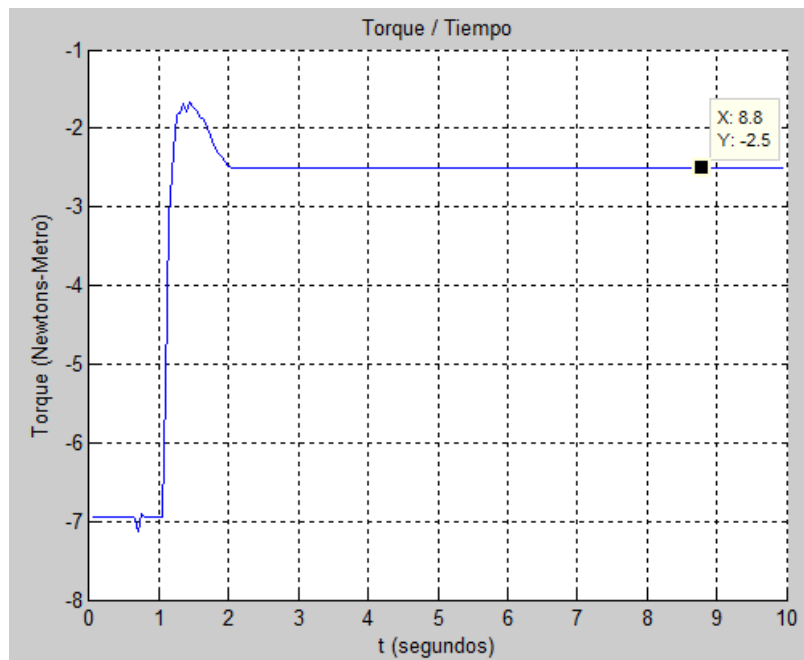


Figura C.22: Gráfica de torque para $q = -135^\circ$.

Como se puede ver en cada caso, el control PD reduce el sobre impulso y el tiempo de estabilización, por lo cual tendrá el efecto de incrementar la estabilidad del sistema mejorando su respuesta, no obstante sin una calibración más precisa de las constantes K_p y K_v no llegará a los valores deseados fácilmente.

APÉNDICE D: PROGRAMA DE CONTROL EN ARDUINO

Programa Principal

En caso de dudas para su implementación escribir al siguiente correo: Xherael@live.com.mx

```

#include <math.h>
#include <TimerOne.h>
int ledPin1[] = {22,23,24,25,26,27,28,29,30,31};
int ledPin2[] = {32,33,34,35,36,37,38,39,40,41};
int ledPin3[] = {42,43,44,45,46,47,48,49,50,51};
const float DROLL[][4]= {
};
int KI=8+1;
int SEN=A0;
volatile float encoderPos1 = 0; //CONTEO ENCODER 1
volatile float PosTilde1 = 1; //
volatile float encoderPos2 = 0; //CONTEO ENCODER 2
volatile float PosTilde2 = 1; //
volatile float encoderPos3 = 0; //CONTEO ENCODER 3
volatile float PosTilde3 = 1; //
int SA1= LOW; //ESTADO INICIAL ENCODER
int SB1= LOW; //
int SA2= LOW; //ESTADO INICIAL ENCODER
int SB2= LOW; //
int SA3= LOW; //ESTADO INICIAL ENCODER
int SB3= LOW; //
#define pinA1 21
#define pinB1 20
#define pinA2 19
#define pinB2 18
#define pinA3 2
#define pinB3 3
int DATA1=0;
int DATA2=0;
int DATA3=0;
float q1=0;
float q2=0;
float q3=0;
volatile float h=0.5;
volatile float hanterior=0;
const double l3=0.35+0.1859; //m
const double lc3=0.051; //m
const double m3=1.2+0.3; //kg
const double l2=0.35;
const double lc2=0.269; //m
const double m2=8.43; //kg
const double l1=1.19;
const double g=9.81; //m/s2
volatile int qd1=0;
volatile int qd2=0;
volatile int qd3=0;
float q1tilde=0;
float q2tilde=0;
float q3tilde=0;

```

```
float q1punto=0;
float q1anterior=0;
float q2punto=0;
float q2anterior=0;
float q3punto=0;
float q3anterior=0;
int qd1anterior=0;
int qd2anterior=0;
int qd3anterior=0;
float kp1=0;
float kv1=0;
float K1=0;
float kp2=0;
float kv2=0;
float K2=0;
float kp3=0;
float kv3=0;
float K3=0;
float kp11=0;
float kp21=0;
float kp31=0;
float kv11=0;
float kv21=0;
float kv31=0;
float TAU1=0;
float TAU2=0;
float TAU3=0;
float TAUs=0;
float qd=0;
int CHO=0;
int i;
int TA=0;
int TO=0;
float XFORCE=0;
float YFORCE=0;
float ZFORCE=0;
float X=0;
float Y=0;
float Z=0;
float Xd=0;
float Yd=0;
float Zd=0;
float XF=0;
float YF=0;
float ZF=0;
float XFd=0;
float YFd=0;
float ZFd=0;
float Xpunto=0;
float Ypunto=0;
float Zpunto=0;
float Xtilde=0;
float Ytilde=0;
float Ztilde=0;
float XFtilde=0;
float YFtilde=0;
float ZFtilde=0;
float XImp=0;
float YImp=0;
float ZImp=0;
float XImpanterior=0;
float YImpanterior=0;
float ZImpanterior=0;
float XImppunto=0;
```

```

float YImpunto=0;
float ZImpunto=0;
float Xanterior=0;
float Yanterior=0;
float Zanterior=0;
float X1=0;
float Y1=0;
float Z1=0;
float X2=0;
float Y2=0;
float Z2=0;
float X3=0;
float Y3=0;
float Z3=0;
float Ks=0;
float w1=0.16398;
float w2=0.01284;
int TIMER=0;
int TIMERB=0;
int flagC=0;
int flagB=0;
void setup()
{
  Serial.begin(9600);
  pinMode(SEN, INPUT);
  pinMode(pinA1, INPUT_PULLUP);
  pinMode(pinB1, INPUT_PULLUP);
  pinMode(pinA2, INPUT_PULLUP);
  pinMode(pinB2, INPUT_PULLUP);
  pinMode(pinA3, INPUT_PULLUP);
  pinMode(pinB3, INPUT_PULLUP);
  for (long int i =0;i<10;i++)
  { pinMode(ledPin1[i], OUTPUT);
    pinMode(ledPin2[i], OUTPUT);
    pinMode(ledPin3[i], OUTPUT);}
  attachInterrupt( digitalPinToInterrupt(pinA1), EncoderA1, CHANGE );
  attachInterrupt( digitalPinToInterrupt(pinB1), EncoderB1, CHANGE );
  attachInterrupt( digitalPinToInterrupt(pinA2), EncoderA2, CHANGE );
  attachInterrupt( digitalPinToInterrupt(pinB2), EncoderB2, CHANGE );
  attachInterrupt( digitalPinToInterrupt(pinA3), EncoderA3, CHANGE );
  attachInterrupt( digitalPinToInterrupt(pinB3), EncoderB3, CHANGE );
  Timer1.initialize(50000);
  Timer1.attachInterrupt(Hgen);}
void loop()
{
  q3=(encoderPos3*360.0/10240.0)*PI/180.0;
  q2=(encoderPos2*360.0/10240.0)*PI/180.0;
  q1=(encoderPos1*360.0/10240.0)*PI/180.0;
  Binary1(DATA1);
  Binary2(DATA2);
  Binary3(DATA3);
  if (hanterior != h) {Serial.print(h);Serial.print(","); Serial.print(XFORCE);Serial.print(","); Serial.print(X);Serial.print(","); Serial.print(Y); Serial.print(",");
  Serial.print(Z); Serial.print(","); Serial.print(q1*180/PI);Serial.print(","); Serial.print(q2*180/PI); Serial.print(","); Serial.print(q3*180/PI); Serial.print(",");
  Serial.print(TAU1);Serial.print(","); Serial.print(TAU2); Serial.print(","); Serial.println(TAU3); hanterior=h; q1anterior=q1; q2anterior=q2; q3anterior=q3;
  Xanterior=X; Yanterior=Y; Zanterior=Z; XImpanterior=XImp; YImpanterior=YImp; ZImpanterior=ZImp; TIMER++; TO=TO+1;
  TA=(TO/25);} //TA++ o =(TO/20);}
  //TauIMP();
  //TauFOR();
  //TauCAR();
  //TauDN();
  //TauG();
  if(TIMER>150){flagC=1;}
  if(TIMER<150 && flagC == 0){qd1=0.0; qd2=35.0; qd3=50.0; TauDN();}
  //if(TIMER>150 && flagC == 1){TauFOR();}

```

```

if(TIMER>150 && flagC == 1){flagB++; BETA(); TauDN();}
if(flagB>99){TIMERB++;}
if(TIMERB>49){TauG();} //TauFOR
if(TIMERB>99){TIMERB=0; flagB=0;}
REGULADOR();
}

```

Programa de lectura de encoders

```

////////////////////
void Binary1(long num1)
{for (long int i =0;i<10;i++)
  {if (bitRead(num1, i)==1){digitalWrite(ledPin1[i], HIGH);}
   else{digitalWrite(ledPin1[i], LOW);}}}
////////////////////
void Binary2(long num2)
{for (long int i =0;i<10;i++)
  {if (bitRead(num2, i)==1){digitalWrite(ledPin2[i], HIGH);}
   else{digitalWrite(ledPin2[i], LOW);}}}
////////////////////
void Binary3(long num3)
{for (long int i =0;i<10;i++)
  {if (bitRead(num3, i)==1){digitalWrite(ledPin3[i], HIGH);}
   else{digitalWrite(ledPin3[i], LOW);}}}
////////////////////
//////////////////// A //////////////////////
void EncoderA1(){
  SA1= digitalRead(pinA1);
  encoderPos1 += (SA1 != SB1) ? +1 : -1;}
////////////////////
void EncoderB1(){
  SB1= digitalRead(pinB1);
  encoderPos1 += (SA1 == SB1) ? +1 : -1;}
////////////////////
//////////////////// B //////////////////////
void EncoderA2(){
  SA2= digitalRead(pinA2);
  encoderPos2 += (SA2 != SB2) ? +1 : -1;}
////////////////////
void EncoderB2(){
  SB2= digitalRead(pinB2);
  encoderPos2 += (SA2 == SB2) ? +1 : -1;}
////////////////////
//////////////////// C //////////////////////
void EncoderA3(){
  SA3= digitalRead(pinA3);
  encoderPos3 += (SA3 != SB3) ? +1 : -1; }
////////////////////
void EncoderB3(){
  SB3= digitalRead(pinB3);
  encoderPos3 += (SA3 == SB3) ? +1 : -1;}
////////////////////
void Hgen(){h=h+0.05;}

```

Programa regulador de energía

```

void REGULADOR(){
  if(TAU1>12.0){TAU1=12.0;}
  if(TAU1<-12.0){TAU1=-12.0;}
  if(TAU2>40.0){TAU2=40.0;}
  if(TAU2<-40.0){TAU2=-40.0;}
  if(TAU3>3.0){TAU3=3.0;}
  if(TAU3<-3.0){TAU3=-3.0;}
  if(TAU1>=0){DATA1=(-abs(TAU1)*512.0/12.0)-512;}
  if(TAU2>=0){DATA2=(-abs(TAU2)*512.0/40.0)-512;}
  if(TAU3>=0){DATA3=(-abs(TAU3)*512.0/3.2)-512;}
}

```

```

if(TAU1<0){DATA1=((abs(TAU1)*511.0/(12.0-0.225)))+511;}
if(TAU2<0){DATA2=((abs(TAU2)*511.0/(40.0-0.6)))+511;}
if(TAU3<0){DATA3=((abs(TAU3)*511.0/(3.2-0.06)))+511;}
if(DATA1>1022){DATA1=1023;}
if(DATA2>1022){DATA2=1023;}
if(DATA3>1022){DATA3=1023;}
//if(DATA1<0){DATA1=0;}
//if(DATA2<0){DATA2=0;}
//if(DATA3<0){DATA3=0;}
//DATA1=map(TAU1, -12.0, 12.0, 1023.0, 0);
//DATA2=map(TAU2, -40.0, 40.0, 1023.0, 0);
//DATA3=map(TAU3, -3.2, 3.2, 1023.0, 0);
}

```

Programa de control articular

```

void TauDN(){ //modo ANGULAR
//BETA();
///KP TANH 0.4,1.0,1.0, KV 0.4,0.15,0.15
///KP ATAN 0.8,1.0,1.0, KV 0.2,0.2,0.2
q1tilde=qd1*PI/180-q1;
q1punto=((q1-q1anterior))/(0.05);
kp1=(0.8*15.0)*1.88*exp(-0.008*abs(qd1));
kv1=0.40*kp1;
q2tilde=qd2*PI/180-q2;
q2punto=((q2-q2anterior))/(0.05);
kp2=(1.0*40.0)*1.88*exp(-0.008*abs(qd2));
kv2=0.15*kp2;
q3tilde=qd3*PI/180-q3;
q3punto=((q3-q3anterior))/(0.05);
kp3=(1.0*4.0)*1.88*exp(-0.008*abs(qd3));
kv3=0.20*kp3;
  X=cos(q1)*(l2*sin(q2)+l3*sin(q2+q3));
  Y=sin(q1)*(l2*sin(q2)+l3*sin(q2+q3));
  Z=l1-l2*cos(q2)-l3*cos(q2+q3);
K2=1.5;
K3=1.8;
//
  TAU1=0+(kp1*tanh(q1tilde))-(kv1*tanh(q1punto));
  TAU2=((m2*g*l2*sin(q2)+(m3*g*l2*sin(q2)+(m3*g*l3*sin((q2+q3)))))/K2+(kp2*tanh(q2tilde))-(kv2*tanh(q2punto));
  TAU3=(m3*g*l3*sin((q2+q3)))/K3+(kp3*tanh(q3tilde))-(kv3*tanh(q3punto));
// TAU1=0+(kp1*(q1tilde))-(kv1*(q1punto));
// TAU2=((m2*g*l2*sin(q2)+(m3*g*l2*sin(q2)+(m3*g*l3*sin((q2+q3)))))/K2+((kp2*(q2tilde))-(kv2*(q2punto));
// TAU3=(m3*g*l3*sin((q2+q3)))/K3+(kp3*(q3tilde))-(kv3*(q3punto));
// TAU1=0+(kp1*atan(q1tilde))-(kv1*atan(q1punto));
// TAU2=((m2*g*l2*sin(q2)+(m3*g*l2*sin(q2)+(m3*g*l3*sin((q2+q3)))))/K2+(kp2*atan(q2tilde))-(kv2*atan(q2punto));
// TAU3=(m3*g*l3*sin((q2+q3)))/K3+(kp3*atan(q3tilde))-(kv3*atan(q3punto));
}

```

Programa de control cartesiano

```

////////// TAU //////////
void TauCAR(){ //modo cartesiano
BETA();
///TANH KP100,60,15, KV,50,20,3
kp1=100;//F
kp2=60;//125
kp3=15;//125
kv1=50;//F
kv2=20;
kv3=3;
  X=cos(q1)*(l2*sin(q2)+l3*sin(q2+q3));/(w1+w2)*sin(q1);
  Y=sin(q1)*(l2*sin(q2)+l3*sin(q2+q3));/-(w1+w2)*cos(q1);
  Z=l1-l2*cos(q2)-l3*cos(q2+q3);
Xtilde=Xd-X;
Ytilde=Yd-Y;
Ztilde=Zd-Z;

```

```

Xpunto=(X-Xanterior)/0.05;
Ypunto=(Y-Yanterior)/0.05;
Zpunto=(Z-Zanterior)/0.05;
K2=1.7;
K3=2.0;
//////////Componentes del jacobiano//////////
X1=sin(q1)*(sin(q2)*l2+sin((q2+q3))*l3);/+cos(q1)*(w1+w2);
Y1=cos(q1)*(sin(q2)*l2+sin((q2+q3))*l3);/+sin(q1)*(w1+w2);
Z1=0;
X2=cos(q1)*(cos(q2)*l2+cos(q2+q3)*l3);
Y2=sin(q1)*(cos(q2)*l2+cos(q2+q3)*l3);
Z2=sin(q2)*l2+sin(q2+q3)*l3;
X3=cos(q1)*cos(q2+q3)*l3;
Y3=sin(q1)*cos(q2+q3)*l3;
Z3=sin(q2+q3)*l3;
//////////
//TAU1= X1*(kp1*tanh(Xtilde)-kv1*tanh(Xpunto))+Y1*(kp1*tanh(Ytilde)-kv1*tanh(Ypunto))+Z1*(kp1*tanh(Ztilde)-kv1*tanh(Zpunto));
//TAU2=
X2*(kp2*tanh(Xtilde)-kv2*tanh(Xpunto))+Y2*(kp2*tanh(Ytilde)-kv2*tanh(Ypunto))+Z2*(kp2*tanh(Ztilde)-kv2*tanh(Zpunto))+((m2*g*l2*sin(q2))+(m3*g*l2*sin(q2))+(m3*g*l3*sin(q2+q3)))/K2;
//TAU3=
X3*(kp3*tanh(Xtilde)-kv3*tanh(Xpunto))+Y3*(kp3*tanh(Ytilde)-kv3*tanh(Ypunto))+Z3*(kp3*tanh(Ztilde)-kv3*tanh(Zpunto))+((m3*g*l3*sin((q2+q3)))/K3;
//////////
//
TAU1= X1*(kp1*atan(Xtilde)-kv1*atan(Xpunto))+Y1*(kp1*atan(Ytilde)-kv1*atan(Ypunto))+Z1*(kp1*atan(Ztilde)-kv1*atan(Zpunto));
TAU2=
X2*(kp2*atan(Xtilde)-kv2*atan(Xpunto))+Y2*(kp2*atan(Ytilde)-kv2*atan(Ypunto))+Z2*(kp2*atan(Ztilde)-kv2*atan(Zpunto))+((m2*g*l2*sin(q2))+(m3*g*l2*sin(q2))+(m3*g*l3*sin(q2+q3)))/K2;
TAU3=
X3*(kp3*atan(Xtilde)-kv3*atan(Xpunto))+Y3*(kp3*atan(Ytilde)-kv3*atan(Ypunto))+Z3*(kp3*atan(Ztilde)-kv3*atan(Zpunto))+((m3*g*l3*sin((q2+q3)))/K3;
//////////
//TAU1= X1*(kp1*(Xtilde)-kv1*(Xpunto))+Y1*(kp1*(Ytilde)-kv1*(Ypunto))+Z1*(kp1*(Ztilde)-kv1*(Zpunto));
//TAU2=
X2*(kp2*(Xtilde)-kv2*(Xpunto))+Y2*(kp2*(Ytilde)-kv2*(Ypunto))+Z2*(kp2*(Ztilde)-kv2*(Zpunto))+((m2*g*l2*sin(q2))+(m3*g*l2*sin(q2))+(m3*g*l3*sin(q2+q3)))/K2;
//TAU3= X3*(kp3*(Xtilde)-kv3*(Xpunto))+Y3*(kp3*(Ytilde)-kv3*(Ypunto))+Z3*(kp3*(Ztilde)-kv3*(Zpunto))+((m3*g*l3*sin((q2+q3)))/K3;
}

```

Programa de control de fuerza

```

////////// TAU //////////
void TauFOR(){ //modo fuerza
XFORCE=(analogRead(SEN)/200.0)*9.8;
kp1=(0.4*15.0)*1.88*exp(-0.008*abs(qd1));
kv1=0.40*kp1;
//PD KP 9,3 KV 4.5,5.0
//TANH KP 25,9 KV 4.5, 5
kp2=9.0;//125
kp3=3.0;//125
kv2=4.50;
kv3=5.0;
X=cos(q1)*(l2*sin(q2)+l3*sin(q2+q3));/+(w1+w2)*sin(q1);
Y=sin(q1)*(l2*sin(q2)+l3*sin(q2+q3));/-(w1+w2)*cos(q1);
Z=l1-l2*cos(q2)-l3*cos(q2+q3);
XFtilde=XFd-XFORCE;
YFtilde=0;//YFd-YFORCE;
ZFtilde=0;//ZFd-ZFORCE;
//////////DERIVADAS//////////
Xpunto=(X-Xanterior)/0.05;
Ypunto=(Y-Yanterior)/0.05;
Zpunto=(Z-Zanterior)/0.05;
//////////
K2=1.7;
K3=2.0;
//////////Componentes del jacobiano//////////
X1=sin(q1)*(sin(q2)*l2+sin((q2+q3))*l3);/+cos(q1)*(w1+w2);
Y1=cos(q1)*(sin(q2)*l2+sin((q2+q3))*l3);/+sin(q1)*(w1+w2);
Z1=0;
X2=cos(q1)*(cos(q2)*l2+cos(q2+q3)*l3);

```

```

Y2=sin(q1)*(cos(q2)*l2+cos(q2+q3)*l3);
Z2=sin(q2)*l2+sin(q2+q3)*l3;
X3=cos(q1)*cos(q2+q3)*l3;
Y3=sin(q1)*cos(q2+q3)*l3;
Z3=sin(q2+q3)*l3;
//////////////////////0+(kp1*tanh(q1tilde))-(kv1*tanh(q1punto)); //
TAU1= X1*(kp1*(XFtilde)-kv1*(Xpunto))+Y1*(kp1*(YFtilde)-kv1*(Ypunto))+Z1*(kp1*(ZFtilde)-kv1*(Zpunto));
TAU2= X2*(kp2*(XFtilde)-kv2*(Xpunto))+Y2*(kp2*(YFtilde)-kv2*(Ypunto))+Z2*(kp2*(ZFtilde)-kv2*(Zpunto))+((m2*g*l2*sin(q2))+(m3*g*l2*sin(q2))+(m3*g*l3*sin(q2+q3)))/K2;
TAU3= X3*(kp3*(XFtilde)-kv3*(Xpunto))+Y3*(kp3*(YFtilde)-kv3*(Ypunto))+Z3*(kp3*(ZFtilde)-kv3*(Zpunto))+((m3*g*l3*sin((q2+q3)))/K3;
//////////////////////
//ADVERTENCIA: CON UN CONTROL SATURADO, EL ROBOT SE TORNA INCONTROLABLE (SE PONE A DAR PUÑETAS A LO LOCO)
//TAU1= 0+(kp1*tanh(q1tilde))-(kv1*tanh(q1punto)); //X1*(kp1*(XFtilde)-kv1*(Xpunto))+Y1*(kp1*(YFtilde)-kv1*(Ypunto))+Z1*(kp1*(ZFtilde)-kv1*(Zpunto));
//TAU2= X2*(kp2*tanh(XFtilde)-kv2*tanh(Xpunto))+Y2*(kp2*tanh(YFtilde)-kv2*tanh(Ypunto))+Z2*(kp2*tanh(ZFtilde)-kv2*tanh(Zpunto))+((m2*g*l2*sin(q2))+(m3*g*l2*sin(q2))+(m3*g*l3*sin(q2+q3)))/K2;
//TAU3= X3*(kp3*tanh(XFtilde)-kv3*tanh(Xpunto))+Y3*(kp3*tanh(YFtilde)-kv3*tanh(Ypunto))+Z3*(kp3*tanh(ZFtilde)-kv3*tanh(Zpunto))+((m3*g*l3*sin((q2+q3)))/K3;
//
//TAU1= 0+(kp1*tanh(q1tilde))-(kv1*tanh(q1punto)); //X1*(kp1*(XFtilde)-kv1*(Xpunto))+Y1*(kp1*(YFtilde)-kv1*(Ypunto))+Z1*(kp1*(ZFtilde)-kv1*(Zpunto));
//TAU2= X2*(kp2*atan(XFtilde)-kv2*atan(Xpunto))+Y2*(kp2*atan(YFtilde)-kv2*atan(Ypunto))+Z2*(kp2*atan(ZFtilde)-kv2*atan(Zpunto))+((m2*g*l2*sin(q2))+(m3*g*l2*sin(q2))+(m3*g*l3*sin(q2+q3)))/K2;
//TAU3= X3*(kp3*atan(XFtilde)-kv3*atan(Xpunto))+Y3*(kp3*atan(YFtilde)-kv3*atan(Ypunto))+Z3*(kp3*atan(ZFtilde)-kv3*atan(Zpunto))+((m3*g*l3*sin((q2+q3)))/K3;
}

```

Programa de par gravitacional

```

void TauG(){ //modo ANGULAR
X=cos(q1)*(l2*sin(q2)+l3*sin(q2+q3));
Y=sin(q1)*(l2*sin(q2)+l3*sin(q2+q3));
Z=l1-l2*cos(q2)-l3*cos(q2+q3);
K2=1.7;
K3=2.0;
TAU1=0;
TAU2=((m2*g*l2*sin(q2))+(m3*g*l2*sin(q2))+(m3*g*l3*sin((q2+q3))))/K2;
TAU3=(m3*g*l3*sin((q2+q3)))/K3;
}

```

Programa de control de impedancia

```

////////////////////// TAU ////////////////////////
void TauLMP(){ //modo cartesiano
XFORCE=(analogRead(SEN)/200.0)*9.8;
BETA();
kp1=40;//10F2
kp2=100;//100F6
kp3=15;//20F4
kv1=4;//F
kv2=10;
kv3=2;
kp11=2.5;
kp21=6;
kp31=4;
kv11=1;
kv21=3.5;
kv31=2;
X=cos(q1)*(l2*sin(q2)+l3*sin(q2+q3));/(w1+w2)*sin(q1);
Y=sin(q1)*(l2*sin(q2)+l3*sin(q2+q3));/-(w1+w2)*cos(q1);
Z=l1-l2*cos(q2)-l3*cos(q2+q3);
Xtilde=Xd-X;
Ytilde=Yd-Y;
Ztilde=Zd-Z;
XFtilde=XFd-XFORCE;
YFtilde=0;//YFd-YFORCE;
ZFtilde=0;//ZFd-ZFORCE;

```

```

//////////velocidad//////////
Xpunto=(X-Xanterior)/0.05;
Ypunto=(Y-Yanterior)/0.05;
Zpunto=(Z-Zanterior)/0.05;
//////////DERIVADA//////////
K2=1.7;
K3=2.0;
//////////Componentes del jacobiano//////////
X1=-sin(q1)*(sin(q2)*l2+sin((q2+q3)*l3));/+cos(q1)*(w1+w2);
Y1=cos(q1)*(sin(q2)*l2+sin((q2+q3)*l3));/+sin(q1)*(w1+w2);
Z1=0;
X2=cos(q1)*(cos(q2)*l2+cos(q2+q3)*l3);
Y2=sin(q1)*(cos(q2)*l2+cos(q2+q3)*l3);
Z2=sin(q2)*l2+sin(q2+q3)*l3;
X3=cos(q1)*cos(q2+q3)*l3;
Y3=sin(q1)*cos(q2+q3)*l3;
Z3=sin(q2+q3)*l3;
//////////
TAU1=
X1*((kp1*(Xtilde)-kv1*(Xpunto))+(kp11*(XFtilde)-kv11*(Xpunto)))+Y1*((kp1*(Ytilde)-kv1*(Ypunto))+kp11*(YFtilde)-
kv11*(Ypunto))+Z1*((kp1*(Ztilde)-kv1*(Zpunto))+(kp11*(ZFtilde)-kv11*(Zpunto)));
TAU2=
X2*((kp2*(Xtilde)-kv2*(Xpunto))+(kp21*(XFtilde)-kv21*(Xpunto)))+Y2*((kp2*(Ytilde)-kv2*(Ypunto))+kp21*(YFtilde)-
kv21*(Ypunto))+Z2*((kp2*(Ztilde)-kv2*(Zpunto))+(kp21*(ZFtilde)-
kv21*(Zpunto)))+(m2*g*lc2*sin(q2))+(m3*g*l2*sin(q2))+(m3*g*lc3*sin(q2+q3))/K2;
TAU3=
X3*((kp3*(Xtilde)-kv3*(Xpunto))+(kp31*(XFtilde)-kv31*(Xpunto)))+Y3*((kp3*(Ytilde)-kv3*(Ypunto))+kp31*(YFtilde)-
kv31*(Ypunto))+Z3*((kp3*(Ztilde)-kv3*(Zpunto))+(kp31*(ZFtilde)-kv31*(Zpunto)))+(m3*g*lc3*sin((q2+q3))/K3;
//////////
//TAU1= X1*(kp1*(Xtilde)-kv1*(Xpunto))+Y1*(kp1*(Ytilde)-kv1*(Ypunto))+Z1*(kp1*(Ztilde)-kv1*(Zpunto));
//TAU2=
X2*(kp2*(Xtilde)-kv2*(Xpunto))+Y2*(kp2*(Ytilde)-kv2*(Ypunto))+Z2*(kp2*(Ztilde)-
kv2*(Zpunto)))+(m2*g*lc2*sin(q2))+(m3*g*l2*sin(q2))+(m3*g*lc3*sin(q2+q3))/K2;
//TAU3= X3*(kp3*(Xtilde)-kv3*(Xpunto))+Y3*(kp3*(Ytilde)-kv3*(Ypunto))+Z3*(kp3*(Ztilde)-kv3*(Zpunto)))+(m3*g*lc3*sin((q2+q3))/K3;
}

```

Control de trayectoria

```

// if(TA<3)
if(TA>(Kl-2)){TA=0;TO=0;}
qd1=DROLL[TA][1];
qd2=DROLL[TA][2];
qd3=DROLL[TA][3];
//
// Xd=DROLL[TA][1];
// Yd=DROLL[TA][2];
// Zd=DROLL[TA][3];
//
}

```

APÉNDICE E: PONENCIAS EN CONGRESOS Y ARTÍCULOS.

Investigación y Desarrollo en Robótica y Computación

Implementación de algoritmos de control en un péndulo de transmisión directa

O. Tepal Nieto, O. Félix Beltrán, F. Reyes Cortés, J. Cid Monjaraz

Facultad de Ciencias de la Electrónica, Benemérita Universidad Autónoma de Puebla, Av. San Claudio y 18 Sur, C.U., Col. Jardines de San Manuel, Puebla, Pue., C.P.72570 México
(oswaldo.tepaln@alumno.buap.mx).

Resumen— En el presente trabajo se presenta la implementación de tres algoritmos de control en un péndulo de transmisión directa poniendo en uso un sistema de control de bajo costo compuesto de una tarjeta de adquisición de datos y control de la instrumentación de su electrónica, para el control de un servo de transmisión directa. Los algoritmos de control implementados son el PD, atan y tanh, con los cuales se obtienen resultados con $\tilde{q} = 1.46, 0.44, 0.28$, respectivamente.

Palabras Clave—Control de fuerza; Péndulo; Robot de Transmisión Directa; Arduino.

I. INTRODUCCIÓN

La robótica es un área de investigación y desarrollo de sistemas mecánicos conocidos como robots [1]. En particular, un robot de transmisión directa se define como un brazo mecánico donde todas sus articulaciones son manejadas con servos de transmisión directa [2]; a su vez, la transmisión directa consiste en que el rotor del motor esté directamente acoplado al eslabón sin necesitar de sistemas de transmisión mecánicos, como engranes [2].

Un ROTRADI (Robot de transmisión directa) es una plataforma experimental que se mueve en su espacio tridimensional equipado con tarjetas electrónicas y programado para realizar aplicaciones de automatización. Estas plataformas poseen características que pueden competir en desempeño y versatilidad con las mejores marcas mundiales de robótica. No obstante su principal característica es el hecho de tener arquitectura abierta, esto significa que se puede evaluar experimentalmente cualquier estrategia de control, en contraste con los robots industriales con arquitectura cerrada que no permiten programar nuevos esquemas de control y quedan confinados a lo que permita realizar su sistema operativo [3].

Desde la creación de estas plataformas hasta la fecha, han sido implementadas diversas tarjetas de adquisición de datos, como lo son las tarjetas Tag1 [4] y la serie de tarjetas MFIO [5]. Debido a la complejidad de estos sistemas, resulta complicada la implementación de nuevos algoritmos de control

para aquellos que no están familiarizados con el sistema. Por lo que, como parte de su actualización, se propone el uso de una tarjeta Arduino, pues en contraste con las tarjetas previamente implementadas, esta tarjeta es programable en un ambiente amigable, de fácil entendimiento, pero sobre todo de fácil acceso. Tomando esto en cuenta se inicia el proceso de implementación para el sistema más simple disponible, en este caso un ROTRADI de un grado de libertad, esto es, un péndulo de transmisión directa.

Específicamente, para esta investigación se escogió la tarjeta Arduino Mega 2560, ya que esta cuenta con compatibilidad con MATLAB. Arduino Mega es una placa microcontrolador basada ATmega1280. Tiene 54 entradas/salidas digitales (de las cuales 14 proporcionan salida PWM), 16 entradas digitales, 4 UARTS (puertos serie por hardware), un cristal oscilador de 16MHz, conexión USB y conector ICSP. El ATmega1280 tiene 128KB de memoria flash para almacenar código (4KB son usados para el arranque del sistema (bootloader), 8 KB de memoria SRAM y 4KB de EEPROM [6].

II. CARACTERÍSTICAS DE LA PLANTA

La plataforma de prueba consta principalmente de tres elementos:

- a) El robot péndulo, en donde se contempla el sistema mecánico a controlar y la interfaz electrónica que permitirá el control del robot.
- b) El controlador, el algoritmo de control que gobernará el comportamiento del robot.
- c) El sensor, el encoder que permitirá conocer el posicionamiento del robot.

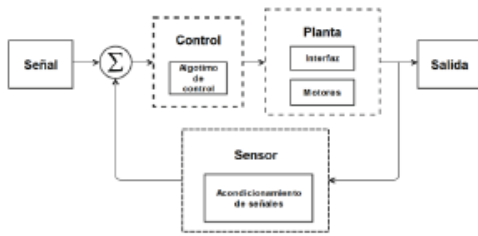


Figura 1 Diagrama a bloques del sistema de control.

Entre las características del robot se tiene que el actuador es un motor de transmisión directa DM1B-015G, cuyo par máximo es de 15 Nm y su encoder tiene una capacidad máxima de 2621440 pulsos por vuelta, lo que nos da una resolución de 0.00013 grados por pulso. Este motor es controlado por un driver DrvG3 el cual permite la programación de diversos parámetros de funcionamiento del motor dependiendo de la modalidad que se requiera, tales como control de posición, velocidad o torque.

Para la implementación abierta de algoritmos de control, se configura la planta en modo torque y adicionalmente se programa el driver para que el encoder funcione en modo diferencial con señales A-B en cuadratura desfasadas 90 grados, como se muestra en la Figura 2.

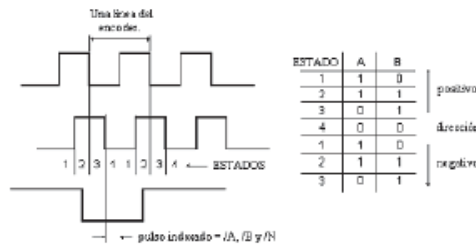


Figura 2 Señales del encoder de cuadratura A-B.

De igual forma y para propósitos de pruebas y ajustes, se configura el sistema para reducir la resolución del encoder a 10260 pulsos por vuelta que nos da una resolución aproximada de 0.035 grados por pulso.

III. CINEMÁTICA

En esta sección se describe el modelo cinemático del robot de 1 grado de libertad (Péndulo) tomando como referencia la Figura 3. Para esto se usa el método de Euler-Lagrange, en donde se da por hecho que el origen del eje x está en el centro del rotor [5].

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_c \sin(q) \\ -l_c \cos(q) \end{bmatrix}, \quad (1)$$

donde l_c es la distancia al centro de masa y q es el ángulo.

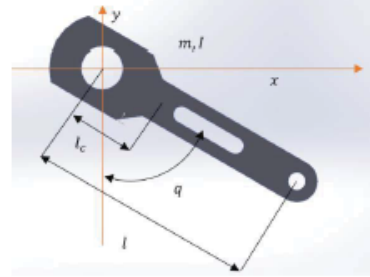


Figura 3 Esquema referencial del péndulo.

El modelo dinámico de un péndulo proporciona una descripción completa entre el par aplicado al servomotor y el movimiento de la estructura mecánica [3]. El lagrangiano está dado por:

$$\mathcal{L}(q, \dot{q}) = K(q, \dot{q}) - U(q), \quad (2)$$

$$\frac{1}{2} [ml_c^2 + I] \dot{q}^2 - mgl_c [1 - \cos(q)], \quad (3)$$

donde \dot{q} es la velocidad angular, m es la masa del eslabón, l es el momento de inercia del eslabón y g es la aceleración de la gravedad.

Las ecuaciones de movimiento de Euler-Lagrange para el caso particular del péndulo adquieren la forma de una ecuación escalar con la siguiente estructura [5]:

$$\tau = \frac{d}{dt} \left[\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \right] - \left[\frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} \right] + f_f(f_e, \dot{q}) \quad (4)$$

donde τ es el par aplicado y f_f es el fenómeno de fricción modelado por la ecuación siguiente:

$$f_f(f_e, \dot{q}) = b\dot{q} + f_c \text{signo}(\dot{q}) + f_e [1 - |\text{signo}(\dot{q})|], \quad (5)$$

donde f_c es la fricción de Coulomb, f_e es la fricción estática, b es el coeficiente de fricción viscosa y la función $\text{signo}(\dot{q})$ está definida como:

$$\text{signo}(\dot{q}) = \begin{cases} 1 & \text{si } \dot{q} > 0, \\ -1 & \text{si } \dot{q} < 0. \end{cases} \quad (6)$$

Por tanto el modelo dinámico de un robot péndulo incluyendo el fenómeno de fricción está dado por:

$$\tau = [ml_c^2 + I] \ddot{q} - mgl_c \text{sen}(q) + b\dot{q} + f_c \text{signo}(\dot{q}) + f_e [1 - |\text{signo}(\dot{q})|], \quad (7)$$

donde \ddot{q} es la aceleración angular.

IV. INTERFAZ ELECTRÓNICA

Para el control del sistema se propone el uso de una tarjeta Arduino Mega, destacándose por poseer una gran cantidad de librerías y soporte, además de su bajo costo y adecuadas prestaciones para los propósitos de cálculo y control. Para el control de la plataforma se precisa de la aplicación de voltaje, el cual debe estar en un rango de $\pm 10V$ y cuyo valor va

a definir el par aplicado. Tomando esto en cuenta, la forma más apropiada para controlar el motor es a través de un DAC. Para esto se diseña un DAC incremental de 10 bits, tanto para valores de voltaje negativos como positivos como se muestra en las Figuras 4 y 5.

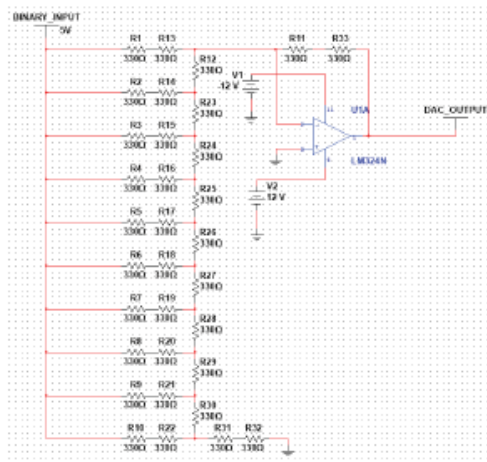


Figura 4 Circuito del DAC utilizado para el control del voltaje.

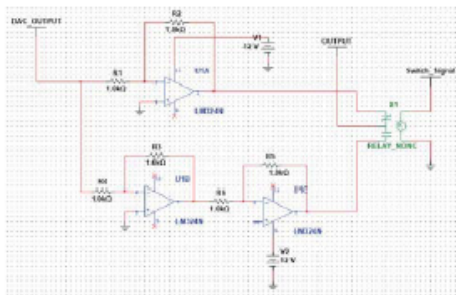


Figura 5 Circuito del DAC utilizado para el control del voltaje.

En la Figura 4 se muestra el típico circuito de un DAC incremental implementado con opamps y resistencias; las entradas al DAC provienen de las salidas digitales del Arduino, que en conjunto con el circuito devuelven un nivel de voltaje negativo, según el valor binario de 10 bits enviado por el microcontrolador. Tomando en cuenta que se requieren valores de voltaje negativos y positivos, se implementa un segundo circuito mostrado en la Figura 4, con el cual se invierte el voltaje y adicionalmente se da la posibilidad de controlar un onceavo bit dando 1023 valores de voltaje positivos y 1023 valores de voltaje negativos.

Para el control del sistema se diseñó un programa descrito en el diagrama de flujo de la Figura 6.

Investigación y Desarrollo en Robótica y Computación

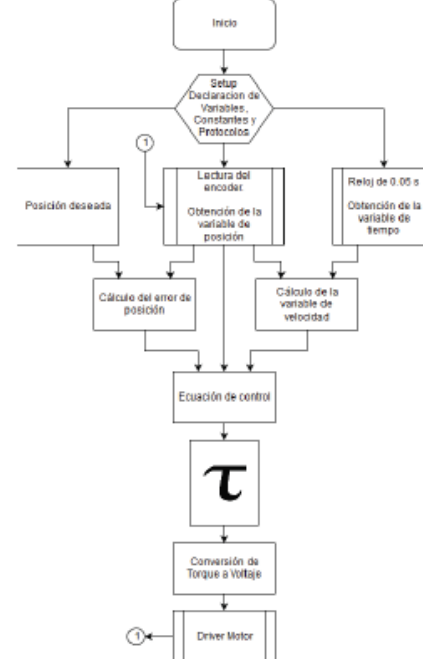


Figura 6 Diagrama de flujo del programa de control.

La función del programa se resume en la obtención de la posición actual del robot, a partir de la cual se obtienen parámetros como velocidad angular y el error de posición entre otros, con base en estos valores se genera un valor de par que es convertido a un equivalente en voltaje, para el caso específico del motor utilizado, el valor de $\pm 10V$ corresponden a un par aplicado de $\pm 15Nm$.

Hay que resaltar que la principal forma de obtener el posicionamiento es a través de las rutinas de interrupción, las cuales son recursos o mecanismos del microcontrolador para responder a eventos, permitiendo suspender temporalmente el programa principal para ejecutar una subrutina de servicio de interrupción (ISR, por sus siglas en inglés) [6]. La tarjeta Arduino Mega tiene capacidad para ejecutar seis rutinas de interrupción a la vez y cada encoder requiere de dos rutinas de interrupción para realizar una correcta lectura de la posición; una vez terminada dicha subrutina se reanuda la ejecución del programa principal.

V. ALGORITMOS DE CONTROL

Dada la estructuración del programa de control, resulta sencillo cambiar el algoritmo de control, pues todo esto se implementa en un simple bloque donde se hacen los cálculos necesarios para la obtención del torque necesario para mover el eslabón a la posición deseada.

Para este trabajo se consideran 3 algoritmos de control, los cuales se describen a continuación.

a) *Controlador PD más compensación de gravedad.*

Para propósitos de prueba y referencia, se implementa en primera instancia el controlador PD más compensación de gravedad, el cual está dado por la expresión [7] [8] [9]:

$$\tau = K_p \ddot{q} - K_v \dot{q} + g(q). \quad (8)$$

El controlador PD es ampliamente conocido y simple de implementar, los resultados de las pruebas con el controlador se pueden tomar como referencia para la comparación de desempeño de otros controladores y a su vez como un punto de partida para sistemas experimentales como el del sistema propuesto.

b) *Controlador atan-atan más compensación de gravedad.*

El segundo controlador propuesto se describe por [7] [8]:

$$\tau = K_p \operatorname{atan}(\ddot{q}) - K_v \operatorname{atan}(\dot{q}) + g(q). \quad (9)$$

c) *Controlador tanh-tanh más compensación de gravedad.*

Finalmente, el tercer controlador propuesto está dado por [7] [8]:

$$\tau = K_p \tanh(\ddot{q}) - K_v \tanh(\dot{q}) + g(q). \quad (10)$$

En los tres algoritmos de control, K_p es la constante de proporcionalidad, K_v es la constante derivativa y \ddot{q} es el error de posición.

VI RESULTADOS

En esta sección se presentan los resultados obtenidos, las gráficas del par aplicado y de su posición angular con respecto al tiempo de cada uno de los controladores utilizados en la pruebas. Para cada prueba se seleccionó un valor en específico como ángulo deseado, en este caso ese valor corresponde a $q_d = -90^\circ$ y el error de posición se calcula como:

$$\ddot{q} = q_d - q. \quad (11)$$

para las condiciones de estas pruebas, se realizaron 30 repeticiones.

A. *Controlador PD*

Para este controlador los resultados se pueden apreciar en la Figura 7, en donde el error de posición promedio fue de $\ddot{q} = 1.46$ grados.

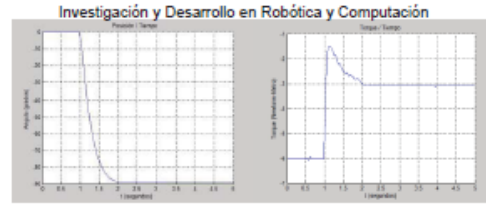


Figura 7 Gráfica de posición y torque con respecto al tiempo para el control PD.

B. *Controlador atan-atan*

Para este controlador los resultados se pueden apreciar en la Figura 8, en donde el error de posición promedio fue de $\ddot{q} = 0.44$ grados.

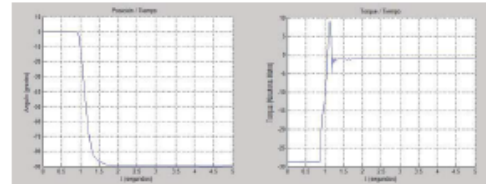


Figura 8 Gráfica de posición y torque con respecto al tiempo para el control atan-atan.

C. *Controlador tanh-tanh*

Para este controlador los resultados se pueden apreciar en la Figura 9, en donde el error de posición promedio fue de $\ddot{q} = 0.28$ grados.

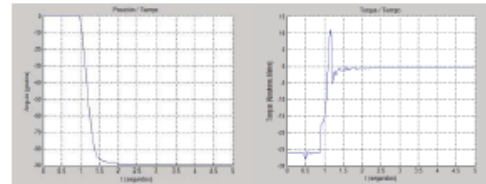


Figura 9 Gráfica de posición y torque con respecto al tiempo para el control tanh-tanh.

A continuación se muestra una tabla comparando el error de posición obtenida en cada uno de los controladores.

TABLA I. TABLA COMPARATIVA DEL PROMEDIO DEL ERROR.

Controlador	\ddot{q} (grados)
PD	1.46
tanh-tanh	0.28
atan-atan	0.44

VII. CONCLUSIONES

Para este trabajo, el desempeño de los controladores a través del sistema propuesto fue eficaz con errores de

posicionamiento aceptables, cabe mencionar que este error puede ser reducido a través de una buena sintonización de las constantes k_p y k_v . Adicionalmente, el microcontrolador utilizado y el hardware diseñado para la planta, demostró ser versátil y completamente funcional para el procesamiento de información, por lo que se puede concluir que el Arduino es un sistema apto para la implementación de algoritmos experimentales de control y por tanto candidato para la implementación de la plataforma ROTRADI de tres grados de libertad en el cual se está trabajando.

La electrónica que se diseñada para el robot de 1 grado de libertad, tiene la cualidad de ser escalable, lo cual significa que se puede copiar e implementar para cualquier número de servomotores.

[1] L. Scavicco y B. Siciliano, *Modeling and Control of Robot Manipulador*, McGraw-Hill 2011.

[2] C. Chávez Olivares, F. Reyes Cortés, E. González Galván, M. Mendoza Gutierrez, I. Bonilla-Gutierrez. "Experimental evaluation of parameter identification schemes on an anthropomorphic direct drive robot". *International Journal of Advanced Robotic Systems*, pp. 9. ISO 690, 2012.

[3] F. Reyes, *Robótica, Control de Robots Manipuladores*, Alfa-omega, 1ª ed. 2011.

[4] E. Leon Bonilla, "Tarjeta PCI para controlar un Robot con tres grados de libertad", Tesis de Maestría en Ciencias de la electrónica opción Automatización, FCE-BUAP, 2007.

[5] A. Palma Asunción, "Control visual de robots manipuladores", Tesis de Maestría en Ciencias de la electrónica opción Automatización, FCE-BUAP, 2016.

[6] C. RUIYAN. "Principle and application of Arduino. *Electronic Design Engineering*", vol. 20, no. 16, pp. 155-157, 2012.

[7] J. G. Cebada Reyes, J. Tuxpan Meneses, P. Sánchez Sánchez, F. Reye Cortés, "Evaluación de 4 Estructuras de Control Mediante un Simulador Basado en un Robot de 2 Grados de Libertad", 8º Congreso Nacional de Mecatrónica, 2009.

[8] G. PALLI, C. MELCHIORRI "Robust control of robots with variable joint stiffness". En *Advanced Robotics. ICAR 2009, International Conference on IEEE*, pp. 1-6, 2009.

[9] O. R. Arrovo, F. R. Cortes, M. A. D. V. Trevino, S. V. Limon. "Variable Control Gains for PD Controller for Robot Manipulators". *Mechatronics, Electronics and Automotive Engineering (ICMEAE)*, pp. 187-192, 2015.



EL INSTITUTO TECNOLÓGICO DE LA PAZ

OTORGA EL PRESENTE

RECONOCIMIENTO

A

OSWALDO TEPAL NIETO, OLGA GUADALUPE FÉLIX BELTRÁN, FERNANDO REYES CORTÉS Y JAIME JULIÁN CID MONJARAZ

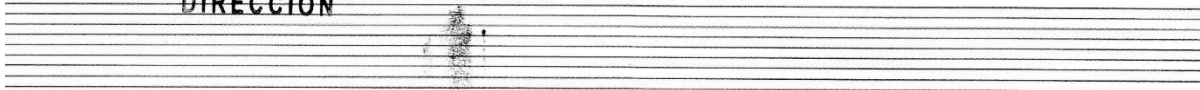
POR SU ARTÍCULO: IMPLEMENTACIÓN DE ALGORITMOS DE CONTROL EN UN PÉNDULO DE TRANSMISIÓN DIRECTA PRESENTADO EN EL TERCER CONGRESO INTERNACIONAL DE ROBÓTICA Y COMPUTACIÓN CELEBRADO DEL 2 AL 4 DE MAYO DEL AÑO EN CURSO.

LA PAZ, B.C.S., 4 DE MAYO DE 2016.



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO
DE LA PAZ
DIRECCIÓN

ING. JESÚS DAVID ESTRADA RUIZ
DIRECTOR



BIBLIOGRAFÍA

- [1] L. Scavicco y B. Siciliano, *Modeling and Control of Robot Manipulador*, McGraw-Hill (2011).
- [2] H. Asada y T. Youcef, *Direc-Drive Robots: Theory and Practice*, The MIT Press, pp. 239 (1987).
- [3] R. Iñigo y H. Asada, *Robots industriales manipuladores*, Alfaomega, España, pp. 23 (2004).
- [4] F. Reyes y R. Kelly, Experimental Evaluation of Identification Schemes on a Direc-Drive Robot, *Robótica*, Vol. 15, United Kingdom, pp. 563 (1997).
- [5] D. A. Wells, *Dinámica de Lagrange*, McGraw-Hill, p. 62 (1972).
- [6] F. Reyes, *Robótica, Control de Robots Manipuladores*, Alfa-omega, 1^{ra} Edición (2011).
- [7] R. Kelly y R. Santibañez, *Control de movimiento de robots manipuladores*, Prentice Hall, México (2003).
- [8] I. Bonilla, F. Reyes y M. Mendoza, Dynamic-compensation approach to impedance control of robots manipulators, *J. Intell Robot Syst*, 63:51-73 (2011).
- [9] N. Hogan, Impedance control: an approach to manipulation: part I-theory, part II-Implementacion and part III-applications, *ASME J Dyn, Syst. Meas*, Vol. 107, pp. 1-24 (1985).
- [10] R. Zotovic y A. Valera, Reajuste de los parámetros de la impedancia mecánica para la mejora de la disipación de la energía durante el impacto, Vol. 6, Num. 4, pp. 21-31 (2009).
- [11] M. Spong y M. Vidyasagar, *Robot Dynamics y Control*, John Wiley and sons, Inc. United Estate of America (1989).
- [12] A. Barrientos y L. Peñin, *Fundamentos de Robótica*, McGraw-Hill, 2^{da} Edición (2009).
- [13] K. C. Jain y L. N. Aggarwal, *Robotics-Principles Practice*, Khanna Publishers, Delhi, 3^{ra} Edición (2002).
- [14] J. Norberto Pires y T. Godinho, *Control difuso de fuerza en robótica industrial*, Universidad de Coimbra, Portugal, Departamento de Ingeniería Mecánica, Vol. 11, No. 41, pp. 55-57 (2008).

- [15] W. Lu y Q. H. Meng, *Impedance control with adaptation for robotic manipulations*, IEEE Trans. Robot. Autom., Vol. 20, No. 3 (2004).
- [16] E. González Tlaxco, *Diseño mecánico y construcción de un robot antropomorfo de tres grados de libertad*, Tesis de Maestría Benemérita Universidad Autónoma de Puebla (2010).
- [17] O. Khatib, *A unified approach for motion and force control of robot manipulators: the operational space formulation*, IEEE J Robot (1987).
- [18] M. Raibert y J. Craig, *Hybrid position/force control of manipuladores*, ASME, J Dyn. Syst. Meas, pp. 126-133 (1981).
- [19] J. García, J. Rodríguez y J. Vidal, *Aprenda matlab 7.0 como si estuviera en primero*, Universidad Politécnica de Madrid (2005).
- [20] S. Casillas, *Tutorial de Matlab* (2008).
- [21] <http://Arduino.cc/en/Main/HomePage> (2015).
- [22] <http://Arduino.cc/en/Main/ArduinoBoardMega2560> (2015).
- [23] Reyes, Cid, Vargas, *Mecatrónica: Control y Automatización*, Ed. AlfaOmega (2014).
- [24] Dynaserv G3 Drive User Guide, April 1 2004, p/n 88-025078-01A.
- [25] Fernando Reyes Cortez / Jaime Cid Monjaraz, *Arduino: aplicaciones en robótica, mecatrónica e ingenierías*, Ed. Marcombo (2015).
- [26] J. G. Cebada Reyes, J. Tuxpan Meneses, P. Sánchez Sánchez, F. Reye Cortés, “Evaluación de 4 Estructuras de Control Mediante un Simulador Basado en un Robot de 2 Grados de Libertad”, 8^o Congreso Nacional de Mecatrónica, 2009.
- [27] G. PALLI, C. MELCHIORRI. “Robust control of robots with variable joint stiffness”. En Advanced Robotics. ICAR 2009, International Conference on. IEEE, pp. 1-6, 2009.
- [28] O. R. Arroyo, F. R. Cortes, M. A. D. V. Trevino, S. V. Limon. “Variable Control Gains for PD Controller for Robot Manipulators”. Mechatronics, Electronics and Automotive Engineering (ICMEAE), pp. 187-192, 2015.
- [29] Takegaki, M., Arimoto, S.: A new feedback method for dynamic control of manipulators. ASME, J. Dyn. Syst. Meas. Control **102**, 119–125 (1981).
- [30] Sciavicco, L., Siciliano, B.: *Modeling and Control of Robot Manipulators*. McGraw-Hill, New York (1996).

