

# **UNIVERSIDAD POLITÉCNICA SALESIANA**

**CARRERA:**

**INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:  
INGENIERA ELECTRÓNICA E INGENIERO ELECTRÓNICO**

**TEMA:**

**CONTROL ROBUSTO DE UN MOTOR AC MEDIANTE WNCS (WIRELESS  
NETWORK CONTROL SYSTEM)**

**AUTORES:**

**SILVIA LORENA CHUQUÍN BALSECA  
EDWIN PAÚL PACHACAMA DURÁN**

**TUTOR:**

**CARLOS GERMÁN PILLAJO ANGOS**

**Quito, julio del 2019**

## CESIÓN DE DERECHOS DE AUTOR

Nosotros, Silvia Lorena Chuquín Balseca con documento de identificación N° 172514748-0 y Edwin Paúl Pachacama Durán con documento de identificación N° 172053459-1, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: “CONTROL ROBUSTO DE UN MOTOR AC MEDIANTE WNCS (WIRELESS NETWORK CONTROL SYSTEM).”, mismo que ha sido desarrollado para optar por el título de: Ingeniera Electrónica e Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

.....  
Silvia Lorena Chuquín Balseca

Cédula: 172514748-0

.....  
Edwin Paúl Pachacama Durán

Cédula: 172053459-1

Quito, julio 2019

## **DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR**

Yo declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación, “CONTROL ROBUSTO DE UN MOTOR AC MEDIANTE WNCS (WIRELESS NETWORK CONTROL SYSTEM).”, realizado por Silvia Lorena Chuquín Balseca y Edwin Paúl Pachacama Durán, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerado como trabajo final de titulación.

Quito, julio del 2019



Msc. Carlos Germán Pillajo Angos  
C.I: 1709255119

## **DEDICATORIA**

Este proyecto de titulación le dedico primeramente a Dios, él fue mi fortaleza en todo momento y a pesar de que no lo vemos sé que está presente en todo momento para ayudarnos y velar por nosotros, también va dirigida a mis padres, mis abuelitos y mi hermano, a pesar de los momentos difíciles hicieron todo lo posible para mi estudio, sus esfuerzos no fueron en vanos, aquí estamos siguiendo adelante, por ultimo le dedico a mi novio Diego Zapata más que compañero fue mi ayuda en toda mi carrera universitaria, llegando así a cumplir una meta más de mi vida.

**Silvia**

El presente proyecto de titulación se lo dedico a todas las personas que confiaron en mí, a mis padres por su sacrificio, esfuerzo, por creer en mis capacidades y por darme la posibilidad de forjar una profesión para defender mi futuro y el de mi familia.

A mi amada esposa le doy las gracias por estar ahí en los peores momentos y a mi hija que es mi fuente de motivación e inspiración para poder ser mejor cada día y luchar por un futuro mejor.

A mis compañeros y amigos que me ayudaron y compartieron su conocimiento, alegrías, tristezas y a aquellas personas que me apoyaron y me ayudaron a lograr que mi sueño se haga realidad.

Gracias por todo.

**Paúl**

## **AGRADECIMIENTOS**

Primeramente, agradezco a Dios, por todo, por mi familia, por mi novio, por mis compañeros, por mis profesores y porque a pesar de que mucha gente dijo no lo vas a lograr, no fue así, Dios sabe porque hace las cosas y nunca hay que desanimarse.

Gracias Dios por todas esas bellas personas de mí vida.

**Silvia**

Agradezco a todas las personas que permitieron que esto sea posible, a mis padres, familia y compañeros que con su apoyo me permitieron continuar con esta nueva fase de mi vida.

Les doy las gracias a todos.

**Paúl**

Además, agradecemos a la Universidad Politécnica Salesiana, a nuestro tutor Ing. Carlos Germán Pillajo quien con sus conocimientos, consejos y apoyo nos guió a lo largo de esta etapa, no solo en lo profesional también en la parte humana, fue un apoyo y nos motivó a seguir, para llegar a concluir con el presente proyecto de la mejor manera.

**Los Autores**

## CONTENIDO

CESIÓN DE DERECHOS DE AUTOR.....	ii
DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR.....	iii
DEDICATORIA .....	iv
AGRADECIMIENTOS .....	v
CONTENIDO .....	vi
ÍNDICE DE FIGURAS.....	ix
RESUMEN.....	xi
ABSTRACT.....	xii
INTRODUCCIÓN .....	xiii
CAPÍTULO 1 ANTECEDENTES .....	1
1.1 Tema.....	1
1.2 Planteamiento del problema .....	1
1.3 Justificación del proyecto .....	1
1.4 Objetivos .....	2
1.4.1 Objetivo General.....	2
1.4.2 Objetivos específicos .....	2
1.5 Alcance.....	2
CAPÍTULO 2.....	3
MARCO TEÓRICO.....	3
2.1 Sistemas Ciber – Físicos.....	3
2.2 Sistemas de control en red Inalámbrica WNCs .....	4
2.3 Internet de las cosas (IoT) .....	4
2.4 Controlador de voltaje AC.....	5
2.4.1 Control On/off.....	5
2.4.2 Control de ángulo de fase .....	5
2.5 Herramientas de software .....	6
2.5.1 Simple Object Access Protocol (SOAP).....	6
2.5.2 Lenguaje de Marcación de Hipertexto (HTML).....	7
2.5.3 Protocolo de Transferencia de Hipertexto (HTTP).....	7
2.5.4 Notación de Objetos de JavaScript (JSON).....	8
2.5.5 Pre Procesador de Hipertexto (PHP).....	8
2.5.6 JavaScript.....	8
2.5.7 Base de Datos MySQL .....	9
2.5.8 Arduino IDE .....	9

2.5.9 Ident de Matlab .....	10
2.6 Hardware utilizado en el proyecto.....	10
2.6.1 Arduino mega 2560 .....	10
2.6.2 Módulo Dragona Yun Shield.....	11
2.6.3 Raspberry Pi.....	11
2.7 Motor de Inducción Monofásico Jaula de Ardilla.....	12
2.8 Representación en espacio de estados .....	12
2.9 Control FUZZY .....	13
2.9.1 Funciones de Membresía .....	13
2.9.2 Reglas de inferencia.....	14
2.9.3 Fuzzificación.....	14
2.9.4 Defusificación.....	14
2.9.5 Ventajas y desventajas del control difuso.....	15
2.10 Control Lineal Cuadrático Gaussiano – LQG .....	15
2.10.1 Control LQR .....	16
2.10.2 Filtro de Kalman .....	16
2.10.3 Ventajas y Desventajas del control LQG.....	17
CAPÍTULO 3 .....	18
DISEÑO E IMPLEMENTACIÓN DE LOS ALGORITMOS .....	18
3.1 Implementación de las etapas de control y potencia .....	18
3.1.1 Diseño de la etapa de potencia.....	18
3.1.2 Diseño de la etapa de control.....	19
3.1.3 Programación del ángulo de disparo en el Arduino.....	21
3.2 Diseño del controlador FUZZY .....	22
3.1.1 Diseño de las entradas difusas .....	23
3.1.2 Diseño de las reglas de inferencia.....	25
3.1.4 Diseño de la Salida Difusa.....	26
3.3 Obtención de la Función de Transferencia.....	29
3.4 Diseño del controlador LQG .....	31
CAPÍTULO 4 .....	33
ALMACENAMIENTO DE LOS DATOS EN LA NUBE.....	33
4.1 Diagrama general del Software .....	33
4.2 Comunicación en el Servidor .....	35
4.2.1 Librería NuSOAP - Servidor .....	36
4.2.2 Service.php - Servidor .....	36

4.2.3 Client.php – Servidor.....	39
4.2.4 Server.php – Servidor .....	39
4.2.5 Índex – Servidor.....	40
4.3 Comunicación en el Cliente .....	40
4.3.1 Arduino Mega 2560 – Cliente .....	40
4.4 Creación de una base de datos en MySQL.....	42
CAPÍTULO 5 .....	43
PRUEBAS Y RESULTADOS .....	43
5.1 Funcionamiento del controlador Fuzzy WAN .....	43
5.1.1 Pruebas y resultados, Funcionamiento Fuzzy WAN .....	43
5.2 Funcionamiento del controlador LQG WAN.....	43
5.2.2 Pruebas y resultados, Funcionamiento LQG WAN.....	44
5.3 Comparación de los controladores Fuzzy y LQG WAN.....	44
CONCLUSIONES .....	47
RECOMENDACIONES .....	48
REFERENCIAS .....	49
ANEXOS .....	52
ANEXO A: Configuración del Servidor en la nube.....	52
Obtención de dominio para el servidor y carga de los archivos.....	53
Configuración de la Base de Datos en MySQL .....	56
ANEXO B: Datos de la planta.....	58

## ÍNDICE DE FIGURAS

Figura 2.1 Diagrama de interacción Sistemas Ciber-Físicos .....	3
Figura 2.2 Control on/off de voltaje alterno.....	5
Figura 2.3 Control de fase.....	6
Figura 2.5 Arduino Mega 2560 .....	10
Figura 2.6 Módulo Dragino Yun Shield .....	11
Figura 2.7 Raspberry Pi 3 Model B .....	12
Figura 2.8 Rotor tipo Jaula de Ardilla.....	12
Figura 2.9 Diagrama de bloques en variables de estado .....	13
Figura 2.10 Diagrama de bloques control LQR.....	16
Figura 2.11 Diagrama de bloques filtro Kalman.....	17
Figura 3.1 Esquema del proyecto.....	18
Figura 3.2 Diagrama de potencia .....	19
Figura 3.3 Optoacoplador aislador .....	19
Figura 3.4 Detector de cruce por cero Arduino.....	20
Figura 3.5 Placa Encoder Óptico .....	20
Figura 3.6 Diagrama parte de control .....	21
Figura 3.7 Ángulo de disparo para un semiciclo.....	21
Figura 3.8 Diagrama de bloques control Fuzzy .....	22
Figura 3.9 Toma de datos motor monofásico.....	22
Figura 3.10 Ventana función Fuzzy .....	23
Figura 3.11 Diseño de la Entrada Difusa 1 .....	24
Figura 3.12 Diseño de la Entrada Difusa 2 .....	25
Figura 3.13 Reglas de inferencia.....	26
Figura 3.14 Diseño de la Salida Difusa.....	27
Figura 3.15 Diagrama de flujo del algoritmo FUZZY .....	28
Figura 3.16 Simulación del algoritmo FUZZY .....	29
Figura 3.17 Diagrama de bloques de la planta en lazo abierto .....	29
Figura 3.18 IDENT de Matlab con un conjunto de datos .....	30
Figura 3.19 IDENT de Matlab con dos conjuntos de datos .....	30
Figura 3.20 Estimación de la función de transferencia .....	31
Figura 3.21 Diagrama de flujo controlador LQG.....	32
Figura 4.1 Diagrama general del software .....	33
Figura 4.2 : Direccionamiento Servidor Local (WLAN).....	35
Figura 4.3 : Direccionamiento Servidor en la Nube (WAN) .....	35
Figura 4.4 Diagrama del client.php y service.php .....	36
Figura 4.5 Server_enc .....	37
Figura 4.6 Server_setp .....	38
Figura 4.7 Server_angulo .....	39
Figura 4.8 Función de llamado y respuesta del cliente del servidor .....	39
Figura 4.9 Flujograma de Índice .....	40
Figura 4.10 Cliente SOAP .....	41
Figura 4.11 Lectura de la respuesta POST del cliente .....	41
Figura 5.1 Prueba algoritmo Fuzzy.....	43
Figura 5.2 Prueba algoritmo LQG .....	44
Figura 5.3 Desempeño de los controladores .....	45

Figura 5.4 Error absoluto y máximo algoritmo Fuzzy.....	45
Figura 5.5 Error absoluto y máximo algoritmo LQG .....	46

## RESUMEN

Wireless Network Control System es un sistema que aprovecha el internet de las cosas (IOT) para controlar y monitorear sistemas de control, se busca aprovechar las nuevas tecnologías de comunicación con varios conceptos de control, los cuales fusionados se forman un sistema donde interactúan elementos electrónicos y controladores donde sus datos pasan a través de una red inalámbrica como Wifi. La principal ventaja es la reducción de costo en la implementación y mantenimiento, además la adición de elementos resulta más simple; la principal limitación es el intercambio de datos en tiempo real, si bien existen protocolos que permiten un intercambio rápido de datos, estos vienen a reducir la seguridad de los sistemas. En el presente trabajo, se desarrolló tanto los algoritmos, así como un sistema de control de velocidad de un Motor AC, para el intercambio de datos se tiene una RASPBERRY PI en la cual se utiliza lenguajes como HTML, PHP y JAVASCRIPT para el intercambio y autenticación del flujo de datos. En la planta los componentes están conectados a una tarjeta Arduino Mega 2560 como controlador local, además cuenta con un módulo Dragino inalámbrico para el intercambio de datos a través de Wifi. En cuanto al servidor se tiene todos los algoritmos de control como LQG y FUZZY además de la administración de datos en una base de datos.

## **ABSTRACT**

Wireless Network Control System is a system that takes advantage of the Internet of Things (IOT) face controlling and monitoring control systems, it seeks to take advantage of new communication technologies with various control concepts, which merged into a system where electronic elements interact and controllers where your data passes through a wireless network such as Wi-Fi. The main advantage is the reduction of cost in the implementation and maintenance, in addition the addition of elements is simpler; The main limitation is the exchange of data in real time, although there are protocols that allow a rapid exchange of data, these come to reduce the security of the systems. In the present work, the algorithms were developed, as well as a speed control system of an AC Motor, for the exchange of data there is a RASPBERRY PI in which languages such as HTML, PHP and JAVASCRIPT are used for the exchange and authentication of the data flow. In the plant the components are connected to an Arduino Mega 2560 card as a local controller, in addition it has a wireless Dragino module for the exchange of data through Wifi. As for the server, we have all the control algorithms such as LQG and FUZZY, as well as data management in a database.

## INTRODUCCIÓN

La proliferación de los dispositivos en red, en conjunto con las ventajas de tener sistemas en línea, tanto como en capacidad de procesamiento y almacenamiento, ha hecho que el sector industrial constituya procesos a través de la red dado nacimiento a Network Control System (NCS).

En el presente trabajo se trata de la implementación de un controlador de motor AC con diferentes algoritmos elegibles entre Fuzzy y LQG. Estructurado en 5 capítulos que justifican los objetivos planteados.

El capítulo 1 contiene la metodología con la que se realizó el proyecto. El capítulo 2 se presenta los conceptos y fundamentos teóricos tanto de control de voltaje alterno, como de Sistemas de Control en Red Inalámbrica (WNCS), también tiene conceptos básicos de los controladores Fuzzy y LQG.

En el capítulo 3 se aborda la implementación de los algoritmos, planta y servidor, además de las herramientas necesarias para poder diseñar los algoritmos de control, también se muestra un esquema de conexión básico del sistema y la función de transferencia de la planta. En el capítulo 4 se detalla la configuración del servidor en la nube, haciendo que el sistema pueda ser controlado desde cualquier parte del mundo. En el capítulo 5 se muestra los resultados obtenidos al comparar ambos controladores en estabilidad, y encontrarse el servidor en un servicio de hosting.

# **CAPÍTULO 1**

## **ANTECEDENTES**

En el capítulo 1 presenta todo lo pertinente al tema, el planteamiento del problema a resolver, la justificación, el objetivo general, los objetivos específicos y el alcance del actual proyecto.

### **1.1 Tema**

Control robusto de un motor AC mediante WNCS (Wireless Network Control System).

### **1.2 Planteamiento del problema**

Para las estructuras de control industrial se necesita de una gran cantidad de instrumentación tanto de sensores y actuadores que admiten monitoreo y control del comportamiento de métodos físicos, la evolución tecnológica en instrumentación acude a instalaciones de comunicación mediante medios no guiados. Asimismo, es necesario controladores de alto desempeño en cada estructura o planta lo que aumenta los gastos de implementación de este tipo de sistemas. (Monsalve Posada & Arias Londoño, 2015)

La dificultad que enfrenta este tipo de industrias, son los retardos presentes en la comunidad usualmente inalámbrica, por ende, se soluciona a través de diferentes algoritmos predictivos que ayuda a prever el proceder del proceso manteniendo la permanencia del sistema. (Monsalve Posada & Arias Londoño, 2015)

### **1.3 Justificación del proyecto**

El adelanto de servicios en la nube permite, que el control no esté ligado a la planta, sino que un beneficiario autorizado podrá vincular diferentes procesos al servidor, contribuyendo al avance de nuevas aplicaciones en diferentes campos, además de la que se implementará en el presente trabajo.

La comunicación de terminales inalámbricos implica retardos llamados “latencia”, en sistemas industriales de control este intercambio de información debe avalar fiabilidad, por esta razón se utiliza un algoritmo de control que no presente problemas de deficiencia que existen en la red. (Monsalve Posada & Arias Londoño, 2015)

La computación en la nube o cloud computing en un tipo de comunicación que admite el acceso por petición en la red, a un acumulado de recursos informáticos parametrizables.

El prototipo de computación en la nube suscita medios y es compuesta por características esenciales que permiten tener a un usuario accesibilidad y disponibilidad de los recursos en la red. (Servetto, Pascal, & Lobo Mirassón, 2017)

## **1.4 Objetivos**

### **1.4.1 Objetivo General**

Controlar un motor monofásico AC de 110 VAC mediante una red inalámbrico (WIFI) para cerrar el lazo de control utilizando métodos de control robusto.

### **1.4.2 Objetivos específicos**

- Analizar la base teórica para la implementación de este tipo de controladores en redes inalámbricas.
- Diseñar los controladores Fuzzy y LQG para el control de velocidad de un motor AC.
- Desarrollar los algoritmos de control robusto en un sistema embebido para el control de velocidad de un motor AC a través de una red inalámbrica (WNCS).
- Almacenar en la nube los datos obtenidos de la planta para su interpretación y ejecución en el controlador
- Determinar el desempeño de los controladores Fuzzy y LQG mediante pruebas y comparación de resultados para la determinación del mejor controlador en este tipo de sistemas.

## **1.5 Alcance**

Los favorecidos por el actual trabajo son: investigadores y desarrolladores inmiscuidos en teoría de control, porque con la creación de un controlador de velocidad para motor AC a través de la red, es una contribución a desconocidos conocimientos que aún se puede mejorar.

## CAPÍTULO 2

### MARCO TEÓRICO

El capítulo 2 describe todas las implicaciones teóricas necesarias para la implementación del control de velocidad de un motor AC a través de la red, subsiguientemente se detalla el diseño de controladores Fuzzy y LQG.

#### 2.1 Sistemas Ciber – Físicos

En la actualidad los Sistemas Ciber-Físicos busca mejorar la calidad de vida de las personas, a través del monitoreo constante y también del respectivo control. En el mundo físico donde vivimos se utiliza las herramientas del mundo cibernéticos y estos acoplando conocimientos de control, comunicación y computación se hace realidad lo que en siglos anteriores se creía imposible. (Pillajo Angos, 2014)

Figura 2.1 Diagrama de interacción Sistemas Ciber-Físicos

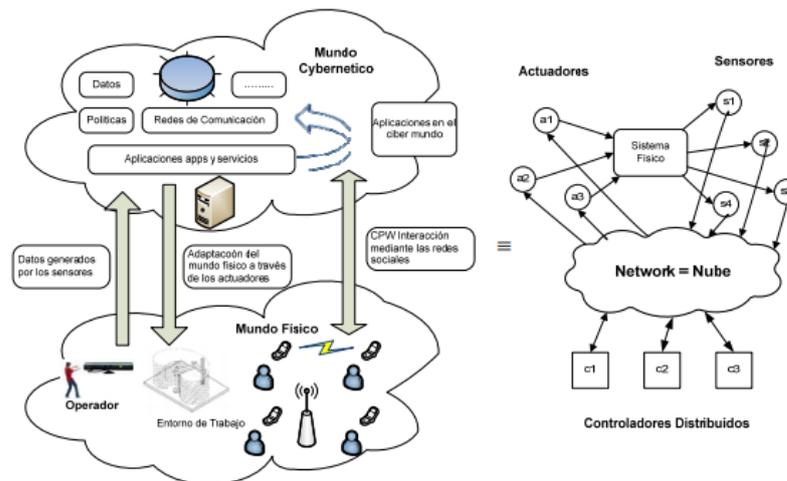


Diagrama gráfico interacción de un Sistema Ciber-Físico. (Pillajo Angos C., 2014)

Los sistemas ciber – físicos son procesos que incorporan la computación en los procesos físicos, prácticamente se tiene ciclos de retroalimentación, los procesos físicos sobresaltan los cálculos y recíprocamente. Podría expresarse que los estudios de los Sistemas Ciber-Físicos (CPS), poseen la capacidad de menguar el cambio del internet de las cosas del siglo XX.(Chandy, 2010)

Estos sistemas incluyen varios beneficios tales como: dispositivos médicos altamente confiables, sistemas recientes de control de tráfico, control de infraestructuras críticas, sistemas de protección y estructuras inteligentes. Con todo esto uno se puede suponer recientes aplicaciones como la generación de energía acoplada a la micro-distribuida

de la red eléctrica, donde la sincronización de tiempo y seguridad son esenciales. Esta rama se ubica muy relacionada a la industria 4.0, la asociación científica está de acuerdo que este tipo de sistemas concretamente los de producción pueden causar la cuarta revolución industrial. (Chandy, 2010) (Beruvides López, Haber Guerra, & Castaño Romero, 2016)

## **2.2 Sistemas de control en red Inalámbrica WNCS**

Los sistemas de control en red (NCS) en la actualidad se están ejecutando a través de redes inalámbricas, debido al desarrollo rápido que ha tenido la tecnología. Cuando recién salió la tecnología llamada también sin cables, era muy insegura nadie la utilizaba, pero en los últimos tiempos esta perspectiva ha cambiado; ahora es confiable y de alta velocidad.

Estos sistemas toman por nombre sistemas de control en red inalámbrica (WNCS), los más simples son los que se aplican en una planta, con un controlador y tiene comunicación punto a punto entre ellos. Entre una de las ventajas más notables es la reducción de costos y un ahorro en el cableado, uno de los desafíos que se enfrenta esta tecnología es sin duda a la fiabilidad y los retardos de comunicación, ya que el canal por donde pasan los datos no es ideal.(Zhang, Wang, & Yu, 2017) (Hernandez Bonilla & Vargas Plazas, 2013)

## **2.3 Internet de las cosas (IoT)**

El internet de las cosas (IoT), consiste en interconectar los diferentes objetos, que se tiene en los hogares con el internet, y no se puede decir que solo se emplea para este fin, en la actualidad este concepto se ha llevado a la industria, a la medicina, etc. A continuación, se describe algunas aplicaciones. (Pillajo & Roberto, 2018)

El potencial del entorno de Internet de las cosas (IoT) para sensores interconectados ha jugado un papel muy importante en la industria, se ha realizado aplicaciones en la salud. Las personas discapacitadas y adultos mayores necesitan de mucho cuidado, en vista de esta necesidad se ha dado paso a la Salud Industrial IO (HealthIIoT) que tiene una infraestructura de vigilancia se salud para así evitar muertes.(Hossain. & Muhammad., 2016)

A partir del avance del Internet de las cosas aparece el Internet industrial de las cosas (IIoT), es una maquina industrial conectada a la nube para el almacenamiento de los diferentes datos que se generen en el proceso industrial, así dando la ventaja de

recuperar los datos si estos se pierden. La 4.0 junto con IIoT puede hacer una gran revolución en los diferentes procesos que se realiza en la industria.(Jayaram, 2016)

## 2.4 Controlador de voltaje AC

Un controlador de voltaje AC son circuitos mediante los cuales se puede regular el voltaje RMS suministrado a una carga de forma controlada, sus aplicaciones varían dependiendo del tipo de controlador que se desea implementar, sus aplicaciones más comunes son el control de luminosidad de luces y el control de velocidad de motores AC. (Rashid, 2015)

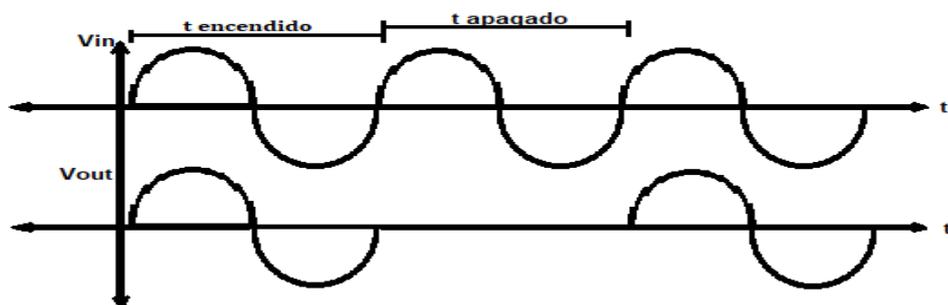
Para el control de velocidad de motores AC se tienen varias estrategias de control, a continuación, se muestran dos formas comunes, pero en general existen más tipos de control:

- Control On/off
- Control de ángulo de fase

### 2.4.1 Control On/off

En este principio de control se conecta durante unos cuantos ciclos a la carga y luego se desconecta por otro número de ciclos, estos se denominan tiempos activos y pasivos y definen el tiempo o número de ciclos que se activara el circuito o se apagara, este tipo de control es utilizado en aplicaciones que tienen una alta inercia mecánica y una constante alta de tiempo térmica como los calefactores industriales.(Rashid, 2015)

Figura 2.2 Control on/off de voltaje alterno



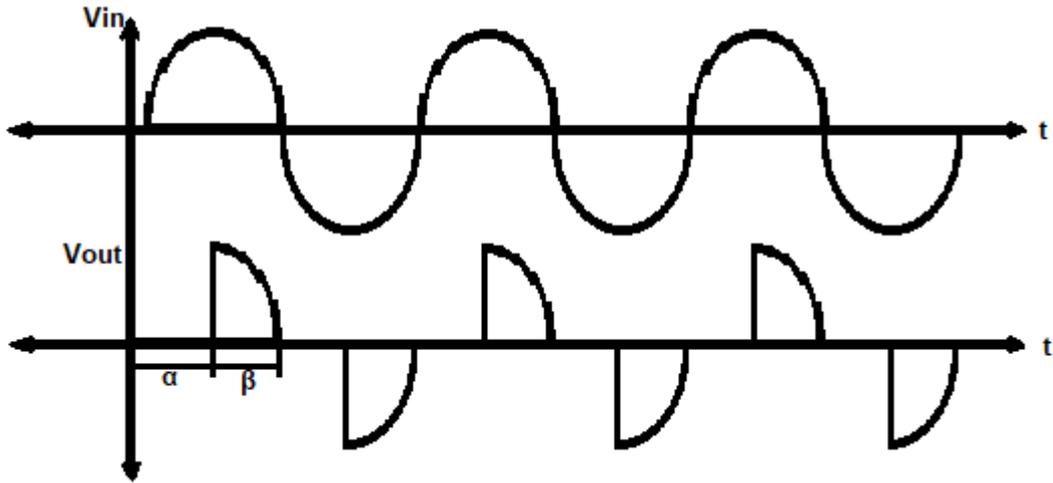
Formas de onda de entrada y salida de un control on/off, elaborado por: Paúl Pachacama

### 2.4.2 Control de ángulo de fase

El principio de control de fase consiste en recortar la onda de corriente alterna haciendo así que se regule la potencia suministrada a la carga, este tipo de control es utilizado en calefacción y la iluminación. Para este tipo de control se tiene un tiempo de

conducción y un tiempo de apagado para cada semiciclo de la onda. (Rashid, 2015)  
(Zambrano & Gonzales, 2013)

Figura 2.3 Control de fase



Control de fase onda completa, elaborado por: Paúl Pachacama

## 2.5 Herramientas de software

En esta sección se presenta todo lo referente a la teoría de las diferentes herramientas de software que se utilizaron, así también se describe software y lenguajes de programación esenciales para el funcionamiento del proyecto.

### 2.5.1 Simple Object Access Protocol (SOAP)

El protocolo SOAP tiene como antecesor el protocolo XML-RCP, en el año 2000 se realizó la primera recomendación para este protocolo llamado W3C.

SOAP busca combinar tecnologías existentes, para no tener la necesidad de inventar nuevas tecnologías y así ser más eficiente, básicamente es un formato de intercambio de mensajes para comunicar aplicaciones. El protocolo SOAP combina XML para lo que es la parte de codificación de los diferentes mensajes y utiliza HTTP como protocolo de transporte, pero esto no quiere decir que sea el único protocolo de transporte que utiliza SOAP, también utiliza FTP, STOP, etc.

Por las más grandes compañías como Microsoft, IBM, SUN ha sido aceptado este protocolo, en el Ecuador una de las empresas de telecomunicaciones llamada BANRED también utiliza protocolo SOAP, esto quiere decir que no es un protocolo viejo u obsoleto, solo que no ha sido explotado en su totalidad en la parte de la ingeniería. (Báez, 2018) (Gutiérrez, 2016)

A continuación, se describe alguna de las ventajas del protocolo SOAP:

- Es un protocolo para las diferentes comunicaciones en Internet.
- SOAP está disponible para todas las plataformas.
- Tiene una buena seguridad siendo simple y altamente efectiva.
- Una de las ventajas más grandes es que es un protocolo que permite pasar firewalls.
- No se encuentra ligado estrechamente a un solo protocolo de transporte.
- Aprovechas las diferentes tecnologías existentes en la industria.(Báez, 2018)

El protocolo SOAP se encarga en el presente proyecto, el intercambiar los datos (Mensajes) desde el servidor hacia el cliente (Motor AC).

### **2.5.2 Lenguaje de Marcación de Hipertexto (HTML)**

HTML es un lenguaje que llegó a ser un estándar en 1993, sólo requiere un editor de texto para poder ser programado y se necesitaría un navegador de internet para que pueda ser interpretado el código, mostrándose al usuario: imágenes, videos, colores y diferentes diseños que se puede observar en una página WEB. La World Wide Web Consorcio (W3C) es una organización que define las normas de HTML, la última versión que ha salido y es la más conocida en la actualidad es la HTML5 (Romero Alamán, 2017).En el presente proyecto se utiliza este lenguaje para la interfaz de la página WEB donde se visualiza la gráfica de la velocidad del motor AC, el ángulo del Fuzzy y LQG. (Pillajo & Roberto, 2018)

### **2.5.3 Protocolo de Transferencia de Hipertexto (HTTP)**

HTTP es el protocolo que todos los navegadores utilizan para realizar peticiones a los servidores web y con esto recibir las respuestas de ellos (Alicia Ramos Martín & Martín, 2014).En la actualidad HTTP prácticamente permite la conversación entre navegadores y servidores Web, siendo esta una de las características más relevantes de este protocolo. Tim Berners Lee unió la tecnología y HTTP, para que todo ser humano en este planeta pueda navegar en el internet.(Gourley, Totty, Sayer, Aggarwal, & Reddy, 2002)

#### **2.5.4 Notación de Objetos de JavaScript (JSON)**

JSON es un tipo de formato que intercambia datos, siendo más fácil para analizar. JSON tiene un formato de texto independiente del idioma, esto implica que es más rápido en comparación con otros formatos de intercambio de datos como el XML, llegando a la conclusión de que es un lenguaje ideal. (Wang, 2011)

En el presente proyecto se utiliza JSON para la fácil transmisión y discriminación como son la velocidad del motor, el Setpoint y el ángulo.

#### **2.5.5 Pre Procesador de Hipertexto (PHP)**

PHP es un lenguaje de programación que tiene código abierto, esto quiere decir que cualquier desarrollador de páginas web puede acceder a este código de cliente - servidor. Este lenguaje es especialmente utilizado para el desarrollo de páginas web siendo fácil de incrustar en HTML. En los últimos años los clientes, ya no están interesados en el diseño sino en la funcionalidad que presta la página web.

El código de PHP se ejecuta en el servidor, por esta razón al hacer una inspección de una página web no sale el código de PHP porque así el usuario no podrá saber lo que se tiene detrás, a diferencia de JavaScript que se ejecuta en la computadora.(Beati, 2015)

En el presente proyecto se programó los diferentes controladores tanto Fuzzy y LQG en PHP.

#### **2.5.6 JavaScript**

En un inicio JavaScript funcionaba del lado del cliente esto quiere decir que se ejecutaba un script solo en los navegadores web, no necesariamente viajaba hacia el servidor. De esta manera se logra que la interpretación y ejecución del lenguaje sea encargado por el navegador, este tiene la capacidad de ejecutarlo a su manera.(Castillo, 2017)

JavaScript tuvo su auge cuando las páginas HTML estáticas pasaron a ser dinámicas sin tener la necesidad de viajar al servidor, así se ahorra mucho ya que no todo hace el servidor, el cliente también cumple su función. Cabe recalcar que JavaScript usa la estructura del lenguaje C por ejemplo usa if, while, for.(Castillo, 2017)

En el presente proyecto se utiliza JavaScript para que el usuario tenga una mejor experiencia con la página web creada, porque la gráfica de la velocidad y la

actualización de los datos tanto del ángulo como de la velocidad, se utilizó JavaScript junto con Ajax.

### **2.5.7 Base de Datos MySQL**

MySQL es el encargado de administrar la base de datos esto quiere decir, que los diferentes datos obtenidos se puedan almacenar, ordenar, buscar y lo más importante, poder recuperar esos datos, ya que muchas veces en grandes empresas se han perdido los datos, convirtiéndose así en un gran problema, por esta razón MySQL facilita al usuario el poder recuperar dichos archivos, también al ser muchos datos poderlos ordenar y facilitar la búsqueda de los mismo. (Luke Welling & Thompson, 2009)

Otra de las funcionalidades de MySQL es que es seguro no cualquier persona o usuario puede ingresar a esta base de datos, debe estar con permisos para que pueda acceder. En otras palabras, el servicio de MySQL controla el acceso, también se puede decir que no es necesario que solo acceda un usuario, otros usuarios simultáneamente puede acceder a esto se llama multiusuario. (Luke Welling & Thompson, 2009)

MySQL al igual que PHP tiene código abierto. Algunas ventajas de MySQL: tiene un bajo costo, es fácil de usar, además tiene un alto rendimiento, esto quiere decir que es rápido.(Angel Cobo, Patricia Gómez, Daniel Pérez, & Rocha, 2005)

En el presente proyecto se utilizó MySQL para almacenar datos críticos del sistema, como son: Setpoint y datos temporales del algoritmo LQG (estados estimados).

### **2.5.8 Arduino IDE**

Un IDE es un entorno de desarrollo integrado utilizado en la informática, Arduino utiliza este entorno ya que tiene las siguientes características: un editor de código, una interfaz gráfica y un depurador, siendo así una herramienta de software muy utilizada por su practicidad.

Una de las ventajas más notorias es el poder trabajar con varios lenguajes de programación, sirve para poder grabar los diferentes programas en la memoria flash de los diferente Arduino existentes en el mercado.

Es gratis teniendo accesibilidad al código fuente en la página oficial, puede ser instalada en diferentes plataformas como Windows, Mac OSX, Linux y Microsoft. Otra de sus ventajas es su interfaz ya que es fácil para que el usuario no se pierda mientras programa su código, y también se puede pasar el depurador para que detecte

errores en el código, hay que tener en cuenta que solo detecta errores de sintaxis mas no de lógica.(Millahual, 2017)

En el presente proyecto se utilizó el Arduino IDE el cual sirvió para la programación del Arduino por medio de la red WIFI local como cliente para que realice las tareas tanto de consulta como de actualización de datos.

### **2.5.9 Ident de Matlab**

Es una herramienta que nos permite obtener lo que son los modelos matemáticos utilizando datos de entrada y salida de una planta, estos datos son adquiridos en el dominio del tiempo o también pueden ser en el dominio de la frecuencia, permitiendo así conseguir funciones de transferencias en tiempo continuo o discreto , para el presente proyecto se utiliza la función de transferencia en tiempo continuo con el fin de obtener las matrices de estados las cuales describen de otra forma el comportamiento de la planta. La función de transferencia nos ayuda en la predicción de la respuesta del sistema. (The MathWorks, Inc., 1994-2018)

### **2.6 Hardware utilizado en el proyecto**

En esta parte se detalla las materiales y elementos utilizados en el presente trabajo, tanto la implementación y construcción del hardware.

#### **2.6.1 Arduino mega 2560**

Es una plataforma de hardware libre, está diseñado para la realización de diferentes proyectos. Tiene un microcontrolador ATmega2560, 54 pines que pueden ser utilizados tanto como entradas o salidas y también algunos tiene la característica de ser un pin dependiendo de la configuración ser digital o analógico.(Campos Verdi, 2016)

Figura 2.4 Arduino Mega 2560



Tarjeta Arduino Mega 2560 utilizada en el proyecto, Silvia Chuquín & Paúl Pachacama

En el presente proyecto se usaron los pines de la siguiente forma:

- PIN 2: Como interrupción en flanco de bajada para detectar el cruce por cero de la onda de voltaje alterno.
- PIN 3: Como interrupción en flanco de subida para realizar un conteo de las RPM, cada interrupción nos muestra un desplazamiento de 120° en el eje del motor.
- PIN 9: Como salida digital para enviar la señal de disparo para activar el TRIAC con cierto ángulo y un por un tiempo de 2us, este tiempo depende del TRIAC a usar.

### 2.6.2 Módulo Dragona Yun Shield

El Dragino Yun Shield obtiene su voltaje de la placa Arduino a la que está conectada, sirve para llevar los proyectos de un Arduino normal a la nube, teniendo un control de algún sistema en una red WLAN o WAN.(Kuhnel, 2015)

Una de las características más relevantes del módulo Dragino Yun Shield es el disponer de ethernet y wifi, teniendo así la elección de ser configurado como cliente mediante la librería HttpClient la cual permite hacer peticiones tales como: HTTP, GET, POST y PUT a un servidor web. (Reina & Vélez, 2015)

Figura 2.5 Módulo Dragino Yun Shield



Módulo Dragino Yun Shield utilizado en la implementación del proyecto, Silvia Chuquín & Paúl Pachacama.

### 2.6.3 Raspberry Pi

Con el uso de Raspberry Pi se puede desarrollar varios proyectos en el ámbito de hogares domóticos inteligentes, en el mercado se encuentra muchos dispositivos electrónicos para el hogar inteligente. Es muy utilizada en el ámbito de la IoT (Internet de las cosas) (Shantanu Bhadoria & Ramos, 2017), se trata de un computador pequeño prácticamente del porte de una mano humana, uno de los mejores modelos que ofrece este hardware es la Raspberry Pi 3 Model B que salió al mercado en febrero del

2016.Las diferentes características que posee se pueden observar en la siguiente cita.(Raspberry, 2016)

Figura 2.6 Raspberry Pi 3 Model B

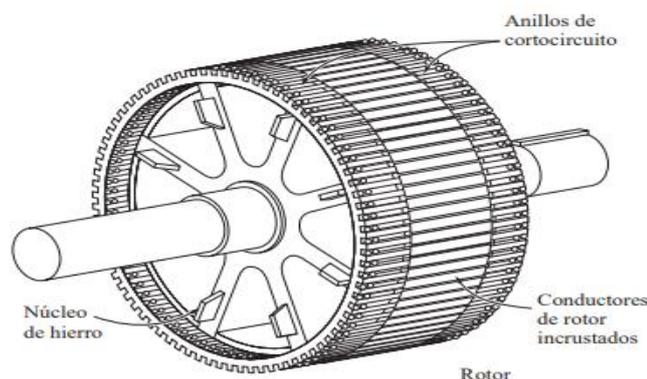


Tarjeta Raspberry Pi 3 Model B manipulada en la implementación del proyecto, Silvia Chuquín & Paúl Pachacama

## 2.7 Motor de Inducción Monofásico Jaula de Ardilla

El motor de inducción monofásico es un motor el cual produce un voltaje inducido en las barras del rotor, las cuales están cortocircuitadas, por el movimiento relativo del rotor en comparación con el campo magnético del estator. Sin embargo, existe un límite para la velocidad del motor, si el rotor gira en sincronía con el campo magnético no existiría ningún voltaje inducido, tampoco existiría corriente y como resultado la velocidad del motor comenzaría a disminuir, a este fenómeno se lo conoce como deslizamiento y es propio de los motores de inducción tanto del tipo Jaula de Ardilla como del de Rotor Devanado. (Chapman , 2012)

Figura 2.7 Rotor tipo Jaula de Ardilla



Aspecto del rotor de un motor asíncrono, Fuente: (Chapman , 2012)

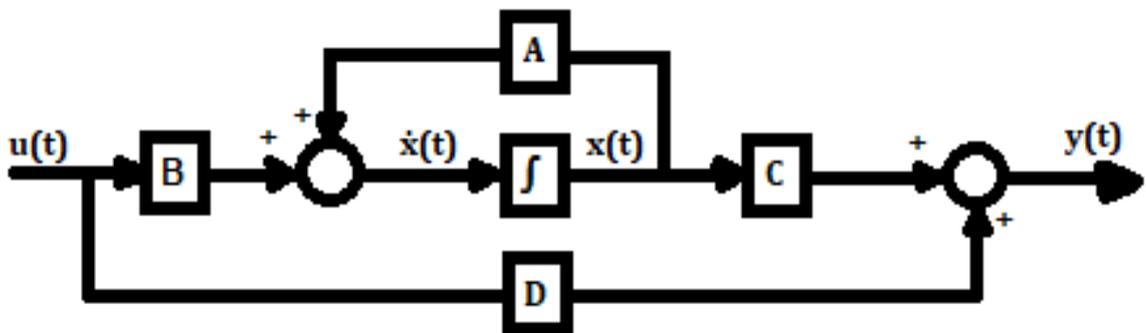
## 2.8 Representación en espacio de estados

La representación en espacio de estados es una forma de representar un sistema mediante matrices, en vez de una sola función como lo es la función de transferencia

y depende de un modelo matemático, la representación por espacio de estados describe internamente a un sistema y se obtiene a través de la función de transferencia, ambas describen el comportamiento de un sistema. (Dominguez, Campoy, Sebastián, & Jiménez, 2006).

Para una representación de un sistema lineal invariante en el tiempo se requieren una serie de matrices que relacionan las variables con la entrada, estas se expresan con las letras A y B, mientras que para la salida se requiere de dos ecuaciones que relacionen las variables de estado y las entradas con la salida, estas se representan con las letras C y D.

Figura 2.8 Diagrama de bloques en variables de estado



Representación en esquema de estados para un sistema lineal, elaborado por: Paúl Pachacama

## 2.9 Control FUZZY

Un controlador difuso está asociado más al pensamiento humano, ya que el ser humano no percibe las cosas de forma exacta, sino que las asocia en conjuntos que el individuo reconoce al relacionar un objeto o un fenómeno con algún objeto o fenómeno similar. Para que un controlador pueda percibir el mundo como lo hace un ser humano se utiliza un tipo de conjuntos que el programador asocia a su forma de analizar el fenómeno, este tipo de conjuntos se les denomina conjuntos difusos y se asigna un rango de la variable de entrada del sistema.

Un conjunto difuso se define como una ambigüedad y son una extensión de la teoría de conjuntos donde un elemento pertenece o no a un conjunto. Con los conjuntos difusos se intenta emular el pensamiento humano. (Kouro & Musalem, 2002).

### 2.9.1 Funciones de Membresía

Son funciones las cuales son una representación del grado de pertenencia de un valor a un conjunto difuso definido por un nombre, las funciones de membresía deben constar de un conjunto difuso y al juntar varias funciones de membresía se forma una entrada o

salida difusa, las cuales deben constar de un número impar menor o igual a nueve, el poner un mayor número de funciones de membresía no solo complica los cálculos posteriores, pero a cambio hace que se incremente la exactitud del controlador.

Las funciones de membresía pueden adoptar varias formas las cuales son:

- Triangulares
- Sigmoides
- Trapezoidales
- Gaussianas

### **2.9.2 Reglas de inferencia**

Las reglas de inferencia son utilizadas para definir el comportamiento del controlador ante un cambio en la entrada del sistema, esta es la forma en la que el controlador enlaza las entradas con la salida. Las reglas de inferencia pueden considerar los elementos de las funciones de membresía de la siguiente forma:

- Inferencia por el Máximo: Cuando se toma en cuenta el valor más alto obtenido de las funciones de membresía involucradas.
- Inferencia por el Mínimo: Se da cuando el valor más importante es el valor más bajo de las funciones de membresía involucradas.

### **2.9.3 Fuzzificación**

Es una operación matemática la cual se realiza en todo instante de tiempo y convierte un valor del universo en un valor para cada función a la cual se presenta. La Fuzzificación junto con las reglas de inferencia nos permiten obtener los datos necesarios para poder obtener una salida final.

### **2.9.4 Defusificación**

Es un proceso matemático el cual se utiliza para convertir un conjunto difuso en un resultado real, el sistema obtiene este resultado a partir de la información obtenida de la entrada, pero en términos difusos (Kouro & Musalem, 2002). Existen varios métodos para defusificar los datos obtenidos y son los siguientes:

- Método del máximo: obtiene un resultado únicamente con el valor más alto, tiene muy poca precisión ya que el valor máximo no siempre es la salida deseada.

- Método de la media del máximo: obtiene el valor de salida haciendo un promedio de los valores más altos en caso de que se repitan.
- Método del centro de área: es uno de los métodos más utilizados porque tiene una gran precisión y además es fácil de implementar ya que únicamente suma los valores obtenidos y obtiene un promedio evaluando los datos obtenidos a través de los conjuntos de membresía de la salida.
- Método del centro de gravedad: es el método más preciso, pero para su cálculo requiere obtener el centro de masa de la salida difusa final.

De los métodos descritos se considera el método por centro de área como uno de los métodos que ofrece una gran exactitud para el fin práctico de este proyecto.

### **2.9.5 Ventajas y desventajas del control difuso**

Las ventajas que tiene el controlador Fuzzy son:

- Permite un método eficiente y efectivo para desarrollar un controlador no lineal sin utilizar recursos matemáticos muy avanzados.
- El controlador no necesita necesariamente el modelo del sistema ya que se puede programar el controlador a voluntad mediante las reglas de inferencia.

Las desventajas que tiene el controlador Fuzzy son:

- El algoritmo no cuenta con herramientas de análisis, además el diseño es muy sofisticado especialmente si se utilizan muchas funciones de membresía.
- El análisis de estabilidad es muy complejo y dependiendo del tipo de sistema puede requerir un algoritmo adicional que ayude a mantener el sistema en estabilidad ya que con este algoritmo la estabilidad no está completamente garantizada.

### **2.10 Control Lineal Cuadrático Gaussiano – LQG**

El control Lineal Cuadrático Gaussiano se conoce por su gran inmunidad al ruido y a las diferentes perturbaciones que se puedan presentar en un sistema, conocidos como ruido gaussiano, tiene varias aplicaciones tanto en sistemas lineales como no lineales. El controlador resulta de combinar un control LQR y un estimador el cual ayuda a predecir un estado futuro del sistema. El modelo del proceso requiere ecuaciones de

entrada y asociación con la salida independientes de cualquier perturbación.(Golden Kumar, 2013)

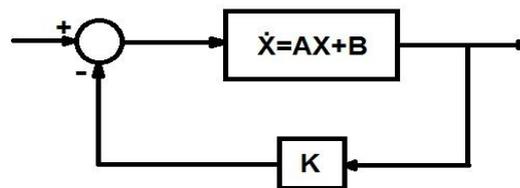
Una solución a un problema utilizando el control LQG se hace subdividiéndolo en dos problemas independientes:

- El problema de control resuelto mediante control LQR.
- El problema del observador, resuelto como filtro de Kalman.

### 2.10.1 Control LQR

El regulador cuadrático lineal o LQR es un algoritmo de control óptimo el cual minimiza una función cuadrática para calcular la ley de control para optimizar la eficiencia de un proceso, para que este controlador regule de manera eficiente se debe calcular un parámetro de optimización de un índice cuadrático (L).

Figura 2.9 Diagrama de bloques controlador LQR



Diseño de un controlador LQR, elaborado por: Paúl Pachacama

### 2.10.2 Filtro de Kalman

El filtro de Kalman o estimador de estados se comporta como un algoritmo predictivo el cual se diseña para tomar acciones de control ante ruidos dentro de un proceso, como el ruido es un elemento inestable que está presente en todo tipo de sistemas se debe tener presente que las estimaciones calculadas por el estimador no son siempre exactas.

Para procesos dinámicos se debe tener presente que para cada estado del sistema se debe calcular de diferente forma el filtro ya que, aunque se trate del mismo proceso o sistema la ganancia del filtro debe ser distinta ya que en un estado del proceso puede dar una mejor estimación del comportamiento, pero en otro estado se incrementa el error.

Figura 2.10 Diagrama de bloques filtro Kalman

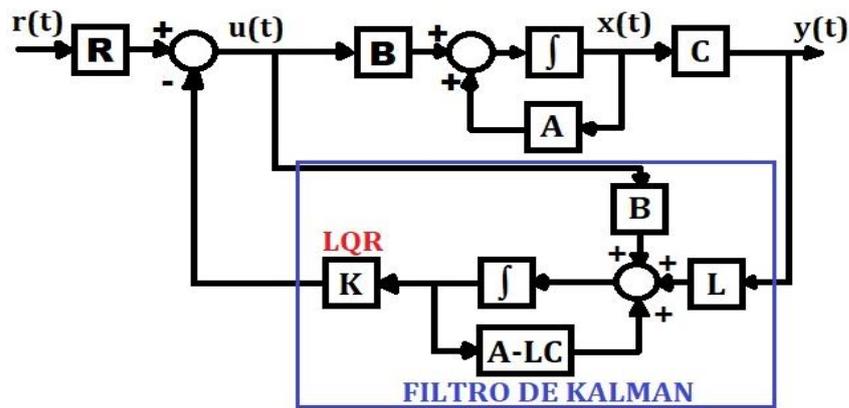


Diagrama de bloques utilizando las ecuaciones de estado, elaborado por: Paúl Pachacama

### 2.10.3 Ventajas y Desventajas del control LQG

Las ventajas del control LQG son:

- Minimiza el error utilizando un estimador el cual hace que el sistema sea más estable y el ruido no tenga mayor influencia dentro del sistema.
- El control LQG al utilizar un modelo interno de la planta hace que se conozca más a fondo el comportamiento del sistema lo cual hace que el desempeño del controlador sea mejor comparado con otros métodos de control clásico.

Las desventajas del control LQG son:

- Este método de control depende mucho del modelamiento interno de la planta es por esto por lo que si el modelamiento es realizado de forma incorrecta o inexacta se puede tener un desajuste del controlador y tener desestabilizaciones.
- El controlador tiene una matemática compleja para implementar ya que utiliza matrices para las variables de estados además de utilizar ganancias las cuales deben ser calculadas correctamente ya que de lo contrario pueden desestabilizar el sistema.

## CAPÍTULO 3

### DISEÑO E IMPLEMENTACIÓN DE LOS ALGORITMOS

En el capítulo 3 se realiza el diseño de los circuitos tanto de la etapa de control como de la etapa de potencia además del modelamiento matemático del motor monofásico de inducción tipo jaula de ardilla, tanto en función de transferencia como en variables de estado, también se realiza el diseño de los controladores FUZZY y LQG.

#### 3.1 Implementación de las etapas de control y potencia

Las etapas de control y potencia permiten al controlador obtener y controlar la planta ya que este no lo puede hacer directamente.

A continuación, se muestra un diagrama de bloques donde se representa la conexión del proyecto.

Figura 3.1 Esquema del proyecto

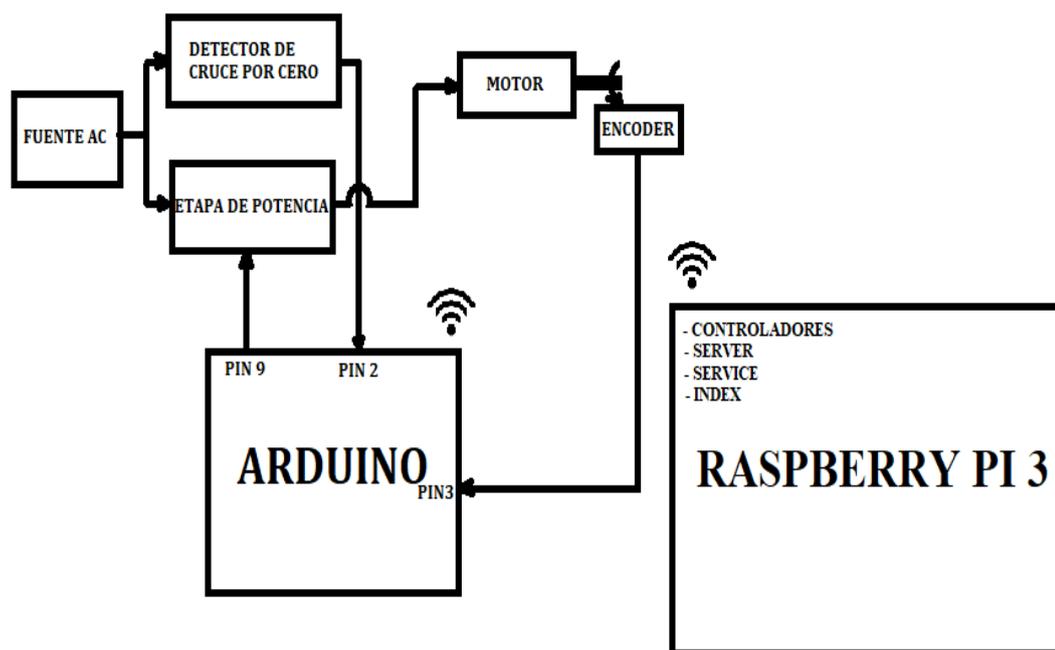


Diagrama de conexión del proyecto, elaborado por: Paúl Pachacama

#### 3.1.1 Diseño de la etapa de potencia

Para controlar la planta se requiere únicamente enviar un pulso para activar el semiciclo de la señal de corriente alterna, dependiendo del ángulo de disparo se deja pasar un mayor voltaje RMS del voltaje alterno.

El diseño final de la etapa de potencia se muestra en la siguiente figura;

Figura 3.2 Diagrama de potencia

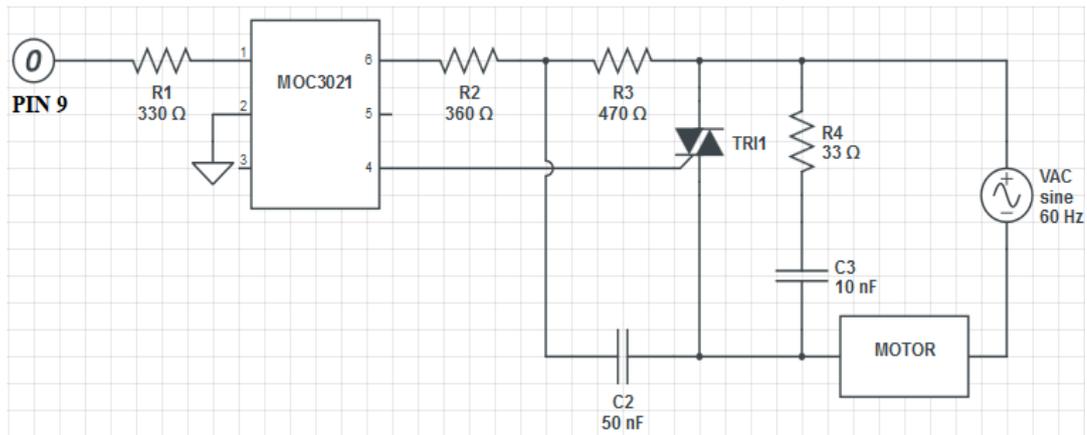
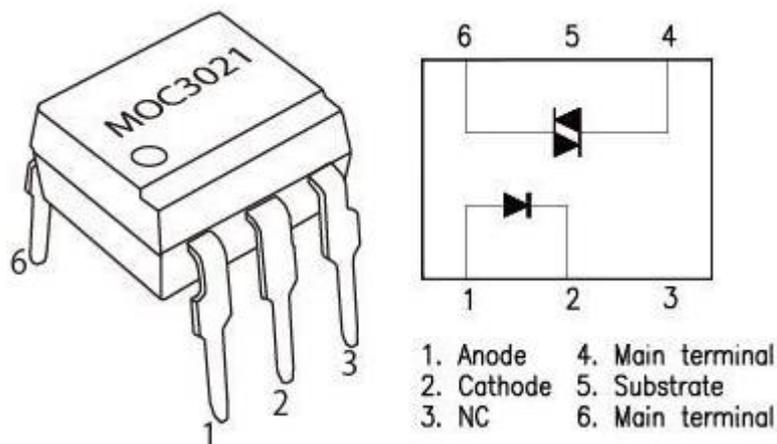


Diagrama del acoplamiento de la señal de control y el motor monofásico, elaborado en CircuitLab, En cuanto a los elementos expuestos se tiene que el valor de R1 es de 330 para limitar la corriente a 1mA, el capacitor y la resistencia en serie R4 y C3 hacen una red Snubber la cual permite mejorar la conmutación del TRIAC el cual por sus características de voltaje y corriente se escogió el BT16-600b con capacidad máxima de 600V y 16 A. Las resistencias R2 y R3 junto con el capacitor C2 tienen valores sugeridos por el fabricante del MOC3021.

Esta parte necesariamente debe estar conectada a la parte de control, pero mediante un optotriac el cual permite aislar el voltaje alterno de las señales de bajo voltaje, MOC3021 tolera 400V y 1A lo que lo hace ideal para aplicaciones de baja potencia.

Figura 3.3 Optoacoplador aislador



Distribución de pines del MOC3021

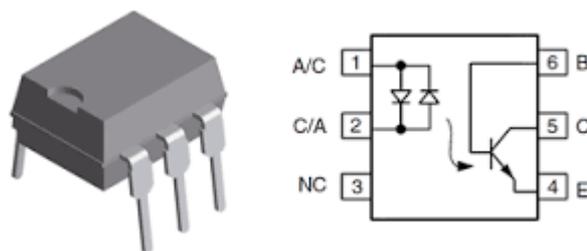
### 3.1.2 Diseño de la etapa de control

Para que la etapa de control y la etapa de potencia estén debidamente sincronizadas se debe obtener el cruce por cero de la señal del tendido eléctrico para que el controlador

pueda saber cuándo poner en marcha el temporizador que va a dar la activación del ángulo de disparo.

Como la señal del tendido eléctrico es de 110VAC, este puede afectar a los dispositivos como los microcontroladores que apenas manejan 5VDC, se utilizan optoacopladores para aislar los voltajes, existen también transformadores de pulsos, pero estos son más utilizados para voltajes y corrientes más grandes. En este caso el integrado desarrollado por Arduino H11AA1 nos permite obtener la señal de cruce por cero sin necesidad de utilizar un puente rectificador.

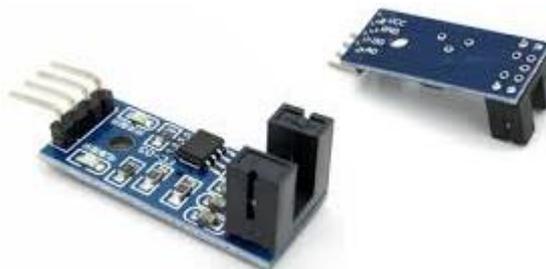
Figura 3.4 Detector de cruce por cero Arduino



Distribución de pines detector de cruce por cero

Las RPM de un motor de corriente alterna se define por el número de polos que tenga, en el motor usado tiene 3 polos quiere decir que tiene un máximo de 1800 RPM, pero como se trata de un motor jaula de ardilla esta velocidad se ve más reducida. Para medir las RPM se tiene varias alternativas como es el acoplamiento mecánico a un generador DC, también existe un módulo de Arduino el cual consta de una placa con un sensor óptico, de esta forma se obtiene las revoluciones contando los pulsos, además se puede aumentar o reducir la precisión del encoder aumentando o disminuyendo las ranuras del disco.

Figura 3.5 Placa Encoder Óptico



Opto interruptor para detectar las ranuras del disco.

El diseño final de la etapa de control se muestra en la siguiente figura:

Figura 3.6 Diagrama parte de control

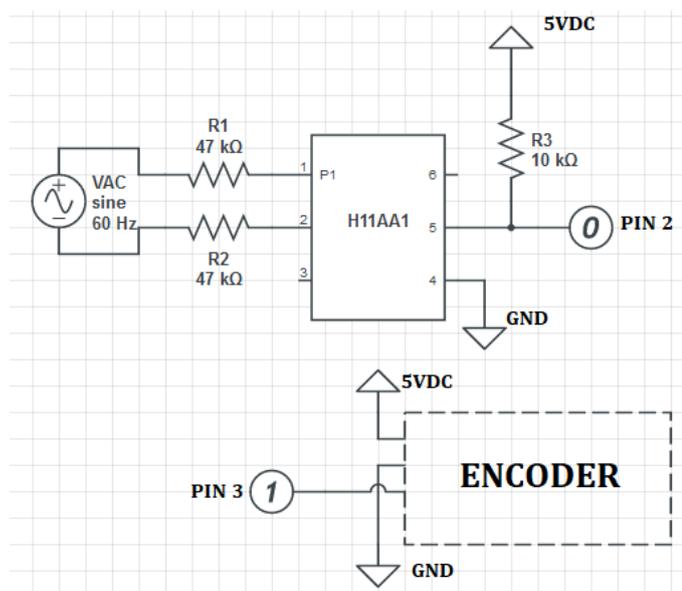


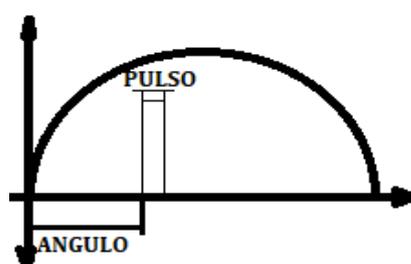
Diagrama del detector de cruce por cero y encoder, realizado en CircuitLab

Los valores expuestos de resistencias R1, R2 y R3 son los propuestos por el fabricante.

### 3.1.3 Programación del ángulo de disparo en el Arduino

Para realizar el disparo para que el TRIAC conduzca se debe considerar que la duración de un semiciclo de la onda dura 16,67ms, por lo que el ángulo de disparo debe tener un retardo máximo de 16ms para evitar que el disparo haga que conduzca el semiciclo posterior, así como se debe tener un retardo mínimo de 1ms para asegurar que el TRIAC este polarizado inversamente al semiciclo anterior y pueda conducir adecuadamente.

Figura 3.7 Ángulo de disparo para un semiciclo



Representación de un pulso en un semiciclo, elaborado por: Paúl Pachacama

La duración del pulso depende del TRIAC a utilizar, en la mayoría de los casos la duración del pulso debe ser mayor a 1ms, pero para asegurar la conducción del TRIAC se deja el tiempo de disparo en 2ms.

### 3.2 Diseño del controlador FUZZY

Para el diseño del control Fuzzy se debe tener disponible el comportamiento de la planta.

Figura 3.8 Diagrama de bloques control Fuzzy

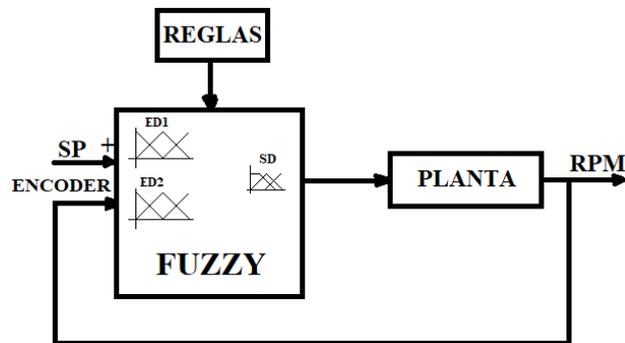
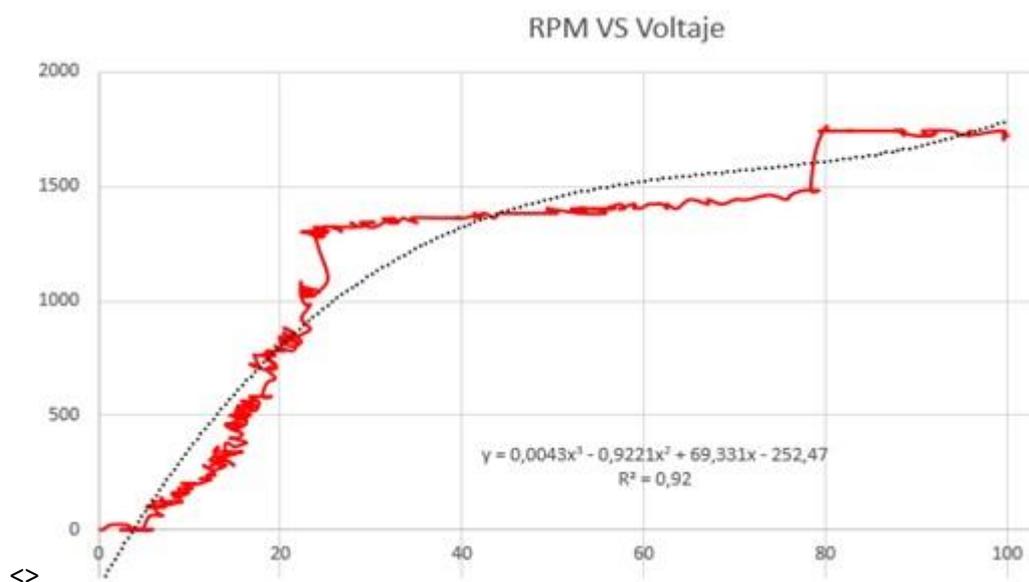


Diagrama de diseño controlador Fuzzy, elaborado por: Paúl Pachacama

El controlador Fuzzy requiere en este caso es necesario conocer la variación de la velocidad del motor en función del voltaje de alimentación, lo cual se puede obtener al alimentar el motor con voltajes de 0 V hasta el Voltaje nominal en pasos de 0,1 V, los datos obtenidos muestran en el anexo Tabla B.1 se puede obtener la siguiente grafica del sistema. Tomando como estado inicial el sistema en reposo, es decir RPM=0 y Angulo=180°:

Figura 3.9 Toma de datos motor monofásico



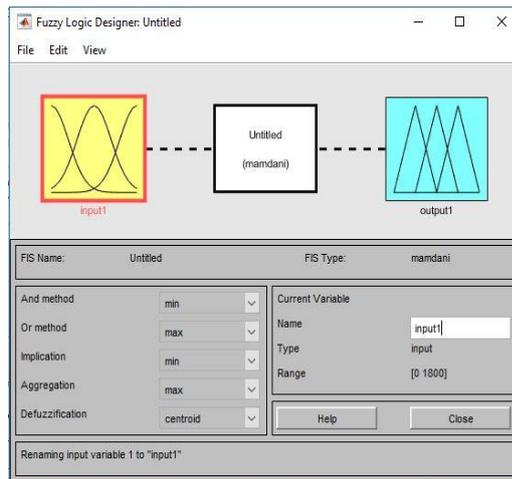
Gráfica de RPM vs Voltaje, elaborado por: Silvia Chuquín

A partir del comportamiento obtenido en la Figura 3.9 se logra apreciar que el comportamiento del sistema es no lineal, además de que para el diseño del algoritmo

se debe tomar en consideración tanto el ángulo de disparo como la velocidad que se genera.

El diseño del controlador para tener una referencia de funcionamiento se lo puede realizar en Matlab mediante el comando *Fuzzy* donde aparece una nueva ventana donde tenemos varias opciones de diseño.

Figura 3.10 Ventana función Fuzzy



Ventana mostrada al hacer compilar la función en Matlab, elaborado en Matlab.

### 3.1.1 Diseño de las entradas difusas

Para el diseño de las entradas difusas se debe considerar lo siguiente:

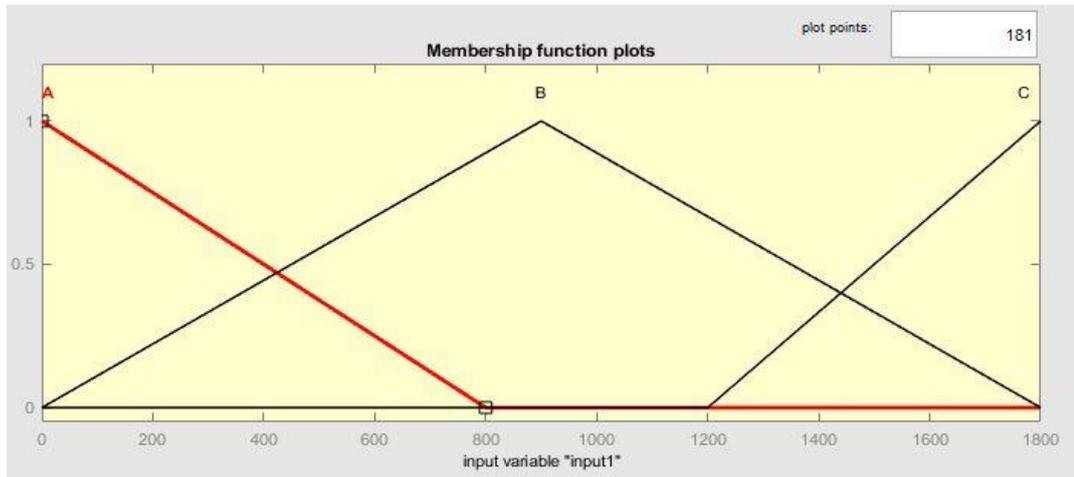
- Los cruces entre funciones de membresía deben ser mayores al 50% del valor en Y de la función, es decir se deben cruzar en referencia al eje Y con 0,5 o más.
- En cada entrada debe existir un número impar de funciones de membresía, pero no debe ser mayor a 9.

Para las entradas del sistema se va a considerar al Set Point como una entrada de referencia y se la considerara como la Entrada Difusa 1, y se van a considerar los siguientes puntos:

- Rango: 0-1800 [RPM]
- Numero de funciones de membresía: 3
- Tipo de funciones de membresía: Triangulares.

El diseño de la Entrada Difusa 1 depende en su mayoría de los datos obtenidos en la Figura 3.9 donde se aprecia las zonas donde se tiene una mayor variación de RPM entre 0-1200RPM así que se considera a estos puntos como cruciales para el diseño de las funciones de membresía, el diseño implementado se muestra a continuación:

Figura 3.11 Diseño de la Entrada Difusa 1



Grafica del diseño de la Entrada Difusa 1, elaborado en Matlab

Matemáticamente cada función de membresía consta de tres partes, en las ecuaciones se muestra matemáticamente los parámetros de cada función de membresía:

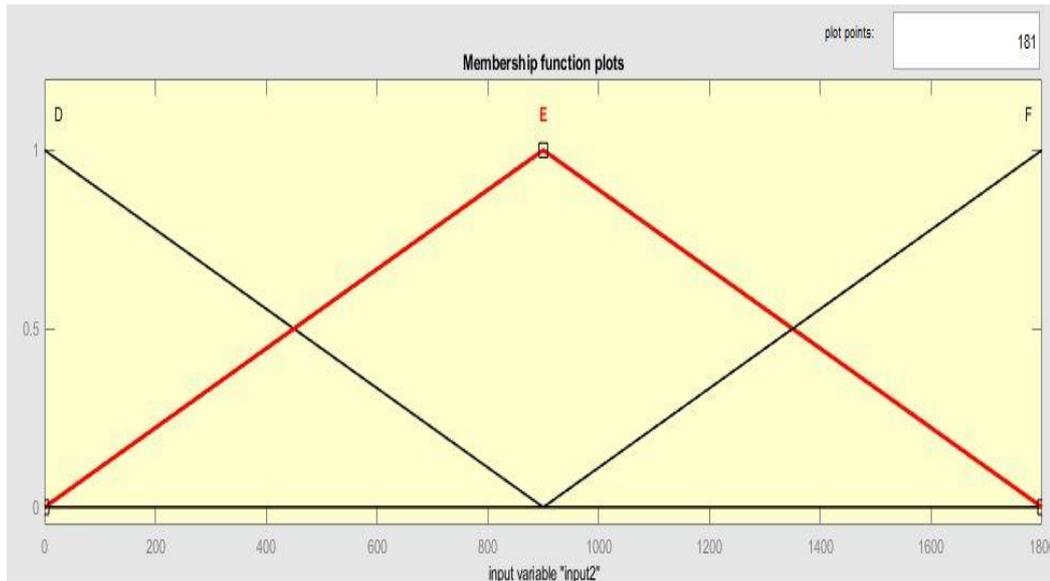
$$u_{bajo} = \begin{cases} \frac{800 - x}{800}, & 0 \leq x < 800 \\ 0, & \text{otro valor} \end{cases}$$

$$u_{medio} = \begin{cases} \frac{x}{800}, & 0 \leq x < 900 \\ \frac{1800 - x}{900}, & 900 \leq x < 1800 \\ 0, & \text{otro valor} \end{cases}$$

$$u_{alto} = \begin{cases} \frac{x - 1200}{600}, & 900 \leq x \leq 1800 \\ 0, & \text{otro valor} \end{cases}$$

La Entrada Difusa 2 va a ser las revoluciones por minuto obtenidas por el encoder acoplado al rotor del motor, para el diseño se consideraron los mismos aspectos en los que está basado el diseño de la Entrada Difusa 1, pero en este caso se debe recorrer las funciones un poco a la derecha a bajas RPM y hacia la izquierda a altas RPM, el diseño implementado se muestra a continuación:

Figura 3.12 Diseño de la Entrada Difusa 2



Grafica del diseño de la Entrada Difusa 2, elaborado en Matlab

Las ecuaciones obtenidas de la Entrada Difusa 2 se muestran a continuación:

$$u_{bajo} = \begin{cases} \frac{900 - x}{900}, & 0 \leq x < 900 \\ 0, & \text{otro valor} \end{cases}$$

$$u_{medio} = \begin{cases} \frac{x}{900}, & 0 \leq x < 900 \\ \frac{1800 - x}{900}, & 900 \leq x < 1800 \\ 0, & \text{otro valor} \end{cases}$$

$$u_{alto} = \begin{cases} \frac{x - 900}{900}, & 900 \leq x \leq 1800 \\ 0, & \text{otro valor} \end{cases}$$

### 3.1.2 Diseño de las reglas de inferencia

Las reglas de inferencia nos ayudan a vincular las entradas difusas con la salida difusa, para la implementación del algoritmo se tomaron en cuenta los siguientes aspectos:

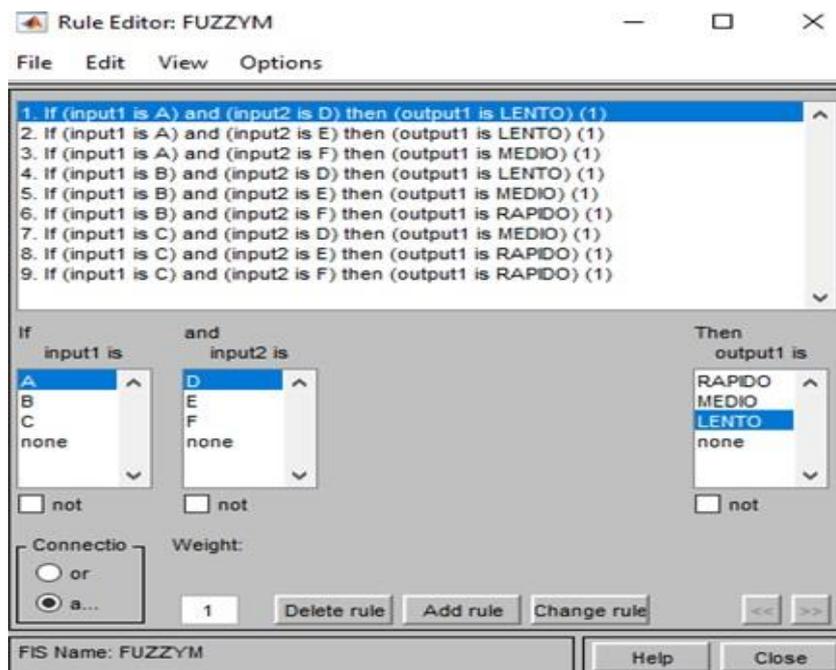
- Siguiendo los modelos de comportamiento se puede observar que el ángulo de disparo es inversamente proporcional a las revoluciones obtenidas por el encoder, esto nos indica que para obtener una velocidad baja debemos enviar un ángulo alto.
- Para el diseño se utiliza la inferencia por el mínimo porque en velocidades bajas de hasta 500 RPM se tiene una variación de ángulo de menos de 30°, si

se lo realizara por el máximo se incrementaría el error porque al momento de vincular los valores obtenidos en las reglas de inferencia con la salida difusa final existirían valores más al centro de la función que en los extremos.

- Las reglas de inferencia marcan el comportamiento y la toma de decisiones del controlador, de ser necesario se debe cambiar en esta parte la asociación de las entradas para cambiar el comportamiento del controlador.

Como se tiene el mismo número de funciones de membresía en las entradas difusas se pueden realizar en la misma ventana de la función *Fuzzy* en Matlab luego se debe presionar las teclas *CRTL+5* donde nos aparece nuestras funciones a un lado y las funciones de la salida al otro lado, aquí se debe seleccionar las funciones de membresía de la entrada y asociarlas a las funciones de membresía de la salida tiene un total de 9 reglas las cuales se detallan a continuación:

Figura 3.13 Reglas de inferencia



Detalle de las reglas de inferencia, elaborado en Matlab

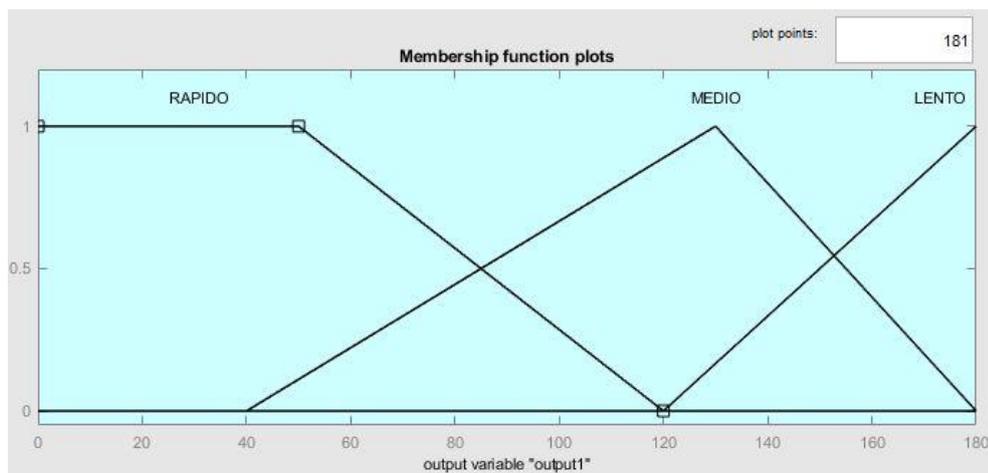
### 3.1.4 Diseño de la Salida Difusa

La Salida Difusa es la parte más importante del algoritmo y su diseño depende de varios factores tales como el comportamiento del sistema, las reglas de inferencia y el método de defusificación utilizado. Para la implementación del controlador se consideraron los siguientes aspectos:

- Debido al comportamiento no lineal del motor se tienen tres zonas distintas, la primera de bajas revoluciones, la segunda que tiene velocidades intermedias y la tercera con velocidades altas, entonces se tomaron 3 funciones de membresía de forma triangular considerando que deben cortar en el eje Y con 0,5 o más.
- La forma de las funciones de membresía no puede ser preparada con anterioridad se deben hacer pruebas del controlador con un diseño básico para analizar mejor el comportamiento ya con el controlador y variar la forma de las funciones para obtener una mejor respuesta del controlador.
- Al considerar el coste computacional y la extensión de la Salida Difusa se considera al método de centro de área como el más efectivo ya que únicamente requiere sumas y multiplicaciones para llegar a un resultado preciso.

El diseño de la Salida Difusa obtenido después de realizar algunos ajustes es el siguiente:

Figura 3.14 Diseño de la Salida Difusa



Grafica del diseño de la Salida Difusa

Matemáticamente se describen las funciones de membresía de la siguiente manera:

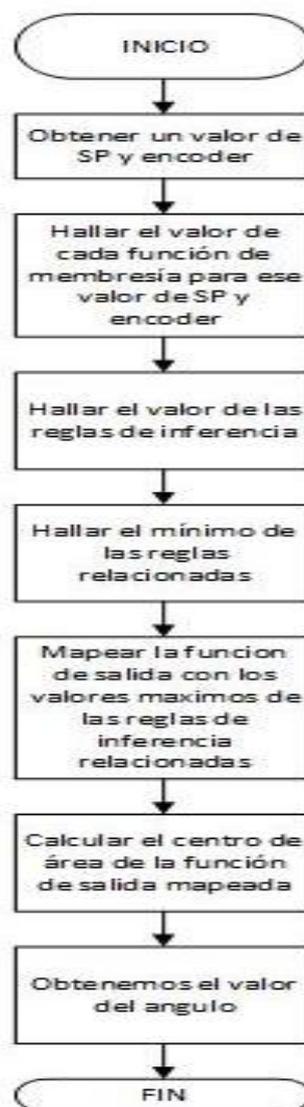
$$urapido = \begin{cases} 1, & 0 \leq x < 40 \\ \frac{120 - x}{80}, & 40 \leq x < 120 \\ 0, & \text{otro valor} \end{cases}$$

$$umedio = \begin{cases} \frac{x - 40}{90}, & 40 \leq x < 130 \\ \frac{180 - x}{50}, & 130 \leq x < 180 \\ 0, & \text{otro valor} \end{cases}$$

$$ulento = \begin{cases} \frac{120 - x}{60}, & 120 \leq x < 180 \\ 0, & \text{otro valor} \end{cases}$$

El proceso del algoritmo se muestra en la siguiente figura:

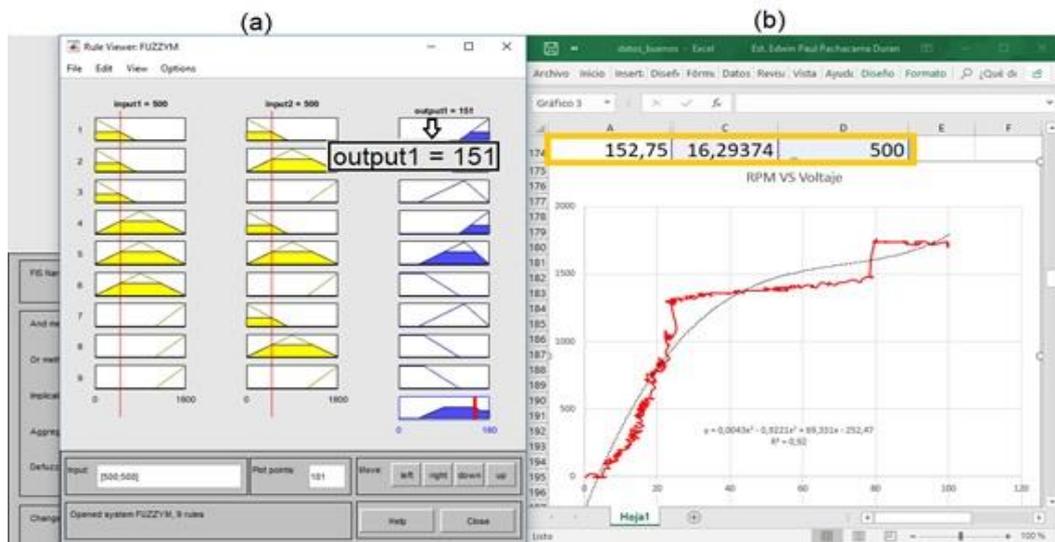
Figura 3.15 Diagrama de flujo del algoritmo FUZZY



Detalle del funcionamiento del algoritmo, elaborado por Silvia Chuquín & Paúl Pachacama

Simulado con el software MATLAB con un Set Point de 500 [RPM] y el encoder de 500 [RPM] se obtiene las siguientes formas en la salida difusa:

Figura 3.16 Simulación del algoritmo FUZZY



Simulación del algoritmo FUZZY bajo 500 RPM de SP, elaborado por: Silvia Chuquín & Paúl Pachacama

### 3.3 Obtención de la Función de Transferencia.

Para obtener la función de transferencia se ingresa al sistema voltaje alterno y se obtiene la reacción de la planta en RPM.

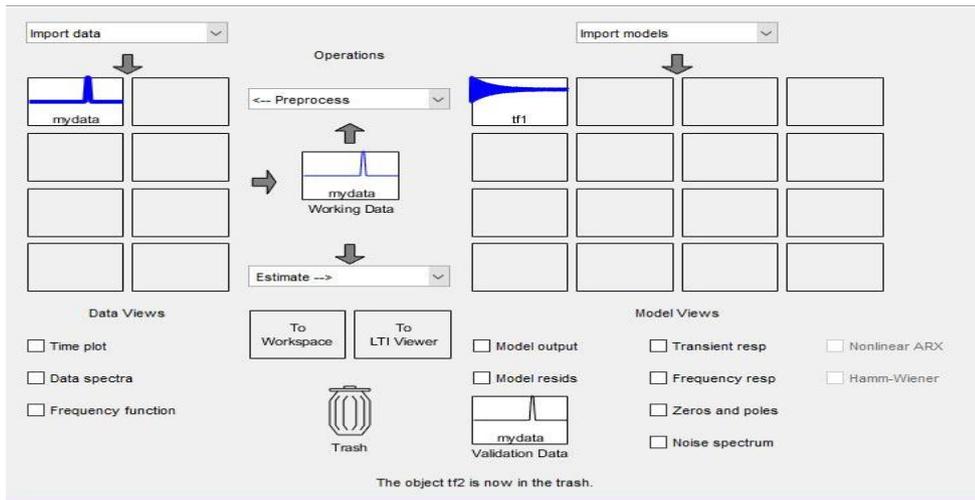
Figura 3.17 Diagrama de bloques de la planta en lazo abierto



Reacción de la planta ante un cambio a la entrada, elaborado por: Paúl Pachacama

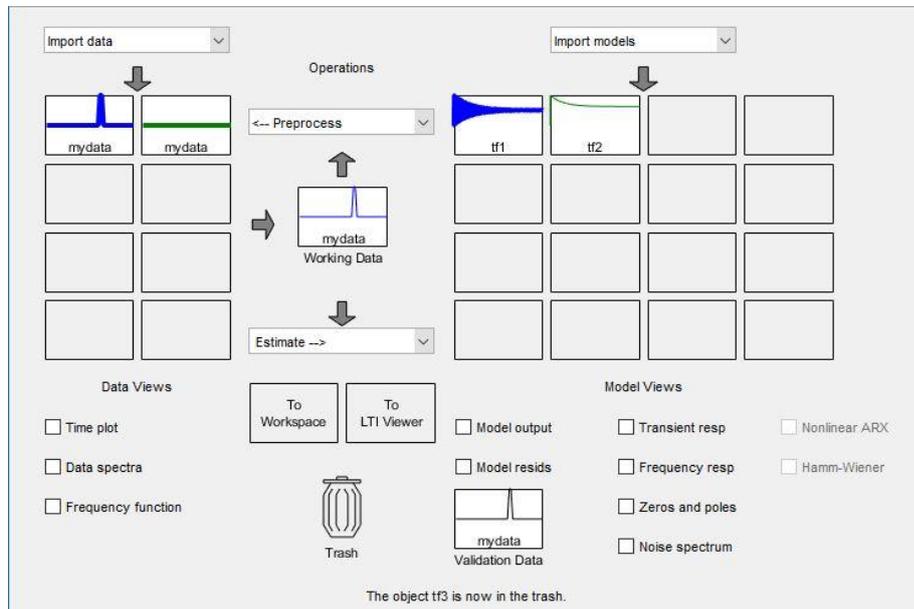
A continuación, se muestra los datos de la función de transferencia utilizando los datos recopilados en la Figura 3.9, los datos son ingresados a MATLAB y por medio de la herramienta IDENT se genera lo siguiente:

Figura 3.18 IDENT de Matlab con un conjunto de datos



Conjunto RPM vs Voltaje importado en la herramienta IDENT de Matlab, elaborado en Matlab  
 Se debe volver a generar datos con el motor para verificar el comportamiento, y se ingresan a la herramienta IDENT para obtener otros datos de comportamiento del motor:

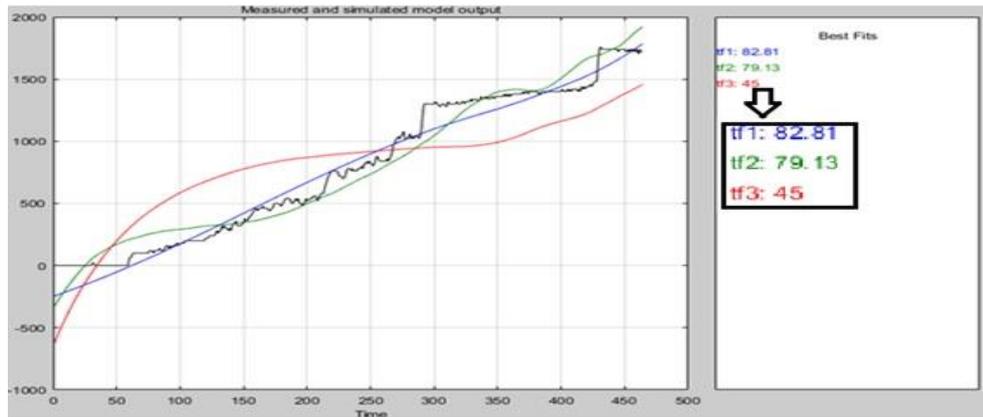
Figura 3.19 IDENT de Matlab con dos conjuntos de datos



Dos conjuntos de RPM vs Voltaje importados en la herramienta IDENT de Matlab, elaborado en Matlab

Para asegurarse que la obtención de datos junto con el modelo generado tenga una buena aproximación se realiza el proceso por tercera vez y al igual que las veces anteriores se ingresan los datos a la herramienta IDENT para su análisis:

Figura 3.20 Estimación de la función de transferencia



Cálculo de la función de transferencia, elaborado en Matlab

A partir de los datos del comportamiento del motor se va a generar la función de transferencia para así tomar la función que tenga el mejor índice de aproximación, se considera como buena aproximación a aquella que tenga 80% o mayor. La función de transferencia utilizada es:

$$tf = \frac{0,04059 s + 0,001236}{s^2 + 3,46e - 07 s + 4,57e - 05}$$

La función de transferencia de un motor de inducción ya sea monofásico o trifásico debe llevar un Zero y dos polos para obtener una mejor aproximación. (Jorge Gallegos, Luis Ballesteros, Vladimir Juárez, & Sánchez, 2016)

### 3.4 Diseño del controlador LQG

El algoritmo de control Gaussiano se compone básicamente de un estimador y de un regulador, juntos hacen que el algoritmo tenga una respuesta eficiente ante cualquier ruido.

El diseño del estimador se lo puede realizar mediante Matlab el cual con el comando *LQR()*, nos permite obtener el resultado óptimo de la matriz K conociendo las matrices de estado A y B, en caso de que la planta lo requiera se debe incluir la matriz de estado C, en este caso las matrices de estados constan los valores de A, B Y C, mediante el comando  $[A \ B \ C \ D]=tf2ss(\text{numerador de la FT}, \text{denominador de la FT})$ , el resultado se muestra a continuación:

$$A = \begin{pmatrix} 0,05482 & -1,6415 \\ 0,038 & 0,9549 \end{pmatrix}$$

$$B = \begin{pmatrix} 0,000173369505170952 \\ 0,00296313899340247 \end{pmatrix}$$

$$C = (-5354,9428)$$

Al ingresar el comando  $LQR()$  se obtiene el siguiente valor de K;

$$K = [0,0930394278753122; 31,1356494176753]$$

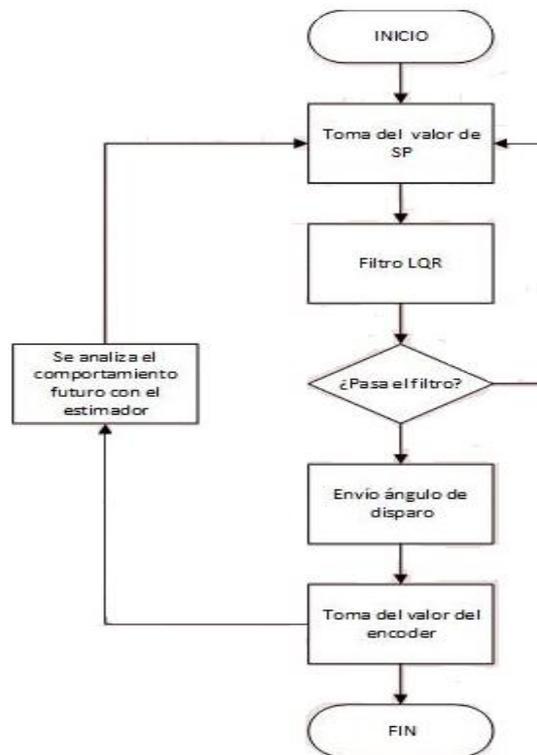
El valor de la constante de Kalman también puede obtenerse mediante Matlab con el comando  $KALMAN()$ , esta función requiere que se conozca la función de transferencia de la planta en matrices de estados, el valor obtenido se representa con la letra L y es el siguiente:

$$L = \begin{pmatrix} -5,87985161274677e - 16 \\ 0,000106067034246058 \end{pmatrix}$$

Para la implementación del algoritmo LQG en el servidor debemos pasar todas las constantes y matrices calculadas a lenguaje PHP para únicamente al obtener un valor de Set Point y Encoder a la entrada entregar correctamente únicamente un valor de salida.

El proceso del algoritmo se muestra en la siguiente figura:

Figura 3.21 Diagrama de flujo controlador LQG



Detalle del funcionamiento del controlador LQG, elaborado por: Silvia Chuquín & Paúl Pachacama

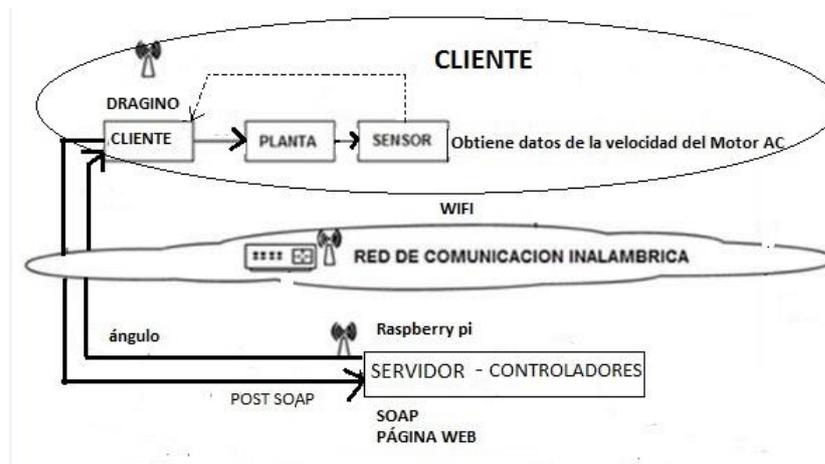
## CAPÍTULO 4

### ALMACENAMIENTO DE LOS DATOS EN LA NUBE

En el presente capítulo se muestra las diferentes configuraciones de las herramientas de software requeridas para hospedar los diferentes controladores en la “nube”, para poder monitorear la velocidad del motor AC monofásico desde cualquier parte del mundo, logrando así acceder al control de este.

#### 4.1 Diagrama general del Software

Figura 4.1 Diagrama general del software



Es un diagrama general del software utilizado en el presente proyecto, elaborado por: Silvia Chuquín. Como se observa en la Figura 4.1 el cliente (Dragino) envía al servidor un POST SOAP, esto significa que pide al servidor los diferentes datos, en este caso el dato que nos interesa es el del ángulo, que sirve para controlar la planta y obtener la velocidad de la misma.

#### 4.2 Configuración del Servidor en la nube

La página SmarterASP es utilizada para albergar los archivos, usualmente tiene un periodo gratuito de 2 meses, se puede renovar cuando el usuario lo desee. Para alojar el servidor en la nube se necesita dicha plataforma ya que se torna necesario adquirir el servicio llamado hosting, consiste en que el proveedor otorga un espacio para los archivos. La configuración se muestra en los anexos.

##### 4.2.1 Adquisición del dominio y carga de los archivos en el servidor

Es preciso tener un dominio para acceder a la página WEB almacenada, así se logra que todos puedan visualizar la página desde cualquier parte que tenga internet. La plataforma SmarterASP.net nos permite conseguir un dominio y realizar la subida de

los archivos en el servidor para acceder a la misma, cada paso se especifica en los anexos.

#### **4.2.2 Direccionamiento para la comunicación cliente-servidor**

En comunicación WLAN para el servidor se determina una dirección IP estática, donde los terminales que se interconectan en el sistema WNCS tienen que estar dentro de la categoría de direcciones IP, esta es una red privada lo que significa que los 3 iniciales octetos son iguales, siendo así: 192.168.0.yyy. En el controlador WAN, la IP del servidor es asignada en el servicio HOST. Esta dirección debe ser establecida para poder enlazar al alias del dominio que es: <http://ine4c.net>

La dirección IP del cliente sigue perteneciendo a la red privada, teniendo una dirección 192.168.0.yyy, esta dirección no aparece en el servidor, es el router encargado de traducir la dirección IP privada en una dirección IP pública, también ejecuta el proceso de identificar el paquete y direccionarlo al cliente.

#### **4.3 Migración del servidor local al servidor en la nube**

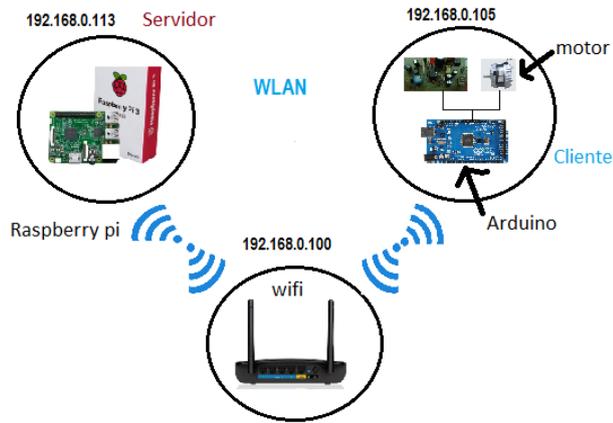
Se denota dos cambios importantes al instante de migrar el servidor local (WLAN) al de la nube (WAN), que se detallan a continuación:

- Uno de los cambios es la dirección IP.
- Para que el cliente pueda dar lectura de los archivos, tiene un obstáculo, que es la seguridad, por esta razón es ineludible añadir la famosa etiqueta <html> al inicio para los archivos pertenecientes de lectura, siendo esto importante porque si no se recibirá los datos en blanco.

#### **4.4 Diferencias del direccionamiento entre Servidor Local y Servidor en la Nube**

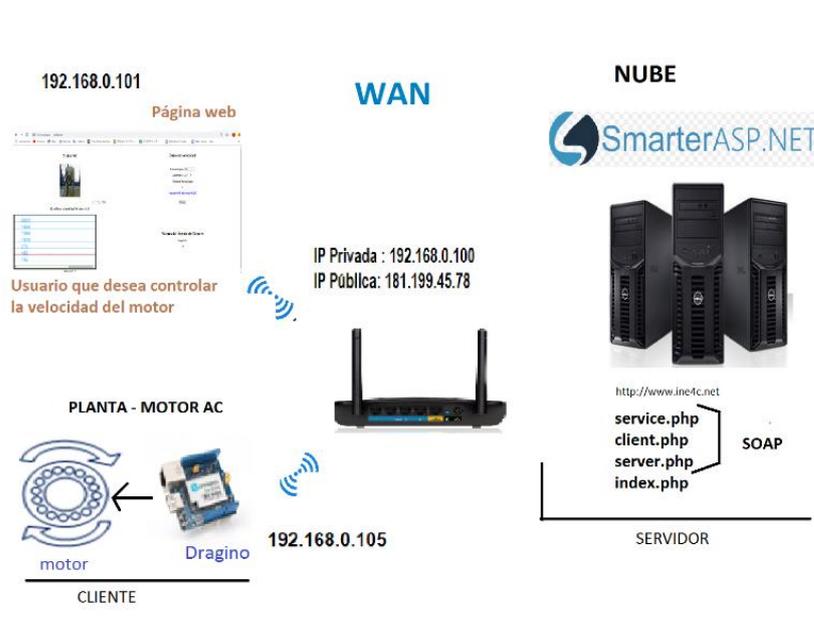
El direccionamiento de los diferentes terminales y servidores que actúan en el sistema de control de red inalámbrica se especifica en la Figura 4.2 (WLAN) y lo que es el servidor en la nube se detalla en la Figura 4.3(WAN).

Figura 4.2 : Direccionamiento Servidor Local (WLAN)



Comunicación Cliente- Servidor WLAN, Silvia Chuquín & Paúl Pachacama

Figura 4.3 : Direccionamiento Servidor en la Nube (WAN)



Comunicación Cliente – Servidor, Silvia Chuquín & Paúl Pachacama

## 4.2 Comunicación en el Servidor

A continuación, se detalla la comunicación cliente-servidor SOAP, con respecto a la parte del servidor se encuentran alojados los diferentes archivos php, tales como: client.php, service.php, server.php y index.php, también se detalla la librería que se utilizó para la comunicación entre el cliente al servidor por medio del protocolo SOAP.

#### 4.2.1 Librería NuSOAP - Servidor

NuSoap es una librería que contiene varias clases que están predeterminadas, estas mismas permite al desarrollador Web efectuar fácilmente los diferentes servicios web (permitiendo la comunicación entre el servidor y un cliente solicitante del mismo) en el lenguaje PHP. Esta librería soporta tanto el desarrollo de los diferentes consumidores, como también el de los servidores (encargados de proveer). NuSOAP está establecido en HTTP 1.0/1.1, WSDL 1.1 y SOAP 1.1. (Brea, 2005)

Una de las ventajas más importantes de esta librería es su sencilla instalación, y su uso no es complejo. También, esta librería no exige módulos complementarios como suele ser otras librerías.

En el presente proyecto se utilizó la Librería NuSOAP para crear los archivos client.php y service.php los cuales sirven para entablar la comunicación entre el servidor y el cliente como se observa en la Figura 4.4.

Figura 4.4 Diagrama del client.php y service.php

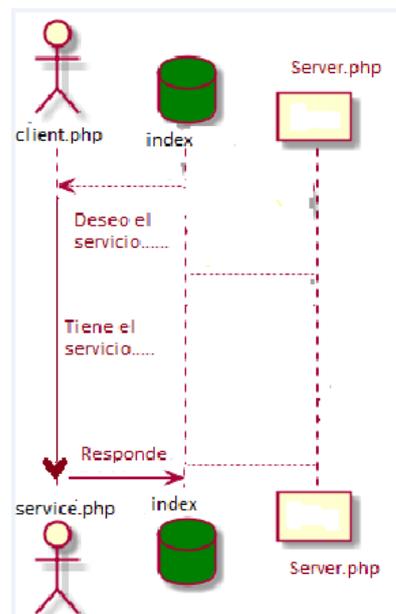


Diagrama de como interactúa el archivo client.php y service.php del servidor SOAP, elaborado por:  
Silvia Chuquín

#### 4.2.2 Service.php - Servidor

Los datos del Setpoint, encoder y ángulo, se encuentran guardados en el archivo llamado enc.json, este archivo es utilizado posteriormente para guardar los mismos, una de las razones por las cuales se utiliza, es para ahorrar recurso en el servidor.

En Service.php se encuentran los diferentes servicios ofrecidos por el servidor y definidos en server.php, como son: Server\_enc, Server-stp y Server\_angulo, definido cada uno a continuación.

- Del archivo enc.json se toma los diferentes datos, en este apartado se toma la parte del encoder, donde al pasar por un proceso es utilizado para la función matemático, como se detalla en la Figura 4.5.

Figura 4.5 Server\_enc



Declaración de la función Server\_enc correspondiente al dato del encoder, elaborado por: Silvia Chuquín & Paúl Pachacama

- En la Figura 4.6 se ve, como es el proceso para la toma de datos del Setpoint del archivo enc.json, para luego ser utilizado en la función matemáticas. En esta dicha función se encuentran los diferentes algoritmos.

Figura 4.6 Server\_setp



Declaración de la función Server\_setp correspondiente al dato del SetPoint, elaborado por: Silvia Chuquín & Paúl Pachacama.

- En la Figura 4.7 se detalla el proceso de obtener el dato del ángulo, desde el archivo enc.json, para luego ser utilizado por la función matemáticas.

Figura 4.7 Server\_angulo

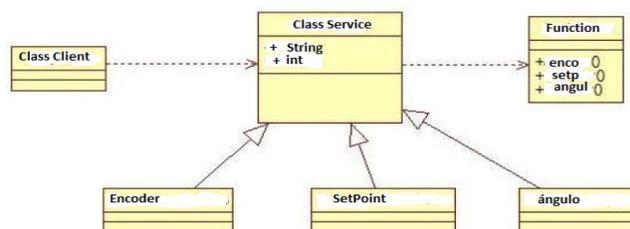


Declaración de la función Server\_angulo correspondiente al dato del ángulo, elaborado por: Silvia Chuquín & Paúl Pachacama.

### 4.2.3 Client.php – Servidor

Se llama a la función correspondiente de la clase server y la respuesta de la función es devuelta al cliente solicitante, como se muestra en la Figura 4.8. En esta parte “soapClient->call “, el cliente llama a la respectiva función enco (), setp () y angul (), que se encuentran en el archivo del Service.php y recibe una respuesta.

Figura 4.8 Función de llamado y respuesta del cliente del servidor



Función de llamado al servicio y respuesta al cliente, elaborado por: Silvia Chuquín & Paúl Pachacama.

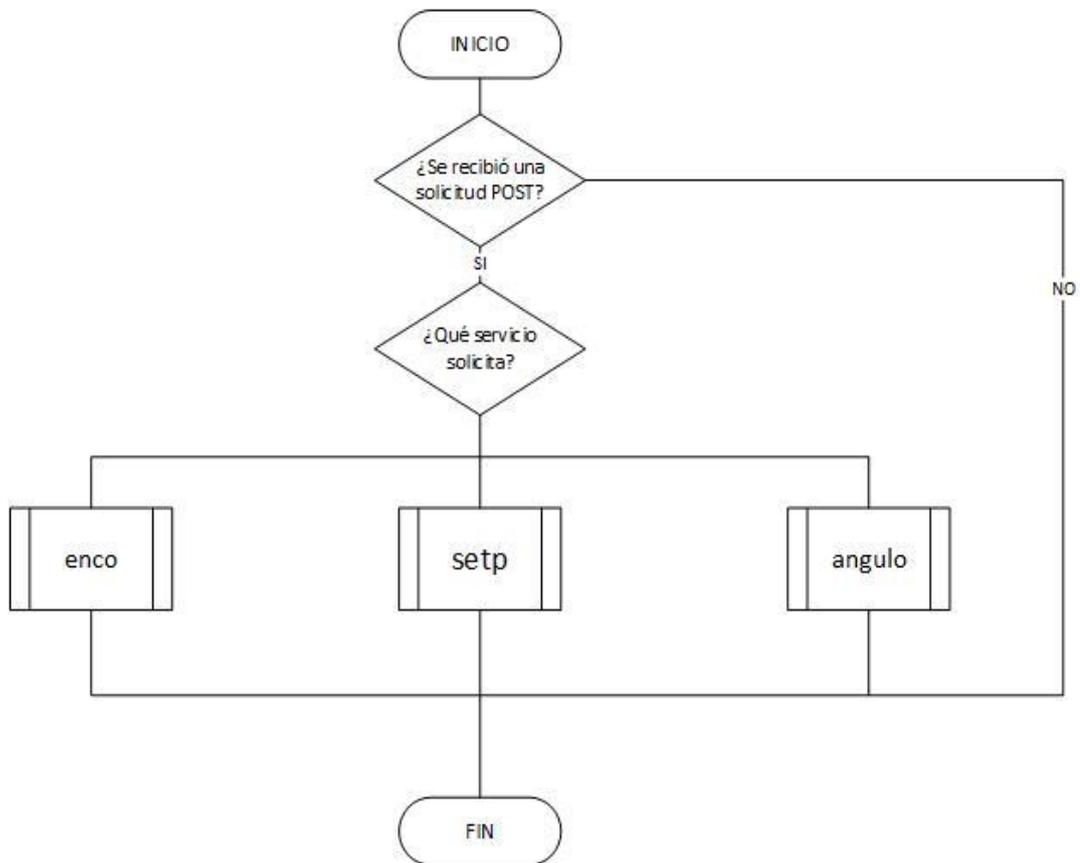
### 4.2.4 Server.php – Servidor

Se declara las funciones a usarse, tales como las variables de entrada y salida a cada función.

### 4.2.5 Índice – Servidor

El índice es la cara del servidor, siendo la parte principal del servidor WEB, es el encargado de mostrar y recibir datos, la cual permite escoger la prestación, como se ve en la Figura 4.9

Figura 4.9 Flujograma de Índice



Flujograma del funcionamiento de Índice de la página WEB, elaborado por: Silvia Chuquín & Paúl Pachacama.

### 4.3 Comunicación en el Cliente

En el siguiente apartado de detalla lo correspondiente al cliente, en este caso del Arduino (Dragino), también se especifica el encabezado que se utilizó para que el servidor responda al cliente SOAP por medio del protocolo SOAP.

#### 4.3.1 Arduino Mega 2560 – Cliente

A continuación, se detalla la parte del cliente, el Arduino mega 2560 necesita del dragino para poderse comunicar inalámbricamente, la programación del cliente se encuentra alojada en el Dragino Yun.

En la Figura 4.10 se visualiza las líneas de comandos para la respectiva llamada al servidor, la cabecera del protocolo SOAP del cliente debe estar bien hecha porque si no el servidor no responderá a la petición del cliente. El método POST se utiliza para enviar el mensaje codificado por medio de HTTP.

La parte correspondiente al protocolo SOAP es “User-Agent” como (NuSOAP) ya que esto lo identifica como cliente SOAP en el servidor.

**Nota:** La librería de Arduino utilizada para enviar la cabecera SOAP por medio de la red WLAN es YunClient, ya que al momento no se ha encontrado una librería específica para SOAP, por esta razón se realiza el envío línea por línea del encabezado por medio de `client.println()`, como se observa en la Figura 4.10.

Figura 4.10 Cliente SOAP

```
Console.println("connected");
client.println("POST /tesis_silvia_paul/ HTTP/1.1");
client.println("Host: 192.168.0.101");
client.println("Connection: keep-alive");
client.println("User-Agent: NuSOAP/0.9.5 (1.123)");
client.println("Content-Type: application/x-www-form-urlencoded");
client.print("Content-Length: ");
client.println(mensaje.length());
client.println("");
client.println(mensaje);
```

Cabecera del protocolo SOAP de un cliente, elaborado por: Silvia Chuquín & Paúl Pachacama

Después de enviar un mensaje POST se recibe una respuesta, la cual es leída y depurada para la correcta interpretación, como se detalla en la Figura 4.11.

Al momento de realizar la lectura de la respuesta, se recibe los diferentes datos como String, se debe realizar la conversión a un dato doble, para ser utilizado posteriormente.

Figura 4.11 Lectura de la respuesta POST del cliente

```
while (cliente.available()) {
  rea=cliente.read();
  if (rea == -1)break;
  lectura += (char)rea;
  angulo = lectura.toDouble();
}
```

Programación de la lectura a la respuesta del cliente, elaborado por: Silvia Chuquín & Paúl Pachacama

#### **4.4 Creación de una base de datos en MySQL**

En el presente proyecto se utilizó MySQL para tener una mejor visión de los datos y respaldos de los mismos. Se sabe que una base de datos nos permite ordenar los mismos de una mejor manera y manejar extensa cantidad de datos. Se especifica la configuración en el Anexo A con el título “Configuración de la Base de Datos en MySQL”.

## CAPÍTULO 5

### PRUEBAS Y RESULTADOS

Finalmente, se muestra las pruebas y resultados conseguidos de los diferentes controladores de velocidad, tanto el del Fuzzy y LQG para la determinación de cual es mejor. Estos controladores se encuentran en la nube, esto quiere decir en una red WAN.

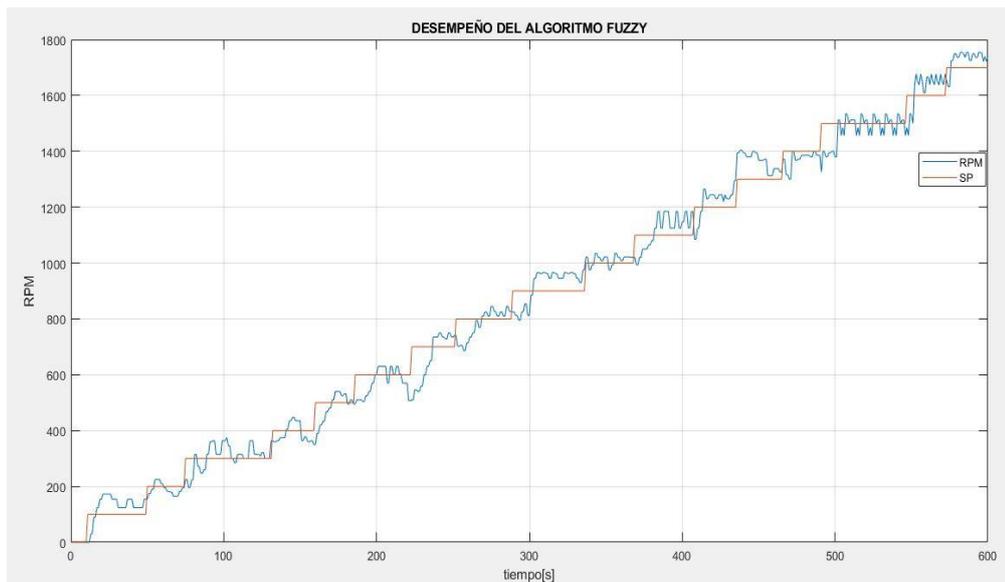
#### 5.1 Funcionamiento del controlador Fuzzy WAN

Para verificar el funcionamiento del controlador Fuzzy se lo realiza con saltos de 100 RPM.

##### 5.1.1 Pruebas y resultados, Funcionamiento Fuzzy WAN

A continuación, se muestra el resultado de la prueba para el algoritmo Fuzzy, tenemos la siguiente respuesta ante los cambios de SP.

Figura 5.1 Prueba algoritmo Fuzzy



Desempeño del algoritmo ante cambios de SP, elaborado en Matlab.

En la Figura 5.1 se puede apreciar que en el SP=1300 se tiene un pico de velocidad, esto se debe a que las curvas de la salida caen en un ángulo más pequeño lo que hace que las RPM aumenten y a el algoritmo le tome más tiempo estabilizar la velocidad.

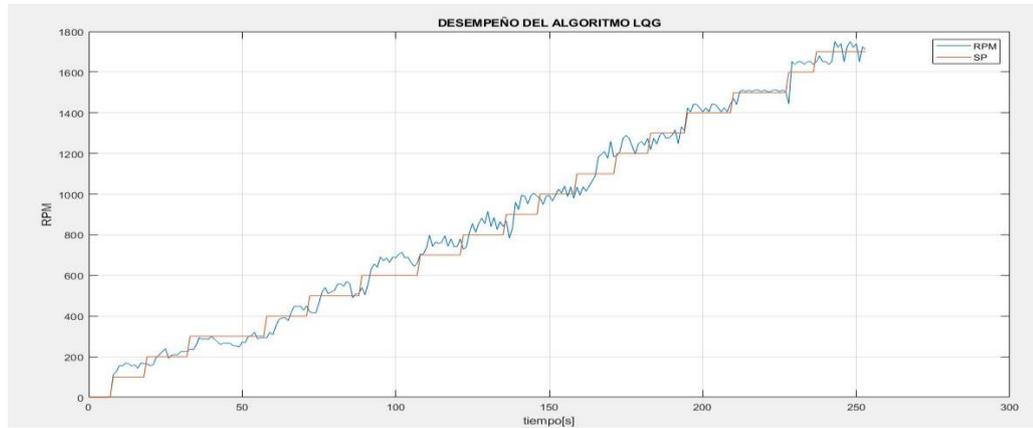
#### 5.2 Funcionamiento del controlador LQG WAN

Para verificar el funcionamiento del controlador LQG se lo realiza con saltos de 100 RPM.

### 5.2.2 Pruebas y resultados, Funcionamiento LQG WAN

A continuación, se muestra el resultado de la prueba para el algoritmo LQG, tenemos la siguiente respuesta ante los cambios de SP.

Figura 5.2 Prueba algoritmo LQG



Desempeño del algoritmo ante cambios de SP, elaborado en Matlab.

En la Figura 5.2 se aprecia un ligero error en  $SP= 500$  y en  $SP= 1100$ , esto se debe a los valores de la matriz  $K$ , esto se debe a que las RPM cambian a una razón distinta por cada grado del ángulo de disparo lo que hace que se las RPM sean más altas.

### 5.3 Comparación de los controladores Fuzzy y LQG WAN

Para comparar el desempeño de los controladores se tomará en consideración el error absoluto y el error máximo con los datos obtenidos en las pruebas de funcionamiento expuestos en los puntos 5.1 y 5.2.

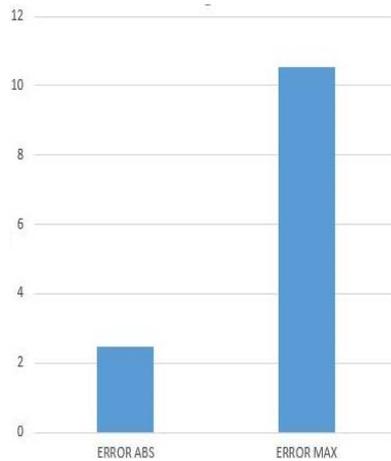
Figura 5.3 Desempeño de los controladores



Desempeño de los controladores a cambios de SP, elaborado en Matlab

Para el algoritmo Fuzzy tenemos una única zona donde el error es considerable y se obtuvo un error máximo de 190 RPM lo que equivale al 10,55%, mientras que el error absoluto que se obtuvo un 2,47% lo que hace que el algoritmo sea eficiente.

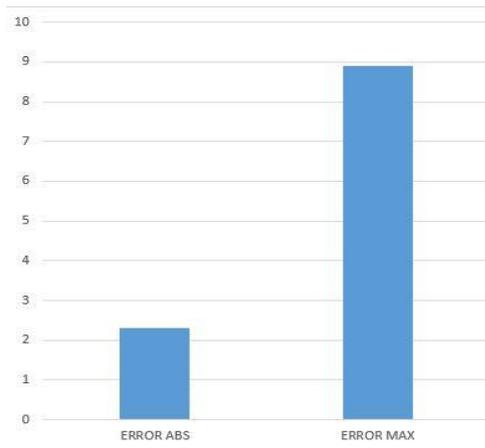
Figura 5.4 Error absoluto y máximo algoritmo Fuzzy



Grafica de los errores obtenidos para el algoritmo Fuzzy, elaborado por: Silvia Chuquín y Paúl Pachacama

Mientras que para el algoritmo LQG tenemos dos zonas donde el error es considerable, pero se obtuvo un error máximo de 41 RPM lo que equivale al 8,89% lo que lo hace más eficiente que el algoritmo Fuzzy, mientras que el error absoluto que se obtuvo un 2,29% lo que hace que el algoritmo sea eficiente.

Figura 5.5 Error absoluto y máximo algoritmo LQG



Grafica de los errores obtenidos para el algoritmo LQG, elaborado por: Silvia Chuquín y Paúl Pachacama

## CONCLUSIONES

Se observó las ventajas y desventajas, al confrontar los diferentes resultados de ambos controladores locales tanto red WLAN y red WAN; en sistemas donde el tiempo no sea una medida crítica, funcionan correctamente, llegando a ser muy rentable. Se mejora la rapidez del control WAN usando servidores locales WLAN, estos servidores envían los datos del servidor a la nube y se tiene monitoreos a partir de cualquier parte, sea desde la casa, trabajo, viajando, lo importante es tener la red más grande llamada Internet.

La principal diferencia entre los controladores utilizados es la obtención de los datos para el diseño del controlador, el controlador Fuzzy es mucho más sencillo ya que solo se necesita una descripción general de la planta para obtener un desempeño aceptable, pero en cuanto al LQG se requiere de un modelamiento matemático más exacto como la función de transferencia y las matrices de estados además de necesitar filtros como las matrices L y K.

El control de velocidad desde cualquier parte del mundo no se ve afectado por el lugar donde se encuentre el operador, ya que las latencias en la red se ven más influenciadas por otros factores, como son el ancho de banda, la capacidad del servidor, etc, que el lugar de procedencia de los datos.

Es importante explicar que si existe latencia, esta se da entre el servidor y el cliente, esto quiere decir que la influencia depende de donde está ubicada la planta, ya que si la planta está ubicada donde la señal Wifi es casi nula, no va existir un buen envío de datos entre el servidor y cliente.

Se concluyó que el controlador LQG es más exacto para este tipo de sistemas ya que cuenta con un error de 2.29% en estado estable además este controlador tiene una reacción más eficiente ante el ruido causado por agentes externos.

## RECOMENDACIONES

Se recomienda que la IP del servidor sea estática, ya que se evitaría estar cambiando, a cada momento las configuraciones del Arduino.

Es importante dar los respectivos permisos de acceso a los archivos para impedir que el cliente tenga problemas con acceder a los mismos del servidor.

JSON permite que el servidor en este caso Raspberry Pi 3 libere espacio en la memoria cada vez que se reciba información nueva.

Se recomienda como trabajo futuro implementar procesos donde intervengan varios sensores y actuadores para indagar el consumo del ancho de banda, también para observar como los demás dispositivos impactan en los diferentes procesos en el respectivo servidor.

Se recomienda el uso de TRIAC en vez de SCR ya que se requieren el doble de componentes, el doble de señales de control tomando en cuenta el uso de una red Snubber para mejorar la conmutación de los elementos, además de optoacopladores para evitar que las componentes de voltaje alterno y voltaje continuo no interfieran entre sí.

Se recomienda que el modelamiento sea lo más exacto posible para el algoritmo LQG ya que influye en gran medida el desempeño del controlador.

Como trabajo futuro se podría recomendar cambiar el uso del control de potencia por ángulo de disparo al uso de inversores para observar el desempeño de los controladores bajo otras condiciones.

## REFERENCIAS

- Alicia Ramos Martín, & Martín, M. J. (2014). *Aplicaciones Web*. España: Paraninfo,SA.
- Angel Cobo, Patricia Gómez, Daniel Pérez, & Rocha, R. (2005). *PHP yMySQL Tecnologías para el desarrollo de aplicaciones web*. España: Diaz de Santos.
- Báez, F. C. (2018). *UF2217-Lenguaje XML*. España: Elearning S.L.
- Beati, H. (2015). *PHP-Creación de páginas Web dinámicas 2a edición*. Buenos Aires - Argentina: Alfaomega.
- Beruvides López, G., Haber Guerra, R., & Castaño Romero, F. (16 de junio de 2016). Sistemas ciberfísicos basados en métodos de inteligencia computacional para la monitorización de procesos de fabricación. Estado actual y líneas futuras. Madrid, España: Universidad Politecnica de Madrid.
- Brea, O. F. (16 de Marzo de 2005). *desarrolloweb.com*. Obtenido de <https://desarrolloweb.com/articulos/1884.php>
- Campos Verdi, K. J. (noviembre de 2016). DISEÑO DE ACOPLÉ MECATRÓNICO PARA. *ARDUINO MEGA 2560 R3*. (L. T. Ojeda, Ed.) Lima, Peru: PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ. Recuperado el 10 de 12 de 2017, de <http://arduino.cl/arduino-mega-2560/>
- Castillo, A. A. (2017). *Curso de Programación Web*. Campus Academy.
- Chandy, J. C. (2010). Desafíos en el diseño de sistemas cyber-físicos. *USBMed*, 71.
- Chapman, S. (2012). *Maquinas Electricas* (5 ed.). New York: The MacGraw-Hill.
- Dominguez, S., Campoy, P., Sebastián, J. M., & Jiménez, A. (2006). *Control en el Espacio de Estado* (2 ed.). Madrid: PEARSON.
- Golden Kumar, B. (2013). An LQR Controller Design Approach For Pitch Axis Stabilisation Of 3-DOF Helicopter System. *International Journal of Scientific & Engineering Research*, 4, 3-4.
- Gourley, D., Totty, B., Sayer, M., Aggarwal, A., & Reddy, S. (2002). *HTTP: The Definitive Guide: The Definitive Guide*. Reilly media.
- Gutiérrez, F. J. (Marzo de 2016). INTEGRACIÓN ENTRE LA RED TRANSACCIONAL COOPERATIVA (RTC) S.A COONECTA Y LA RED INTERBANCARIA DEL ECUADOR, BANRED EN EL SERVICIO DE CAJEROS AUTOMÁTICOS PARA LAS INSTITUCIONES DE LA RED COONECTA. Quito, Pichincha, Ecuador: Universidad del Pacífico.
- Hernandez Bonilla, A., & Vargas Plazas, F. (2013). *Diseño e implementación de una tarjeta de adquisición de datos inalámbrica para el control del helicóptero 2-DOF de Quanser®*. Bogotá, Colombia : Pontificia Universidad Javeriana.

- Hossain., M. S., & Muhammad., G. (2016). Cloud-assisted Industrial Internet of Things (IIoT) - Enabled framework for health monitoring. *Computer Networks*, 1-5.
- Jayaram, A. (14-17 de Diciembre de 2016). El enfoque Lean six sigma para la gestión de la cadena de suministro global utilizando la industria 4.0 y IIoT. Noida, India: IEEE.
- Jorge Gallegos, Luis Ballesteros, Vladimir Juárez, & Sánchez, Á. (2016). Análisis del comportamiento transitorio y en estado estable de un motor monofásico de inducción: Modelado, simulación y verificación. *Tecnológico Nacional México/Instituto Tecnológico de Celaya*, 13.
- Kouro, S., & Musalem, R. (2002). Control Mediante Logiza Difusa. *Universidad Técnica Federico Santa Marñia*, 1-2.
- Kuhnel, C. (2015). *Arduino for the cloud*. Florida, USA: Universal-Publishers.
- Luke Welling, & Thompson, L. (2009). *Desarrollo web con PHP Y MySQL*. ANAYA MULTIMEDIA.
- Millahual, C. P. (2017). *Arduino - De Cero a Experto*. Autónoma de Buenos Aires: Lorena Blanco.
- Pillajo Angos, C. (01 de julio de 2014). CPS Cyber-Physical Systems. *Introducción a Sistemas Cyber Físicos*. Medellin, Colombia: Universidad Pontificia Bolivariana.
- Pillajo, C., & R. H. (2018). *WIRELESS NETWORK CONTROL SYSTEMS De la teoría a la práctica*. Quito: Universitaria Abya-Yala.
- Rashid, M. (2015). *Electronica de potencia* (4 ed.). Mexico DC: PEARSON.
- Raspberry. (2016). *raspberrypi.org*. Obtenido de <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- Reina, G. A., & Vélez, R. M. (2015). TELECONTROL DOMÓTICO DE CASA DE CAMPO. Guayaquil, Guayas, Ecuador.
- Romero Alamán, J. (20 de julio de 2017). *M-flashcarding como recurso de aprendizaje del HTML*. Cataluña, España: Universitat Jaume I.
- Shantanu Bhadoria, & Ramos, R. O. (2017). *Raspberry Pi 3 Home Automation Projects*. Packt Publishing Ltd.
- Sobrevilla, G., Hernández, J., Velasco-Elizondo, P., & Soriano, S. (2017). Aplicando Scrum y Prácticas de Ingeniería de Software para la Mejora Continua del Desarrollo de un Sistema Ciber-Físico-Applying Scrum and Software Engineering Practices to Continu. *ReCIBE*, 1-15.
- The MathWorks, Inc. (1994-2018). *MathWorks*. Obtenido de <https://la.mathworks.com/help/ident/index.html>

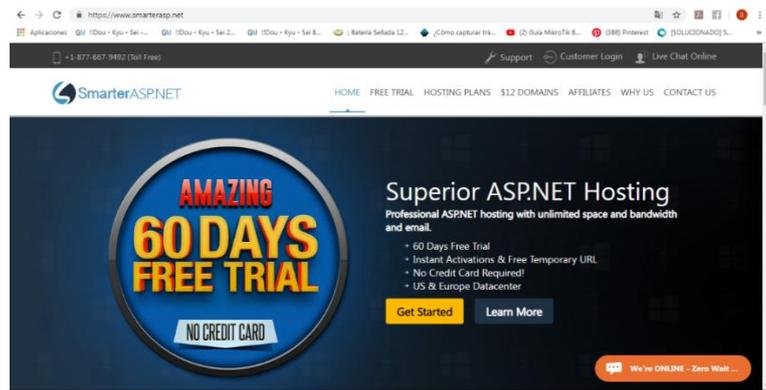
- Wang, G. (2011). Improving Data Transmission in Web Application via the translation between XML and JSON. *2011 Third International Conference on Communications and Mobile Computing*, 4.
- Zambrano, J., & Gonzales, A. (2013). Estados sobre una planta de simulacion desarrollada en MATLAB. *INGENIUS*, 6-7.
- Zhang, D., Wang, Q.-G., & Yu, L. (2017). *Filtering and Control of Wireless Networked Systems*. Suiza: Springer.

## ANEXOS

### ANEXO A: Configuración del Servidor en la nube

- Como primer paso se crea una cuenta en la página web de SmarterASP.net, para esto se da clic en el botón que dice “Get Started”. Como se visualiza en la Figura A.1.

Figura A.1: Página de la plataforma SmarterASP.net



Página de SmarterASP.net, elaborado por: Silvia Chuquín & Paúl Pachacama

- Se procede a llenar los diferentes campos de la cuenta, como se detalla en la Figura A.2, no hay que olvidarse de seleccionar el paquete esperado.

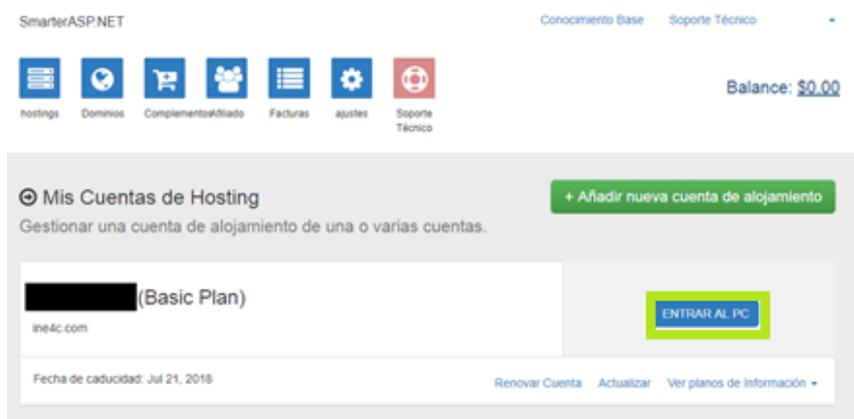
Para que se proceda activar el servicio, se debe ingresar al correo electrónico ingresado en la plataforma para confirmar el correo.

Figura A.2: Registro de datos para la cuenta

Registro de los datos para la creación de la cuenta en SmarterASP.net, elaborado por: Silvia Chuquín & Paúl Pachacama.

- Se da clic en el botón azul que dice “ENTRAR AL PC” para visualizar las diferentes páginas WEB establecidas y dominios inscritos, como se especifica en la Figura A.3.

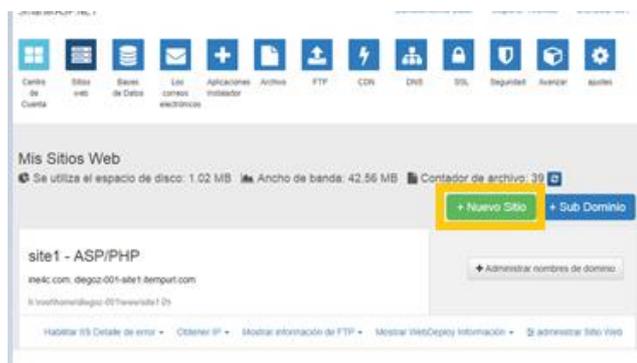
Figura A.3: Página del SmarterASP.net del usuario



Página principal de los hostings, elaborado por: Silvia Chuquín & Paúl Pachacama

- Para configurar al hosting se lo realiza en la página principal del SmarterASP.net del usuario, donde se consigue establecer sitios WEB y organizar las diferentes carpetas como desee el beneficiario, lo podemos visualizar en la Figura A.4.

Figura A.4: Gestión de los diferentes sitios WEB del usuario

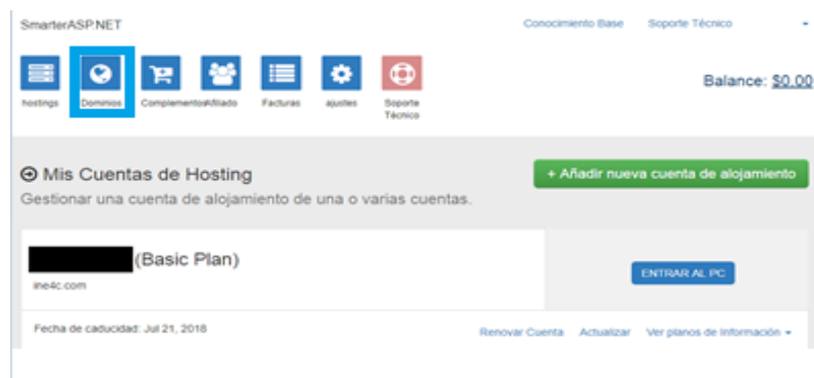


Página de los diferentes sitios WEB, elaborado por: Silvia Chuquín & Paúl Pachacama

### **Obtención de dominio para el servidor y carga de los archivos**

- Como primer paso se debe seleccionar la pestaña de dominios para ingresar, como se detalla en la Figura A.5.

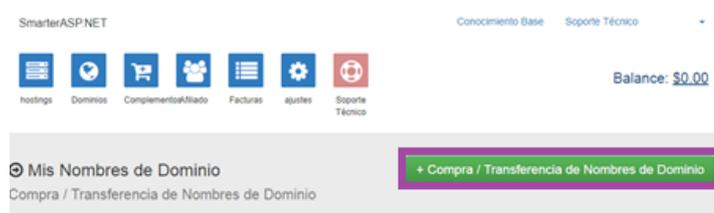
Figura A.5: Opción dominios de la página SmarterASP.net del usuario



Pestaña de la página principal – Dominios, elaborado por: Silvia Chuquín & Paúl Pachacama.

- Como segundo paso se elige la opción Compra/Transferencia de nombres de Dominios, como se demuestra la Figura A.6.

Figura A.6: Botón de Compra/Transferencia de Nombres de Dominios



Botón de la opción Compra/Transferencia de Nombres de Dominios, elaborado por: Silvia Chuquín & Paúl Pachacama.

- La figura A.7 muestra la selección del tipo de dominio y el nombre que se va a colocar para ingresar a la página WEB.

Figura A.7: Nombre y Tipo de Dominio de la página WEB

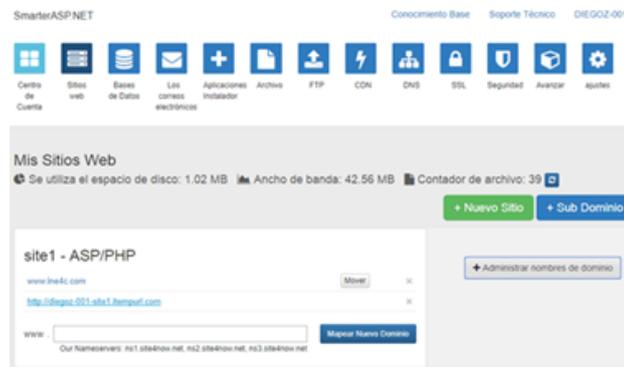


Nombre y Tipo de Dominio de la página WEB, elaborado por: Silvia Chuquín & Paúl Pachacama.

- Una vez hecho el pago del dominio, se procede a retroceder a la página principal de administración. Se relaciona el sitio WEB con el dominio anteriormente comprado, esto se encuentra en la parte de “Administrar

nombres de dominio”, y como último paso se proceder al respectivo mapeo del dominio con el sitio WEB, esta parte se visualiza en la Figura A.8.

Figura A.8 : Sitio WEB asociado al dominio

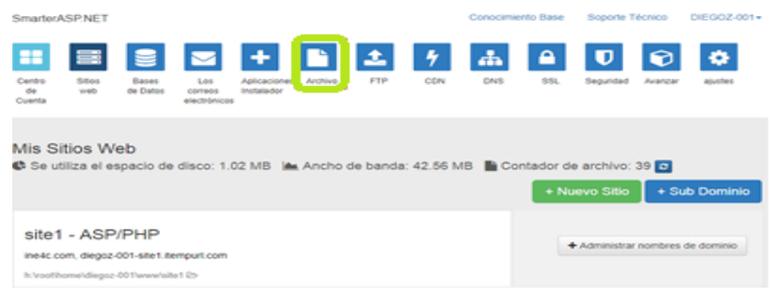


Sitio WEB asociado al dominio, elaborado por: Silvia Chuquín & Paúl Pachacama.

Como otro punto, se procede con la subida de los archivos en el servidor.

- Se da clic en Archivos, esta pestaña se encuentra en la página de administración de SmarterASP.net del usuario, como se detalla en la Figura A.9.

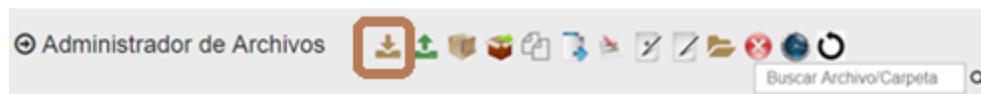
Figura A.9 : Opción para subir los diferentes archivos al Servidor



Pestaña para cargar los diferentes archivos al Servidor, elaborado por: Silvia Chuquín & Paúl Pachacama.

- Como penúltimo paso se da clic en el icono con una flecha hacia abajo para poder subir los archivos como se ve en la Figura A.10.

Figura A.10 : Cuadro del SmarterASP.net para subir los diferentes archivos



Recuadro para la subida de los archivos, elaborado por: Silvia Chuquín & Paúl Pachacama

- Como último paso se abre una página donde se debe seleccionar los archivos de la página WEB y se procede a subir, como lo especifica la Figura A.11

Figura A.11 : Área donde se realiza la importación de los archivos de la página WEB



Área donde se realiza la importación de los archivos de la página WEB, elaborado por: Silvia Chuquín & Paúl Pachacama.

### Configuración de la Base de Datos en MySQL

- Para la creación de una base de datos, primeramente, se debe ingresar usuario y contraseña, como se detalla en la Figura A.12.

Figura A.12 : Pantalla de Login



Pantalla de ingreso a la interfaz, para la creación de la Base de Datos, elaborado por: Silvia Chuquín & Paúl Pachacama.

- Se da clic en nuevo, para proceder a la creación de la Base de Datos, como se muestra en la Figura A.13.

Figura A.13 : Opción “Nueva” para la creación de la Base de Datos



Clic en la opción “Nueva” para crear una nueva Base de Datos, elaborado por: Silvia Chuquín & Paúl Pachacama.

- Se escribe un nombre para la Base de Datos y posterior a eso se da clic en el botón “Crear”, como se ve en la Figura A.14.

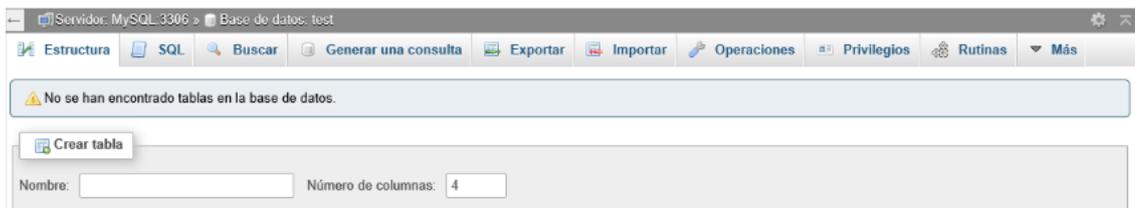
Figura A.14 : Nombre de la Base de Datos



Nombre que llevara la Base de Datos, elaborado por: Silvia Chuquín & Paúl Pachacama.

- En la Figura A.15 se observa 2 casillas, en el primer recuadro se ingresa el nombre que llevara la tabla y en el segundo cuantos campos.

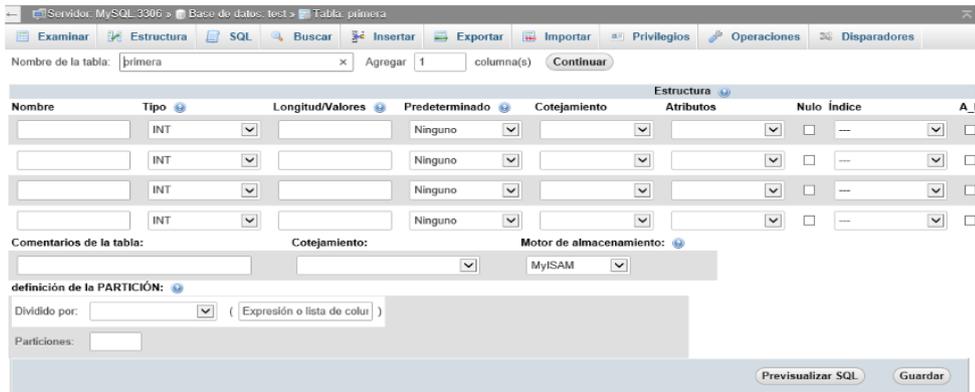
Figura A.15 : Creación de la Tabla.



Nombre de la tabla y cuantos campos llevara la tabla, elaborado por: Silvia Chuquín & Paúl Pachacama.

- En la figura A.16 se detalla la organización de la tabla de la Base de Datos, por ejemplo, se puede seleccionar el tipo de texto que admitirá (INT o TEXT), la longitud de caracteres. Como último paso, se pulsa insertar para agregar los campos en la tabla.

Figura A.16 : Colocación de la Tabla de la Base de Datos



Distribución de la Tabla, elaborado por: Silvia Chuquín & Paúl Pachacama.

## ANEXO B: Datos de la planta

En el siguiente anexo se muestra la tabla con los datos obtenidos del motor.

Tabla B.1 Comportamiento de la planta

ANGULO	VOLTAJE	RPM
175,89	0,97576299	0
173,83	1,79361479	0
174,35	1,57201043	0
174,86	1,36426971	0
174,35	1,57201043	0
173,32	2,02013045	0
174,35	1,57201043	0
173,32	2,02013045	0
172,29	2,50380313	0
170,75	3,28765023	20
171,26	3,02037985	20
171,26	3,02037985	60
170,23	3,56766424	60
170,75	3,28765023	80
170,23	3,56766424	80
169,2	4,14378664	80
169,72	3,84943053	80
167,66	5,05548554	100
167,66	5,05548554	100
167,15	5,37000624	100
168,18	4,74113827	120
166,63	5,69687573	100
164,58	7,04344192	100
164,06	7,39903303	140
164,06	7,39903303	140
163,55	7,75305108	140

163,55	7,75305108	140
160,98	9,61241489	180
160,98	9,61241489	180
160,46	10,0031788	220
159,43	10,790845	220
159,95	10,3909522	240
158,92	11,1873846	260
158,4	11,5960291	280
158,4	11,5960291	280
156,86	12,8310118	300
157,38	12,4099502	340
155,32	14,1012554	340
156,86	12,8310118	360
155,83	13,6768212	400
156,86	12,8310118	420
154,8	14,5377473	440
152,75	16,2937426	460
155,32	14,1012554	500
153,26	15,8517864	520
153,78	15,4045914	540
153,26	15,8517864	540
153,26	15,8517864	520
151,72	17,1961765	560
151,72	17,1961765	580
149,66	19,0385894	600
151,2	17,6566356	660
149,15	19,5020484	720
150,69	18,1113138	760
150,18	18,5689678	800
149,66	19,0385894	800
148,63	19,977451	820
147,6	20,9273584	840
146,58	21,8784086	840
146,06	22,3670639	900
144,52	23,8285511	920
144	24,3266618	980
144,52	23,8285511	1080
145,55	22,8487343	1220
144,52	23,8285511	1280
144	24,3266618	1280
141,95	26,3114621	1300
141,95	26,3114621	1320
142,46	25,8146583	1320
141,95	26,3114621	1320
138,86	29,3601644	1320

137,32	30,9017405	1340
136,29	31,9399702	1340
133,2	35,0842427	1340
132,18	36,1304533	1340
130,63	37,7266793	1360
128,06	40,3868141	1360
127,55	40,9162394	1360
122,4	46,2760665	1380
120,86	47,8791553	1380
114,17	54,8021116	1380
111,6	57,4295037	1400
108,52	60,5433328	1420
105,43	63,6205134	1440
103,89	65,1339458	1440
101,32	67,6261392	1460
98,75	70,0724907	1480
95,15	73,4138414	1530
94,12	74,3500846	1660
90	78	1740
88,97	78,8875212	1740
85,89	81,47834	1740
82,8	83,9781649	1740
80,23	85,9777632	1740
78,69	87,1401477	1740
78,17	87,5264579	1720

Tabla obtenida a partir de variar el voltaje en el motor, elaborado por: Silvia Chuquín y Paúl Pachacama

- A continuación, se muestra en 20 intentos los tiempos obtenidos, al esperar una reconexión al existir una desconexión del servidor.

Figura B.2 Tiempo de Reconexión del cliente al servidor



Tiempos obtenidos al existir una caída del servidor, elaborado por: Silvia Chuquín y Paúl Pachacama  
 Se demora en promedio 17,68 segundos en volverse a conectar el cliente con el servidor cuando este se cae o pierde conexión.