



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO
CARRERA DE INGENIERÍA ELECTRÓNICA**

**IMPLEMENTACIÓN DE UN SISTEMA WNCS MEDIANTE UNA
PLATAFORMA BASADA EN LA NUBE PARA EL CONTROL DE UN
PÉNDULO INVERTIDO.**

Trabajo de titulación previo a la obtención del
Título de Ingeniero Electrónico

AUTOR: Alex David Ñacata Paucar

TUTOR: Carlos Germán Pillajo Angos

Quito-Ecuador

2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Alex David Ñacata Paucar con documento de identificación N° 1720671476 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 15 de marzo del año 2022

Atentamente,



Alex David Ñacata Paucar

1720671476

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Alex David Ñacata Paucar con documento de identificación N° 1720671476, expreso mi voluntad y por medio del presente documento concedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Artículo Académico: “Implementación de un sistema WNCS mediante una plataforma basada en la nube para el control de un péndulo invertido”, el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana

Quito, 15 de marzo del año 2022

Atentamente,



Alex David Ñacata Paucar

1720671476

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Carlos Germán Pillajo Angos con documento de identificación N° 1709255119, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: IMPLEMENTACIÓN DE UN SISTEMA WNCS MEDIANTE UNA PLATAFORMA BASADA EN LA NUBE PARA EL CONTROL DE UN PÉNDULO INVERTIDO, realizado por Alex David Ñacata Paucar con documento de identificación N° 1720671476, obteniendo como resultado final el trabajo de titulación bajo la opción Artículo Académico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 15 de marzo del año 2022

Atentamente,



Ing. Carlos Germán Pillajo Angos PhD.
1709255119

DEDICATORIA

Dedico el presente artículo a mis padres Segundo y Fabiola quienes siempre me han apoyado en todo el trayecto de mi vida que gracias a su apoyo hoy puedo lograr un escalón más en mi vida, gracias por los valores enseñados.

A mis hermanitos Rony y Juan por ser ese apoyo incondicional, gracias por ser unos hermanos buenos, colaboradores y unidos.

A mis abuelitos que en paz descansen Jacinto y Lucila gracias por el apoyo, los consejos y las enseñanzas que me dieron en vida.

Dedico este trabajo a todas las personas que me han apoyado ya sea directa o indirectamente en toda mi etapa estudiantil. *Alex David Ñacata Paucar*

AGRADECIMIENTO

Quiero agradecer primero a Dios por todas las bendiciones recibidas y la oportunidad de poder llegar a terminar mis estudios universitarios.

A mis padres Segundo y Fabiola quienes siempre me han brindado apoyo en todo lo que me propuesto para lograr ser un profesional.

A mis hermanos Rony y Juan por ser los mejores hermanos.

Al Doctor Carlos Pillajo por la ayuda brindada como tutor en el presente proyecto de titulación, por tener esa gran vocación de ser docente y formar buenos profesionales.

Finalmente gracias a la Universidad Politécnica Salesiana donde me forme mis conocimientos y además conocí profesores y amigos quienes han sido una pieza clave en mi formación académica. *Alex David Ñacata Paucar*

Implementación de un sistema WNCS mediante una plataforma basada en la nube para el control de un péndulo invertido.

Ñacata Paucar Alex David
Ingeniería Electrónica
Universidad Politécnica Salesiana
Quito, Ecuador
anacata@est.ups.edu.ec

Pillajo Angos Carlos Germán
Ingeniería Electrónica
Docente Universidad Politécnica Salesiana
Quito, Ecuador
cpillajo@ups.edu.ec

Resumen: El presente artículo detalla la implementación de un sistema Wireless Network Control System mediante la utilización de la plataforma virtual Google Cloud para el control y monitoreo de un péndulo invertido, el cual fue diseñado mediante la utilización de sistemas embebidos y el desarrollo de algoritmos de control para lograr estabilizarlo como son el algoritmo de control PID clásico y un algoritmo de control óptimo estocástico además de la implementación de una arquitectura WNCS para su monitoreo de forma remota, y validar su funcionamiento mediante pruebas experimentales de monitoreo y control a través de la nube.

Palabras Claves— Control LQR, Control PID, NodeMCU ESP2866, Node Red, Péndulo Invertido, WNCS.

Abstract: This article details the implementation of a Wireless Network Control System through the use of the Google Cloud virtual platform for the control and monitoring of an inverted pendulum, which was designed through the use of embedded systems and the development of control algorithms to stabilize it, such as the classic PID control algorithm and an optimal stochastic control algorithm, in addition to the implementation of a WNCS architecture for remote monitoring, and to validate its operation through experimental monitoring and control tests through the cloud.

Keywords— LQR Control, PID Control, Google Cloud, NodeMCU ESP2866, Node Red, Pendulum, WNCS.

I. INTRODUCCIÓN

Las redes de información inalámbricas actuales ofrecen la capacidad de comunicarse a través de largas distancias sin la utilización de cables, por el motivo que la distribución física del cableado en grades distancias puede elevar el costo total de una aplicación en particular y hacerla poco flexible. En la actualidad las comunicaciones inalámbricas son utilizadas en distintas áreas de la electrónica así también en la automatización industrial donde son muy utilizados los sistemas de control en lazo cerrado WNCS (Wireless Network Control System) en los cuales se deben tomar en cuenta los sistemas de red inalámbrica para realizar la transmisión de datos, por la razón que son sistemas de control distribuidos espacialmente, en el cual la comunicación entre el controlador, los sensores y los actuadores se la realiza en una red inalámbrica compartida, siendo una de las principales ventajas de los sistemas WNCS su arquitectura flexible,

permitiendo la disminución del cableado, así como la disminución de costos sin embargo se tiene la desventaja del ancho de banda. [1]

En los últimos años Internet ha tenido un gran avance en el mundo de la computación. Analizando el devenir histórico de la computación, la cual comenzó en paralelo, luego vino la computación distribuida para grid computing, y en los últimos años ha dado un giro hacia la computación en la nube, convirtiéndose en un campo emergente de la informática.[2]

El apareamiento del Internet de las Cosas (IoT) y su contribución con otras tecnologías como el cómputo en la nube ha permitido procesar y analizar datos en tiempo real facilitando la toma de decisiones. [3]

En proyectos anteriores se han realizado sistemas de control de red inalámbrica (WNCS) como es el trabajo “Sistema de control en red inalámbrica (WNCS) implementado al control de voltaje de un generador DC didáctico”. En el cual se desarrolló un sistema de control para un generador de voltaje DC didáctico mediante control de velocidad PWM, en donde el controlador se encuentra alojado en una red local y en un servidor en la nube, utilizando una red WLAN y red WAN así realizando comparaciones de cada uno de los servidores de tiempo de estabilización y latencia.[4]

Asimismo se han realizado trabajos de controladores óptimos como es el trabajo “Diseño e implementación de un controlador óptimo LQG, para un sistema de péndulo invertido aplicado en un equipo Lego Mindstorms ” la cual es la implementación de un controlador óptimo, para un péndulo en donde se realizó una comparación de un algoritmo de control clásico con un algoritmo óptimo en donde se pudo contrastar el rendimiento de cada uno de ellos, además en este trabajo se menciona que en trabajos futuros se puede agregar un control WNCS al péndulo para su control mediante la nube.[5]

El presente artículo se basa en desarrollar un sistema WNCS (Wireless Network Control System) aplicado a un péndulo invertido mediante el estudio de un algoritmo óptimo estocástico el cual mejorara problemas de robustez en el sistema, ahorro de costos por la independencia del cableado dedicado para la comunicación y los costos asociados con ello. Además esta investigación está orientada a

investigadores inmiscuidos en el área de la teoría de control y la telemática.

I. DISEÑO Y CONSTRUCCION DEL PÉNDULO INVERTIDO

El péndulo invertido (robot balancín) diseñado en el presente proyecto es un sistema mecánico y electrónico con dos ruedas en donde sus ejes están ubicados de forma colineal y cada rueda tiene un motor independiente los cuales poseen un encoder incorporado, la estructura del péndulo invertido está sujeta al eje de las ruedas y se balancea alrededor de sí mismo, el centro de gravedad del péndulo esta sobre el eje de giro por lo que es un sistema relativamente inestable y de carácter no lineal.[6]

El péndulo invertido diseñado y ensamblado tiene las dimensiones 16 cm de alto y 25 cm de ancho donde la estructura está formada por cuatro varillas roscadas de 5 mm de diámetro y 10 cm largo. La primera base está diseñada e impresa en 3D en material PLA y sus dimensiones son de 17 cm de largo y 7 cm de ancho esta base cuenta con dos acoples que se encuentran en la parte inferior y sirven para poder sostener los dos motores marca Pololu con una relación de transmisión de 50:1 a 12 VDC de alimentación los cuales tienen un encoder internamente incorporado el cual cuenta las revoluciones por minuto del eje del motor donde están sujetas las ruedas que tienen un ancho de 3,5 cm y un diámetro de 7 cm las cuales permiten que el robot se mantenga estable y pueda desplazarse, además en la parte superior de la primera base se encuentran ubicados de forma espaciada el giroscopio con acelerómetro (MPU6090) y la batería de Lipo de 11.1 VDC a 2200 mAh. La segunda y tercera base están diseñadas en acrílico con dimensiones de 17 cm de largo y 7 cm de ancho, en donde en la segunda base se encuentran ubicados el controlador que es la tarjeta de desarrollo Arduino Mega y el sistema de potencia VNH2SP30 (Monster Moto Shield) y por último en la tercera base se encuentran ubicados el interruptor de encendido y apagado del controlador (Arduino Mega) y el NodeMCU ESP2866 el cual envía y recibe los datos de la nube y tiene la función de antena, también se encuentra contiguo el interruptor del sistema de potencia VNH2SP30, además en la tercera base se encuentra el regulador de voltaje de 12 VDC a 5 VDC con las conexiones de VCC y GND del circuito del péndulo invertido que van a los encoder.

Como un bosquejo inicial el péndulo invertido se diseñó en el software Solidworks en donde se definió las dimensiones de la estructura como las medidas de las tres bases que componen el péndulo, además la longitud y el diámetro de las varillas que lo sujetan así también una simulación aproximada de la ubicación de los componentes electrónicos como son los microcontroladores, sensores, motores, regulador de voltaje, ruedas y batería así obteniendo los valores de las propiedades físicas del balancín para luego ser utilizado en la formulación de las matrices de estados.

En la Figura (1) se muestran los principales componentes del robot balancín péndulo invertido.

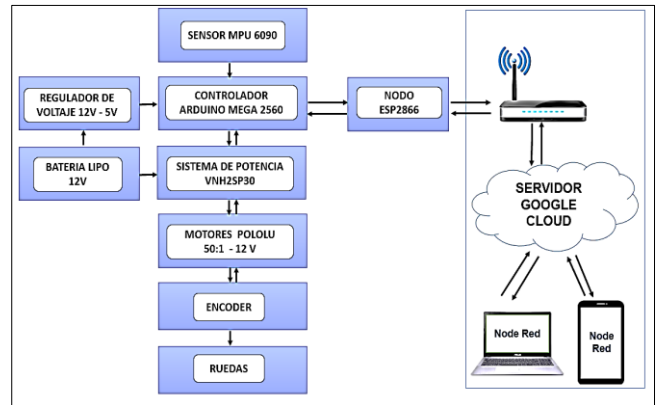


Figura 1. Diagrama de bloques de los componentes del robot balancín.

En la Figura (2) se presenta el robot balancín ensamblado y funcionando en donde se puede apreciar la distribución de los componentes del robot péndulo invertido.

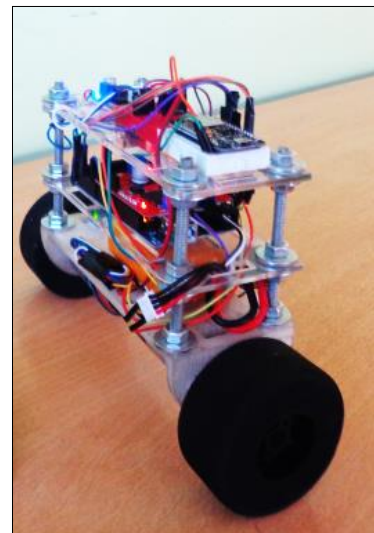


Figura 2. Péndulo invertido construido.

II. DISEÑO DE LA ARQUITECTURA WIRELESS NETWORK CONTROL SYSTEM (WNCS) PARA LA CONEXIÓN DE DISPOSITIVOS IOT Y LA NUBE

El presente numeral detalla el diseño de una arquitectura de red WIRELESS NETWORK CONTROL SYSTEM (WNCS) para la comunicación entre la planta que es el péndulo invertido “robot balancín” y la nube mediante la utilización del software para IoT como es NODE RED, la plataforma GOOGLE CLOUD y hardware como es NodeMCU ESP2866 para el envío y recepción de datos.

A. Wireless Network Control System (WNCS)

Dado que el control WNCS crea un sistema de control en lazo cerrado para la transmisión de datos inalámbricos se utilizó este tipo de arquitectura de comunicación en el presente proyecto para poder controlar y monitorear el péndulo invertido desde la nube hacia la planta y desde la planta hacia la nube en donde se necesita una estructura que envíe y reciba datos inalámbricamente y de forma remota.[1]

B. NodeMCU ESP 2866

El NodeMCU ESP8266 fue utilizado por ser un dispositivo orientado al Internet de las Cosas IoT y además

porque tiene un lenguaje de programación similar a Arduino permitiendo realizar comunicaciones seriales con el controlador Arduino Mega brindando la posibilidad de realizar una conexión inalámbrica a Internet por medio del WIFI, donde hace la función de antena para poder conectarse al servidor alojado en la nube de Google Cloud. [7]

C. Protocolo MQTT

El protocolo MQTT se utilizó para poder realizar la publicación y suscripción de datos de forma inalámbrica ya que se necesitó publicar los datos hacia la nube como son el ángulo y la distancia así como la suscripción de datos como son los controles de giros, paro, setpoint de distancia y setpoint de velocidad del péndulo invertido.[8]

D. Protocolo SSH

El protocolo SSH fue utilizado porque se necesitó conectarse desde un servidor físico hacia un servidor virtual de forma remota y de manera segura por medio de un canal cifrado.

Para el servidor virtual se utilizó la opción de SSH que ofrece la herramienta virtual Machine de la plataforma Google Cloud y para el servidor del usuario se utilizó el software Putty porque trabaja con el sistema operativo de Windows. Cabe recordar que el servidor SSH en el sistema operativo Ubuntu utiliza el puerto 22 por defecto.[9]

E. Protocolo TCP

El protocolo TCP se utilizó para el manejo del software Node Red porque es un protocolo que obliga a establecer una conexión previa entre dos máquinas para poder realizar la transmisión de datos, permitiendo que a través de esta conexión los datos lleguen a su destino de una forma ordenada y segura, además porque es fiable en él envió la información, la información que envía el emisor llega de forma correcta a su destino, de esta manera las aplicaciones que lo utilizan no tienen que preocuparse por la integridad de la comunicación porque se da por hecho que la información que se recibe es correcta.[10]

F. Mosquitto Broker

Se instaló el bróker Mosquitto en el servidor virtual Ubuntu que está alojado en la nube de Google Cloud el cual es un servidor de mensajes (bróker) con código abierto que implementa el protocolo MQTT. El bróker Mosquitto es apropiado y liviano para el uso en todos los dispositivos, desde computadoras de baja potencia hasta servidores completos además el bróker Mosquitto provee de librerías en lenguaje C para poder implementar en clientes MQTT.[11]

G. Node-RED

El software Node-RED se utilizó porque es una herramienta programación visual en la cual se consiguió publicar visualmente los parámetros de rendimiento del péndulo invertido y además porque es una herramienta sencilla de programar, desempeñándose así como un editor de flujo el cual puede funcionar en cualquier navegador donde se puede añadir o eliminar nodos y conectarlos entre sí con el fin de hacer que se comuniquen entre ellos.[12]

H. Google Cloud

La plataforma Google Cloud fue utilizada para poder alojar un servidor virtual en la nube en la cual está instalado el software Node Red se escogió esta herramienta porque brinda

la posibilidad de alojar servidores en la nube con un periodo de prueba gratuito de tres meses abarcando un conjunto de servicios que ofrecen prestaciones muy dispares; desde machine learning hasta Inteligencia artificial pasando por el big data, todo englobado bajo el paraguas del cloud computing. [13]

I. Remote.it

El aplicación Remote.it se utilizó por el motivo que el péndulo invertido es monitoreado y controlado remotamente y esta aplicación brinda la posibilidad del manejo de dispositivos por medio de un control remoto desde cualquier parte del mundo siempre y cuando este instalado en el servidor virtual y estén registrados todos los puertos a utilizar, lo que realizara este software es generar un proxy con el cual el usuario podrá acceder a la ventana de programación de los nodos y al dashboard del software Node Red instalado en el servidor virtual.[14]

J. Comunicación serial entre la tarjeta de desarrollo Arduino Mega y nodeMCU ESP2866.

La planta compuesta por el péndulo invertido (robot balancín) la cual esta enlazada mediante una comunicación serial entre el controlador Arduino Mega 2560 y NodeMCU ESP2866 para luego enlazarse mediante una arquitectura WNCS a un servidor virtual alojado en la nube de Google Cloud.

Mediante la comunicación serial entre los dos sistemas embebidos los cuales son la placa de desarrollo Arduino Mega y el NodeMCU ESP2866, en donde se utilizaron conceptos básicos de comunicaciones seriales entre estos dos dispositivos para así poder establecer un enlace de comunicación entre estos dos elementos para el envío y recepción de datos hacia el servidor que se encuentra en la nube que son los valores medidos de ángulo de inclinación del péndulo invertido, la distancia que recorrida así mismo para la recepción de datos que llegan desde el servidor virtual que se encuentra en la nube hacia la planta los cuales son control de giros horario y giro anti horario, desplazamiento hacia delante, paro y control de velocidad de los motores.

En la Figura (3) se puede apreciar la arquitectura WNCS utilizada en el proyecto.

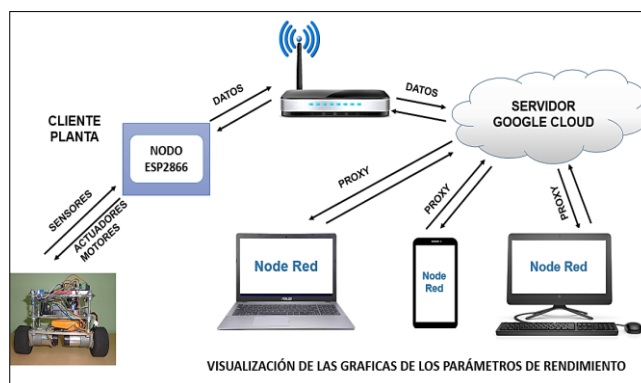


Figura 3. Arquitectura WNCS utilizada para el control del péndulo invertido.

K. Servidor virtual alojado en Google Cloud.

El servidor alojado en la herramienta Google Cloud se utilizó para poder instalar el software Node Red en donde se procedió a programar los nodos que sirvieron para la

visualización de los parámetros de rendimiento que son ángulo y distancia así como los controles de navegación del péndulo invertido y la aplicación Remote it se utilizó para poder generar un acceso remoto y acceder al dashboard desde cualquier parte del mundo siempre y cuando el servidor virtual se encuentre encendido.

Como primer paso para la creación del servidor virtual se estableció una instancia de VM en la herramienta Google Cloud donde se realizó la configuración de la región, la ubicación donde estará alojado el servidor virtual, la zona, la serie y el tipo de máquina a utilizarse.

En la Figura (4) se puede apreciar los parámetros generales de la máquina virtual utilizada.

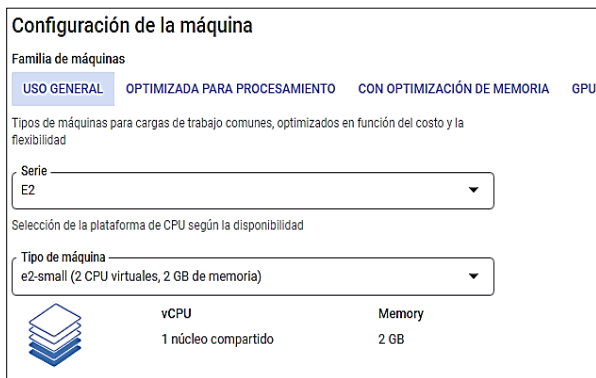


Figura 4. Parámetros generales de la máquina virtual utilizada en Google Cloud.

Después de realizar los pasos descritos anteriormente se procedió a seleccionar el sistema operativo del servidor virtual en el cual se utilizó el sistema operativo Ubuntu versión 18.04 LTS con un disco de arranque SSD persistente de 32 GB de tamaño.

Como último paso de la creación de la instancia virtual se activaron las reglas de firewall para autorizar el tráfico de determinado desde internet.

L. Configuración del Sistema Ubuntu en el servidor virtual.

En el momento que el servidor virtual se encuentra inicializado en la nube de Google Cloud se procede a realizar la conexión SSH con el servidor virtual para poder ingresar los comandos de configuración grafica del servidor virtual Ubuntu después se procede a instalar el software ThightVNC Viewer en el servidor virtual como en el computador del usuario el cual sirve para mostrar de una manera gráfica el servidor virtual Ubuntu. En este paso es necesario crear una contraseña la cual nos permitirá inicializar el servidor ThightVNC cada vez que se utilice el servidor virtual de manera gráfica.

En la Figura (5) se muestra de manera gráfica el servidor virtual Ubuntu alojado en la nube de Google Cloud con la ayuda del software ThightVNC Viewer.

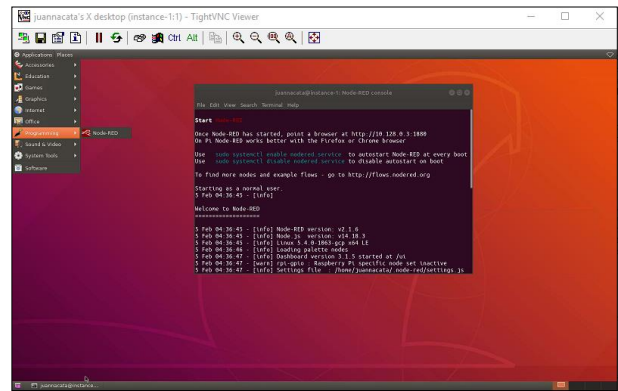


Figura 5. Servidor virtual Ubuntu mostrado mediante la herramienta ThightVNC Viewer.

M. SDK de Google Cloud.

El SDK de Google Cloud sirve principalmente para poder realizar la conexión SSH desde el servidor virtual hacia el servidor del usuario.

Para esto se debe descargar la aplicación SDK de Google e instalarlo en el computador del usuario que se conectara a la nube, luego se procede registrarse en la aplicación con una cuenta de Google y se realiza la configuración en el servidor virtual donde se escoge la cuenta del usuario, el nombre del proyecto con el que se creó el servidor virtual y la zona en la que se creó el servidor virtual.

De esta manera se habilita el funcionamiento del software Putty en el computador del usuario, el mismo que permite la validación de la configuración realizada en la aplicación Google Cloud SDK habilitando la conexión SSH hacia el servidor virtual.

N. Configuración de Remote it.

Para la utilización de la aplicación Remote it es necesario instalarlo en la máquina virtual para luego realizar la su configuración de los protocolos SSH, TCP donde se tendrá que asignar un puerto en la aplicación para cada uno de los protocolos y luego realizar una validación como se puede observar en la Figura (6).

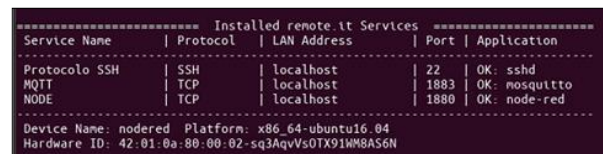


Figura 6. Verificación del funcionamiento de los puertos en el software Remote it.

Una vez instalado el software Remote it y se procede a generar un proxy para los protocolos TCP para la conexión MQTT que es el NodeMCU ESP2866 y un proxy para el software Node Red permitiendo conectarse desde varios servidores remotos siempre y cuando se conozca la dirección del proxy, ofreciendo conexiones seguras en red privada de tipo VPN para así evitar accesos no solicitados ya que elimina la exposición a ataques de puertos abiertos.

En la Figura (7) se puede observar la asignación de la dirección proxy para ser utilizada en la conexión MQTT en este caso será en el NodeMCU ESP8266.

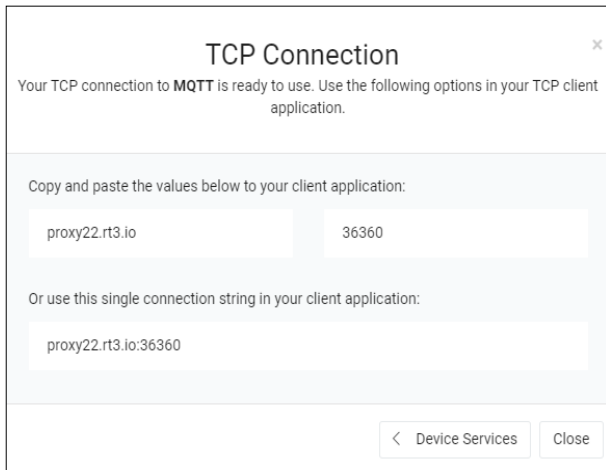


Figura 7. Asignación de la dirección proxy por parte del software Remote it para la conexión MQTT.

O. Funcionamiento del servidor virtual Ubuntu instalado en Google Cloud.

Una vez instalado el servidor virtual en la nube de Google Cloud se procede a acceder a al servidor virtual mediante la generación de un proxy por medio de la aplicación Remote it en donde se ingresa al software Node Red y se configura todos los nodos que se utilizaran el proyecto.

Así mismo se realizaron pruebas de envío y recepción de datos desde el péndulo invertido hacia la nube y desde la nube hacia la plata las cuales no presentaron ningún problema.

En la Figura (8) se muestra el dashboard desarrollado en el software Node Red para el control y monitoreo del péndulo invertido.

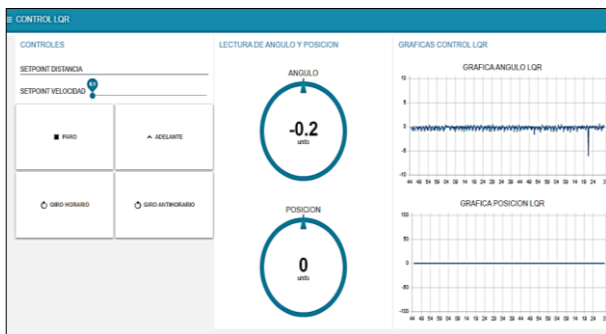


Figura 8. Dashboard diseñado en Node Red para el Control y monitoreo del control LQR.

III. DESARROLLO DE LOS ALGORITMOS DE CONTROL PARA LA ESTABILIZACIÓN DEL PÉNDULO INVERTIDO.

En el diseño del sistema de control del robot balancín se realiza un esquema del control de equilibrio del péndulo invertido, luego se diseñan los algoritmos de control que se utilizaran para este caso se realizó un controlador PID y un controlador optimo estocástico LQR. Como parte final se aplicaron estos dos controladores al péndulo invertido.[6]

A. Modelado matemático del péndulo invertido

Para que el péndulo invertido pueda mantenerse en equilibrio se necesitó obtener el modelado matemático del péndulo y luego se realizó el diseño de los controladores PID y LQR. Como primer paso se obtuvo una ecuación que describa la dinámica completa del motor en el cual se utilizó un diagrama de un motor de corriente continua con una carga aplicada la cual es el toque necesario para mantener el péndulo invertido en equilibrio.[6]

En la Figura (9) se puede apreciar un diagrama de un motor de corriente continua.

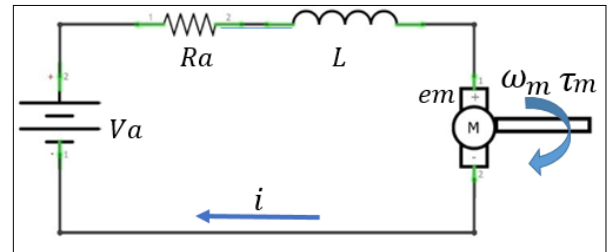


Figura 9. Diagrama de un motor de corriente continua.

Mediante la aplicación de la ley de voltaje de Kirchoff al diagrama del motor de corriente continua de la Figura (9) se obtiene la Ecuación (1).

$$V_a = L \frac{di}{dt} + R_a i + e_m \quad (1)$$

Donde:

V_a : Voltaje aplicado al motor.

R_a : Resistencia del motor

i : Corriente de armadura

L : Inductancia

e_m : Fuerza contraelectromotriz

τ_m : Torque aplicado al motor

La fuerza contra electromotriz está representada por la ecuación (2).

$$e_m = K_e \omega_m \quad (2)$$

Donde:

K_e : Constante electromotriz

ω_m : Velocidad angular del motor

Si se desprecia la inductancia en la Ecuación (1) se obtiene la Ecuación (3).

$$i = \frac{V_a - e_m}{R_a} \quad (3)$$

Por otra parte se obtiene ecuación (4) que es del par aplicado al motor donde K_m es la constante de torque.

$$\tau_m = K_m i \quad (4)$$

Mediante la aplicación de la segunda ley de Newton sobre el eje del motor, en donde se desprecia la fricción se obtiene la ecuación (5) donde τ_a es el torque de fuerza que el motor DC tendrá que vencer para obtener el péndulo invertido en equilibrio.

$$\sum M = \tau_m - \tau_a = I_m \dot{\omega}_m \quad (5)$$

Donde:

I_m : Momento de inercia del motor.

τ_a : Torque de fuerza

Realizando la sustitución de las ecuaciones (2) y (4) en (3) y (5) se obtienen las ecuaciones (6) y (7).

$$i = \frac{Va - K_e \omega_m}{R_a} \quad (6)$$

$$\dot{\omega}_m = \frac{K_m i - \tau_a}{I_m} \quad (7)$$

Donde:

$\dot{\omega}_m$: Aceleración angular del motor.

Además mediante la sustitución de (6) en (7) se obtiene la Ecuación (8) que describe la dinámica completa del motor DC.[6]

$$\dot{\omega}_m = \frac{K_m Va}{I_m R_a} - \frac{K_m K_e \omega_m}{I_m R_a} - \frac{\tau_a}{I_m} \quad (8)$$

Como siguiente paso se procede a conseguir el modelo dinámico de las ruedas en el cual se utiliza un diagrama de cuerpo libre de las ruedas. Teniendo en cuenta que por la simetría de la estructura del robot y la simetría de las ruedas se realiza un supuesto que las fuerzas que actúan sobre las ruedas van a ser iguales.[6]

En la Figura (10) se puede apreciar el diagrama de cuerpo libre de una rueda del péndulo.

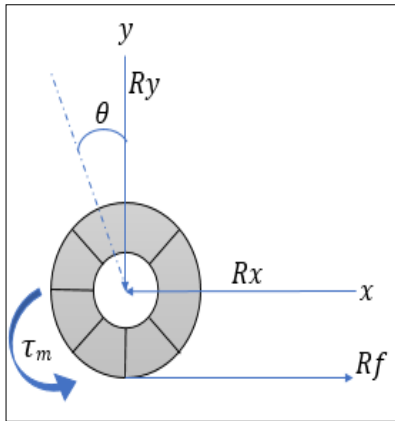


Figura 10. Diagrama de cuerpo libre de una rueda del péndulo invertido.

En donde la Ecuación (9) describe la dinámica del motor como se explicó anteriormente el modelado de la rueda se lo hizo solo para una rueda ya que las dos son simétricas, reciben la misma carga y los motores a los que van ajustadas son de similares características entonces la ecuación se multiplicara por dos para poder obtener el modelado de las dos ruedas.[6]

$$2(M_R + \frac{I_R}{r^2})\ddot{x} = -2\frac{K_m K_e}{R_a r^2}\dot{x} + 2\frac{K_m}{R_a r}Va - 2R_x \quad (9)$$

Donde:

\ddot{x} : Aceleración lineal.

M_R : Masa de la rueda.

I_R : Momento de inercia de la rueda.

R_x : Reacción de la rueda sobre el péndulo.

r : Radio de la rueda.

\dot{x} : Posición.

Como siguiente paso se realizara un diagrama de cuerpo libre del péndulo invertido para así poder representar su dinámica.

En la Figura (11) se puede apreciar el diagrama de cuerpo libre de una rueda del péndulo.

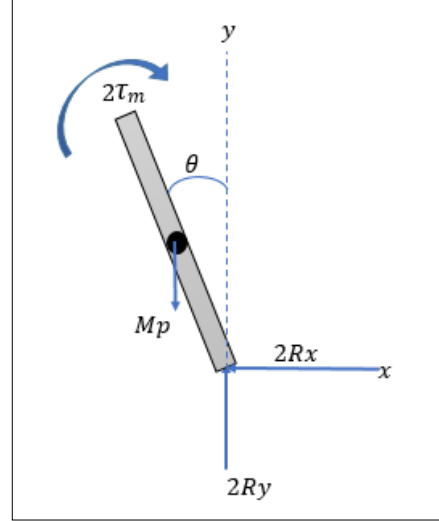


Figura 11. Diagrama de cuerpo libre del péndulo invertido.

En las ecuaciones se demuestra la dinámica del péndulo invertido. En las cuales se puede concluir que son ecuaciones no lineales, por lo que se tuvo que linealizarlas alrededor del punto de equilibrio del péndulo invertido, con la finalidad de utilizar un control que se pueda mantener el péndulo de forma vertical hacia arriba.

Las Ecuaciones (10) y (11) resultan de la linealización alrededor de un punto de equilibrio y están escritas en espacio de estados.[6]

$$\ddot{\phi} = \frac{M_p l}{l_p + M_p l^2} \dot{x} + 2\frac{K_m K_e}{R_a r(l_p + M_p l^2)} \dot{x} - 2\frac{K_m}{R_a(l_p + M_p l^2)} Va + \frac{M_p g l}{l_p + M_p l^2} \phi \quad (10)$$

$$\ddot{x} = \frac{2K_m}{R_a r(2M_R + 2\frac{I_R}{r^2} + M_p)} Va + \frac{2K_m K_e}{R_a r^2(2M_R + 2\frac{I_R}{r^2} + M_p)} \dot{x} + \frac{M_p l}{(2M_R + 2\frac{I_R}{r^2} + M_p)} \ddot{\phi} \quad (11)$$

Donde:

M_p : Masa del péndulo invertido

$\ddot{\phi}$: Aceleración angular.

l_p : Longitud del péndulo.

Además se obtienen las ecuaciones de β y α .

$$\beta = \left(2M_R + 2\frac{I_R}{r^2} + M_p\right) \quad (12)$$

$$\alpha = \left[I_p \beta + 2M_p l^2 \left(M_R + \frac{I_R}{r^2}\right)\right] \quad (13)$$

Finalmente se obtiene la matriz de espacios de estados del sistema.

$$\dot{Z} = AZ + BU \quad (14)$$

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{2k_m K_e (M_p I_r - I_p - M_p l^2)}{R r^2 \alpha} & \frac{M_p^2 g l^2}{\alpha} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2k_m k_e (\tau \beta - M_p l)}{R r^2 \alpha} & \frac{M_p g l \beta}{\alpha} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2k_m (I_p + M_p l^2 - M_p l r)}{R r \alpha} \\ 0 \\ \frac{2k_m (M_p l - r \beta)}{R r \alpha} \end{bmatrix} V_a \quad (15)$$

En donde

$$\dot{z} = [x \dot{x} \phi \dot{\phi}] \text{ y } U = V_a$$

Después de realizar el modelado matemático se procedió a efectuar la simulación en un software, en este caso se utilizó Matlab.[6]

La simulación se realizó en lazo abierto, sin ninguna entrada de control. Esto con el objetivo de verificar si el sistema es inestable, para esto se necesitó modificar su dinámica a través de una entrada de control o la utilización de un control matemático para lograr así el comportamiento deseado del sistema.[6]

Respuesta del sistema para un impulso unitario:

$$U(1) = 1$$

Asignación de condiciones iniciales para observar el comportamiento del sistema.[6]

$$z(0) = [x(0) \dot{x}(0) \phi(0) \dot{\phi}(0)] = [0 \ 0 \ 0 \ 0]^T$$

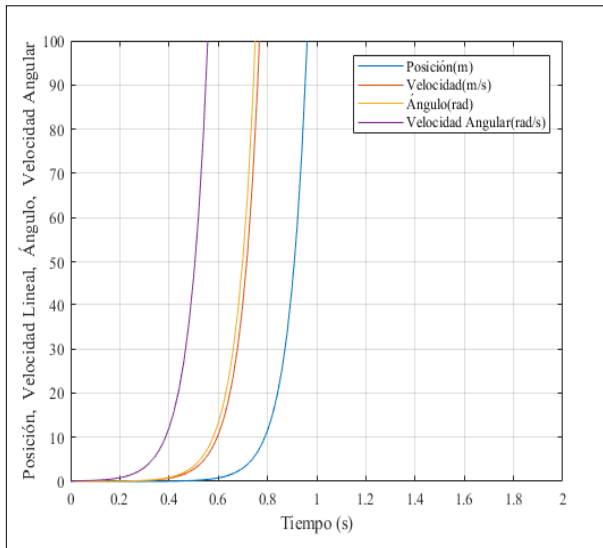


Figura 12. Grafica la simulación de la respuesta del sistema en lazo abierto.

En la Figura (12) se observa que es un sistema inestable porque las gráficas de posición, velocidad lineal, ángulo y velocidad angular del péndulo invertido tienden al infinito. Es por esta razón que es muy importante la utilización de un sistema de control para que el péndulo invertido pueda estabilizarse en un punto deseado.[6]

B. Obtención de los valores de las variables para el cálculo de las matrices de estado en el software Matlab

Resistencia de armadura: Para obtener la resistencia de armadura del motor de corriente continua se realizó la medición de la resistencia directamente de los dos terminales del motor. En esta medición de realizaron 10 muestras para así obtener un valor promedio aproximado de la resistencia de armadura en donde se obtuvo un valor de $R_a = 4,328 \Omega$

Momento de inercia del péndulo invertido I_p : El momento de inercia del péndulo invertido se obtuvo mediante el diseño el software Solidworks, teniendo en cuenta los pesos, las densidades, los materiales y las medidas de los elementos que conforman el robot balancín dando como resultado $I_p = 0.0030126558 \text{ kg} \cdot \text{m}^2$.

Constante de fuerza contra electromotriz: Para la obtención de la constante de fuerza contra electromotriz se realizó de manera experimental utilizando la ecuación (8) donde la velocidad angular en rad/s es ω , el voltaje de entrada es V_a , la resistencia de armadura es R_a y la intensidad es i .

$$K_e = \frac{V_a - R_a i}{\omega} \quad (16)$$

Donde:

ω : Velocidad angular [rad/s]

V_a : Voltaje de entrada [V]

R_a : Resistencia de armadura [Ω]

Se realizaron 12 mediciones para así realizar un promedio de todas las muestras. En las cuales se aumentó el voltaje de entrada conforme a al número de muestras para así obtener un valor más exacto de (k_e) .

Tabla 1. Promedio de la constante de fuerza contra electromotriz de un motor DC utilizado.

Resistencia R_a (Ω)	Voltaje V_a (V)	Corriente I (A)	Velocidad Angular (rad/s)	K_e
4,3	1,5	0,14	17,4882	0,051348
4,3	2	0,17	24,295	0,052232
4,3	3	0,16	38,118	0,060653
4,3	4	0,16	50,056	0,066166
4,3	5	0,16	63,9838	0,067392
4,3	6	0,18	77,9115	0,067076
4,3	7	0,18	91,7345	0,067869
4,3	8	0,18	105,0339	0,068797
4,3	9	0,18	118,438	0,069454
4,3	10	0,19	131,8422	0,069651
4,3	11	0,19	145,7699	0,069686
4,3	12	0,20	158,9646	0,070078
Promedio				0,065033

C. Valores de los parámetros obtenidos para la simulación en el software Matlab.

En la Tabla (2) se muestran los parámetros de las mediciones realizadas para la simulación de las matrices de estado.

Tabla 2. Parámetros de los valores de las mediciones realizadas del péndulo invertido.

Descripción	Símbolo	Valores
Masa del péndulo	Mp	945 gr / 0.945kg
Masa de la rueda	Mr	40 gr / 0,04 kg
Inercia del péndulo	Ip	0.0030126558 kg * m ²
Inercia de la rueda	Ir	0.00002705349 kg * m ²
Radio de la rueda	r	3,55cm / 0.0355 m
Constante del torque	K _m	0,065033 (Nm/A)
Gravedad	g	9,81 m/s ²
Longitud al centro de masa	l	0,0747 m
Constante contra electromotriz	K _e	0,065033 (Vs/rad)
Resistencia devanado del motor	Ra	4,3Ω

D. Control proporcional integral derivativo PID

Se utilizó el control PID, para el controlar la estabilidad del péndulo invertido además porque es un controlador muy utilizado en la industria y ofrece robustez en una amplia condición de operación asimismo por ser un controlador sencillo de entender y operarlo.[15]

Con las matrices de espacio de estados obtenidas anteriormente Ecuación (15) se procede a efectuar el diseño de un controlador de estados por retroalimentación para que el péndulo invertido pueda mantener su estabilidad. [16]

En la Ecuación (17) se puede observar la ecuación de espacio de estados utilizada para la simulación del controlador PID

$$\begin{bmatrix} \dot{x} \\ \dot{\dot{x}} \\ \dot{\phi} \\ \dot{\dot{\phi}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -2.409 & 12.61 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -13.74 & 191 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1.297 \\ 0 \\ 7.396 \end{bmatrix} Va \quad (17)$$

A partir de la función de la ecuación en espacio de estados Ecuación (17) se obtuvo la función de transferencia tomando en cuenta que al momento de realizar la simulación en el software Matlab se obtiene cuatro funciones de transferencia las cuales son de posición, velocidad lineal, ángulo y velocidad angular.

En esta ocasión se utilizara solo la función de transferencia de ángulo ya que con el controlador PID solo controlara estabilidad.

En la Ecuación (18) se muestra la función de transferencia de ángulo la cual fue utilizada para la estabilización del péndulo invertido.

$$G(s) = \frac{7.396 s + 4.436 e^{-15}}{s^3 + 2.409 s^2 - 191 s - 286.9} \quad (18)$$

Para la obtención de las constantes del controlador PID se utilizó el comando pidtune del software Matlab en donde se trabajó con la función de transferencia obtenida en la Ecuación (18) obteniendo así los valores no optimizados de las constantes mostradas en la Tabla (3).

Al momento de implementar los valores de las constantes no optimizados mostrados en la Tabla (3) se constató que estos valores eran muy altos y el péndulo comenzaba a tener perturbaciones y no podía mantenerse estable.

Es por esta razón que se utilizó la herramienta PIDTuner de Matlab la misma que permite ajustar los valores de las constantes K_p, K_i, K_d conforme a la amplitud y tiempo en el cual se estabiliza el péndulo invertido obteniendo así los valores que luego fueron probados en la programación en Arduino.

En la Figura (13) se puede apreciar la simulación de la respuesta del sistema en lazo abierto con impulso unitario.

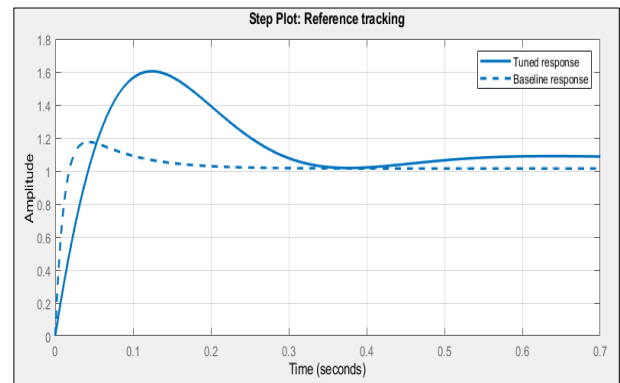


Figura 13. Estabilización del control PID mediante la herramienta PIDTuner.

Cabe resaltar que los valores optimizados mostrados en la Tabla (3) se utilizaron en la estabilización del péndulo invertido, obteniendo una respuesta óptima.

Tabla 3. Valores de las constantes obtenidas del controlador PID.

Constantes	Valor no optimizado	Valor optimizado
K_p	409	91.60
K_i	2.76e+03	511.7
K_d	15.2	4.1

E. Control lineal cuadrado LQR

Para el diseño del controlador lineal cuadrado LQR se utilizó el software Matlab en donde se consideró el sistema continuo en espacio de estados $\dot{Z} = AZ + BU$ como el visto en la Ecuación (10) la cual se puede controlar a través de una matriz apropiada de realimentación de estados K. [6]

$$U(t) = -Kx(t) \quad (19)$$

Y se tuvo que despreciar la función de costo definida mostrada en la Ecuación (20). [6]

$$J = \int_0^{\infty} (X^T \hat{Q}_x + U^T \hat{R}_U) dt \quad (20)$$

En donde Q es una matriz hermítica determinada, con dimensión n x n, en el cual n es el número de variables de estado, y R es una matriz hermítica determinada positiva de dimensión m x m, en donde m es el número de entradas de control, obteniendo así la matriz de ganancias de retroalimentación óptima Ecuación (21). [6]

$$K = R^{-1} B * P \quad (21)$$

En donde P es una matriz hermítica positiva definida. [6]

$$A * P + PA - PB R^{-1} B * P + Q = 0 \quad (22)$$

Para realizar el diseño del control LQR se debe seguir los siguientes pasos, primero se debe resolver la Ecuación (22), la cual es una ecuación matricial reducida de Riccati, para la matriz P, como segundo paso se realiza la sustitución de esta matriz P en la Ecuación (21). La matriz óptima es la matriz resultante K. En la figura 14 se puede apreciar un controlador LQR. [6]

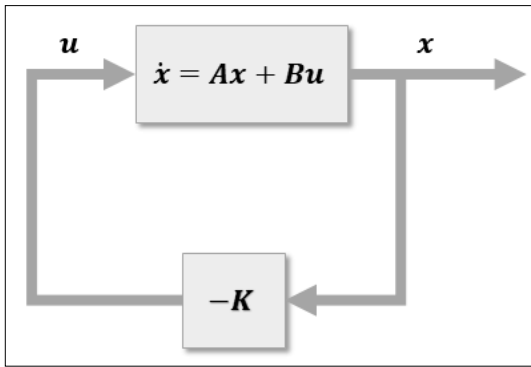


Figura 14. Regulador lineal cuadrático.

En la Tabla (2) se obtuvieron los parámetros de construcción del péndulo invertido los cuales fueron reemplazados en la Ecuación (15) para poder conseguir la matriz de espacio de estados Ecuación (17) con la cual se simuló en el software Matlab el controlador LQR.

Para la implementación del control lineal se verificó que el sistema sea controlable, en otros términos que el rango de controlabilidad F de dimensión nxn sea n. [6]

$$F = [B \ AB \ \dots \ A^{n-1}B] \quad (23)$$

Se realizó la comprobación mediante la utilización del software Matlab, donde se obtuvo una matriz de 4x4 demostrando ser así un sistema completamente controlable. La matriz de Riccati también se resolvió en el software Matlab en donde la matriz de ganancia K es evaluada con diferentes valores de Q y R que son las matrices de ponderación, la matriz Q tiene forma mostrada en la Ecuación (24).

$$Q = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & d \end{bmatrix} \quad (24)$$

De igual manera las variables a, b, c y d son las ganancias de los estados los cuales son \hat{x} (posición), \dot{x} (velocidad lineal), ϕ (ángulo), $\dot{\phi}$ (velocidad angular), mientras que la

ganancia R es una matriz de valor escalar. Finalmente cabe recalcar que los valores de las ganancias utilizados en el presente proyecto fueron las mostradas en la Ecuación (25) y la Ecuación (26).

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (25)$$

$$R = 1 \quad (26)$$

El principal objetivo del control LQR es lograr que los estados se acerquen a cero en un menor tiempo posible.

Para esto es indispensable ir variando los valores de la matriz Q y la matriz R hasta que se obtenga estabilizar al sistema de una forma deseada y el péndulo invertido se mantenga estable de forma vertical y robusto ante las perturbaciones.

En el caso del proyecto el péndulo invertido se estabilizó cuando se utilizaron las ganancias de la Ecuaciones (25) y (26) así obteniendo los resultados de la Figura (15) y la matriz de ganancias K mostrados en la Ecuación (27).

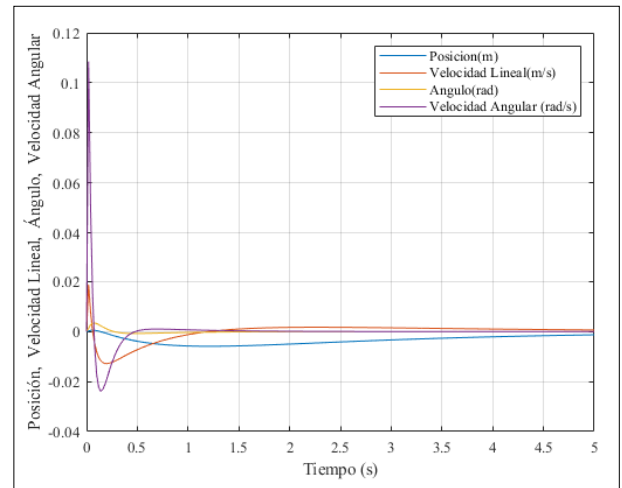


Figura 15. Simulación del control LQR en el software Matlab.

$$K = [-1.000 \quad -4.6439 \quad 60.3602 \quad 4.6610] \quad (27)$$

En la Ecuación (27) se puede apreciar la matriz de ganancias K en donde obtienen cuatro valores los cuales son posición, velocidad lineal, ángulo y velocidad angular que se muestran en la Tabla (4).

Tabla 4. Valores de las constantes de la matriz K del controlador LQR.

Constantes	Valor
Posición	-1.000
Velocidad Lineal	-4.6439
Angulo	60.3602
Velocidad Angular	4.6610

IV. ANÁLISIS Y RESULTADO

En las presentes pruebas se realizaron la implementación del controlador PID y el controlador LQR en la planta (péndulo invertido), para así probar el funcionamiento de los

dos controladores y medir los tiempos de asentamiento y máximo sobreimpulso de los mismos.

A. Prueba de control de estabilidad del péndulo invertido mediante la utilización del control PID y LQR.

Las pruebas de estabilidad de los dos controladores se realizaron al momento de encender el Péndulo Invertido utilizando la referencia de un ángulo de 0 radianes y un tiempo de simulación de 60 segundos.

Mediante la gráfica de la Figura (16) se puede observar el tiempo de asentamiento del péndulo invertido, en donde se evidencia que el control PID se estabiliza más rápido que el control LQR, además el controlador PID es más susceptible ante las perturbaciones en cambio el control LQR es más robusto ante las perturbaciones

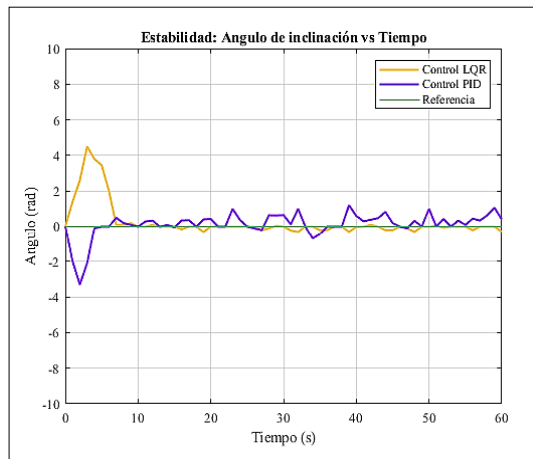


Figura 16. Respuesta del control de estabilidad para los controladores PID y LQR.

En la Tabla (5) se muestra los tiempos de tiempo de asentamiento y máximo sobreimpulso de los dos controladores implementados en el péndulo invertido.

Tabla 5. Tiempo de asentamiento y máximo sobreimpulso

Controlador	Tiempo de Asentamiento (Ts)	Máximo Sobreimpulso (Mp)
LQR	10 segundos	4,5 %
PID	5 segundos	-3,31 %

Por medio de la medición realizada en la Tabla (5) del tiempo de asentamiento y máximo sobre impulso se puede comprobar que existe una diferencia de 5 segundos entre el tiempo de asentamiento del control PID y el control LQR.

B. Prueba del control de un giro de 90 grados del péndulo invertido mediante la utilización del control PID y LQR.

Las pruebas de giro del péndulo invertido con los dos controladores se realizaron cuando el péndulo se mantenía encendido y estable en una sola posición, empleando la referencia de un ángulo de 0 radianes y un ángulo de giro de 90 grados.

A través de gráfica de la Figura (17) se puede observar que el tiempo en el que el péndulo invertido tarda en dar un giro de 90 grados es de 34 segundos además se puede observar que para dar un giro de 90 grados el control LQR tiene más sobre impulso que el controlador PID.

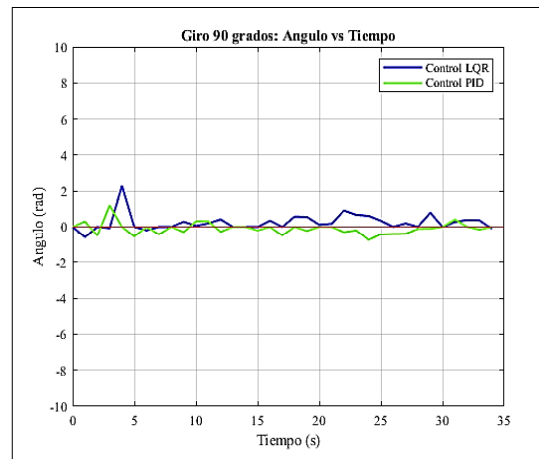


Figura 17. Respuesta del control de estabilidad para los controladores PID y LQR.

En la Tabla (6) se muestra los tiempos de asentamiento y máximo sobreimpulso de los dos controladores implementados en el péndulo invertido.

Tabla 6. Tiempo de asentamiento y máximo sobreimpulso

Controlador	Tiempo de Asentamiento (Ts)	Máximo Sobreimpulso (Mp)
LQR	7 segundos	2,3 %
PID	8 segundos	1,2 %

Por medio de la medición realizada en la tabla 6 del tiempo de asentamiento y máximo sobre impulso se puede comprobar que existe una diferencia de 1 segundo entre el tiempo de asentamiento del control PID y el tiempo de asentamiento del control LQR.

C. Prueba de perturbacion del péndulo invertido mediante la utilización del control PID y LQR.

Estas pruebas se las realizaron cuando el péndulo invertido se mantenía encendido y estable en una sola posición empleando un ángulo de referencia de 0 radianes en un tiempo de 60 segundos.

Mediante la gráfica de la Figura (18) se puede observar que a 44 segundos que el péndulo invertido se encuentra estable se aplica una perturbación y el controlador PID tiende a variar el ángulo de manera fuerte alcanzando un ángulo de inclinación de -2,78 radianes en cambio el controlador LQR alcanza un ángulo de inclinación de -1 radianes siendo este controlador más estable. Es por esta razón que existe una diferencia de -1,78 (rad) llegando a la conclusión que el controlador LQR más robusto ante perturbaciones.

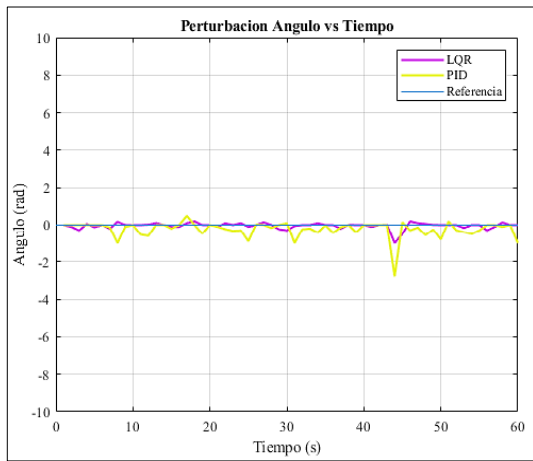


Figura 18. Respuesta del control de perturbación para los controladores PID y LQR.

En la Tabla (7) se muestra los tiempos de asentamiento y el máximo sobreimpulso de los dos controladores implementados en el péndulo invertido.

Tabla 7. Tiempo de asentamiento y máximo sobreimpulso

Controlador	Tiempo de Asentamiento (Ts)	Máximo Sobreimpulso (Mp)
LQR	50 segundos	-1 %
PID	56 segundos	-2,78 %

Por medio de la medición realizada en la Tabla (7) del tiempo de asentamiento y máximo sobreimpulso se puede comprobar que existe una diferencia de 6 segundos entre el tiempo de asentamiento del control PID y el tiempo de asentamiento del control LQR cuando se aplican al péndulo una perturbación.

D. Prueba de desplazamiento con el control PID.

En pruebas anteriormente realizadas se demostró que el control LQR es un controlador robusto para el control del péndulo invertido ante perturbaciones además el controlador LQR permite controlar posición, velocidad lineal, ángulo y velocidad angular en un solo controlador, en cambio el controlador PID permite controlar una variable a la vez y para controlar posición se necesitaría un control PID de posición y otro control PID de velocidad es por esta razón que se realizó un solo controlador de estabilidad y no se logró controlar el desplazamiento como en el controlador LQR

E. Prueba de desplazamiento con el control LQR.

Las prueba de desplazamiento del controlador LQR se realizó empleando un ángulo de referencia de 0 radianes en un tiempo de 40 segundos para el control de una distancia de 50 cm.

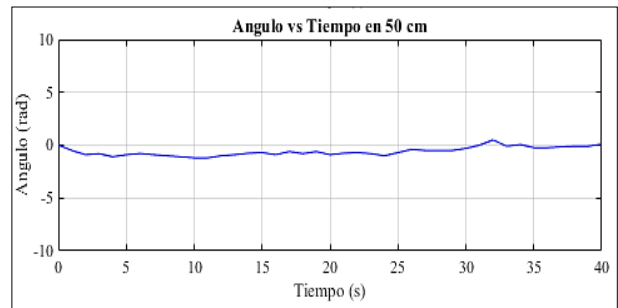


Figura 19. Respuesta del control de desplazamiento para el controlador LQR.

Por medio de la gráfica de la Figura (19) se puede observar la prueba de variación de ángulo en una distancia de 50 cm donde el ángulo tiene una variación negativa tratando de alcanzar un ángulo de -1 radianes al momento que el péndulo invertido comienza a realizar el desplazamiento, en cambio al momento que el péndulo comienza a estabilizarse y llega a la distancia del setpoint el ángulo trata de mantener un ángulo de 0 radianes para estabilizar el péndulo y mantenerlo en una sola posición.

La Figura (20) se puede apreciar la prueba de desplazamiento del péndulo invertido en donde el péndulo invertido se demoró aproximadamente 26 segundos en lograr recorrer una distancia 50 centímetros

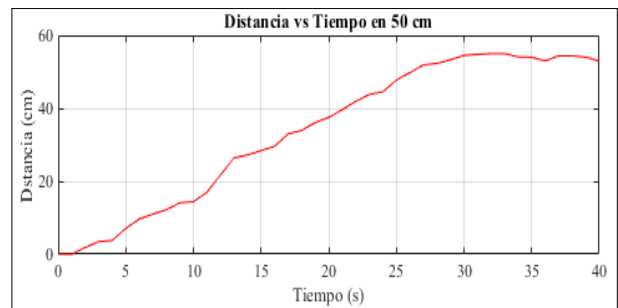


Figura 20. Respuesta del control de desplazamiento para el controlador LQR.

V. CONCLUSIONES

Mediante la investigación de los diferentes tipos de algoritmos de control que se manejan para la estabilización un péndulo invertido, se conoció que los algoritmos más utilizados son el control PID y el control LQR en donde es indispensable realizar el modelado matemático de la planta para poder implementar los controladores.

El diseño del péndulo invertido es muy importante porque es un punto de partida para el modelado matemático, su boceto fue realizado en el software Solidwork definiendo la ubicación de los componentes electrónicos del péndulo así también las dimensiones del péndulo, los tipos de materiales a utilizar, el diseño de las bases y acoples, el tipo de estructura y el tamaño de las ruedas para luego realizar su ensamble en donde es muy importante que los pesos se encuentren equilibrados y los ejes de los motores estén fijos a las ruedas para que el balancín pueda equilibrarse de forma correcta.

La arquitectura Wireles Network Control System (WNCS) fue diseñada con la ayuda de un NodeMCU ESP2866 el cual tiene la función de una antena que envía y recibe datos dese la

nube mediante el protocolo de comunicación MQTT que se enlaza con el servicio de la nube de Google Cloud en donde se encuentra instalado un servidor virtual y dentro del mismo se encuentra funcionando el software Node Red el cual permite el control y monitoreo del péndulo invertido de forma remota.

Por medio del modelado matemático el cual es muy indispensable en el diseño del péndulo invertido se obtuvo la matriz de espacio de estados la misma que sirvió para obtener la función de transferencia para el control PID y la realización del control LQR en donde se controló la estabilidad y distancia recorrida del péndulo invertido.

VI. REFERENCIAS

- [1] C. Pillajo y R. Hincapie, *Wireless Network Control Systems. De la teoría a la práctica*. 2018. Accedido: 22 de marzo de 2021. [En línea]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/17082>
- [2] A. González, Y. García, D. E. Gallego, J. A. Sastoque, y E. Ramírez, «Impacto medioambiental de la integración de la computación en la nube y la Internet de las cosas», *Producción + Limpia*, vol. 11, n.º 2, pp. 22-30, dic. 2016, doi: 10.22507/pml.v11n2a2.
- [3] J. A. Guerrero-Ibáñez *et al.*, «SGreenH-IoT: Plataforma IoT para Agricultura de Precisión», *Revista de Sistemas, Cibernética e Informática*, vol. 14, n.º 2, pp. 53-58, dic. 2017.
- [4] A. López y D. Zapata, «Sistema de control en red inalámbrica (WNCS) implementado al control de voltaje de un generador DC didáctico», jun. 2018, Accedido: 22 de marzo de 2021. [En línea]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/15651>
- [5] K. Solórzano, «Diseño e implementación de un controlador óptimo LQG, para un sistema de péndulo invertido aplicado en un equipo Lego Mindstorms.», jul. 2018, Accedido: 22 de marzo de 2021. [En línea]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/15814>
- [6] A. Duarte, «Control de un Robot Autónomo Tipo Péndulo Invertido». agosto de 2016. [En línea]. Disponible en: <http://posgrado.lapaz.tecnm.mx/uploads/archivos/57ab85cdaefba.pdf>
- [7] J. Valderrama y E. Brea, «ESP8266: Un microcontrolador para el Internet de las Cosas», 2018. [En línea]. Disponible en: <http://www.ing.ucv.ve/jifi2018/documentos/electronica/ERT-006.pdf>
- [8] L. Llamas, «¿Qué es MQTT? Su importancia como protocolo IoT», *Luis Llamas*, 17 de mayo de 2019. <https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/> (accedido 27 de noviembre de 2021).
- [9] «Servidor SSH en Ubuntu», *Adaweb*, 4 de enero de 2020. <https://www.adaweb.es/servidor-ssh-en-ubuntu/> (accedido 7 de febrero de 2022).
- [10] G. Hernández, «Protocolo de Control de Transferencia (TCP)». <https://www.uaeh.edu.mx/scige/boletin/huejutla/n3/r1.html> (accedido 7 de febrero de 2022).
- [11] S. Rodríguez, «Despliegue de clusters de brokers MQTT usando SBCs de bajo coste», *Universitat Politècnica de València*, 2020. Accedido: 8 de febrero de 2022. [En línea]. Disponible en: <https://riunet.upv.es/handle/10251/158889>
- [12] J. Crespo, «Qué es Node-RED», *Aprendiendo Arduino*, 5 de marzo de 2020. <https://aprendiendoarduino.wordpress.com/2020/03/05/que-es-node-red/> (accedido 19 de enero de 2022).
- [13] I. Ordorica, «Qué es y para qué sirve Google Cloud Platform», *Incentro*, 19 de octubre de 2020. <https://www.incentro.com/es-es/blog/stories/que-es-google-cloud-platform/> (accedido 27 de noviembre de 2021).
- [14] E. Ibarra y M. Jurado, «Sistema de control de tráfico vehicular aplicando la arquitectura Fog Computing», mar. 2021, Accedido: 19 de enero de 2022. [En línea]. Disponible en: <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/32676>
- [15] A. Otero, «Diseño, análisis e implementación de un controlador para un péndulo invertido tipo segway», *instname: Universidad de los Andes*, 2020, Accedido: 20 de enero de 2022. [En línea]. Disponible en: <https://repositorio.uniandes.edu.co/handle/1992/45049>
- [16] B. González y A. Bautista, «Diseño e implementación del algoritmo de control para un robot balance, usando fuzzy logic en la plataforma de national instruments.», 2018, Accedido: 20 de enero de 2022. [En línea]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/16009>