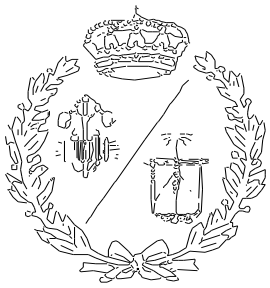
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



Proyecto Fin de Grado

DISEÑO DE SERVOMECANISMO BOLA-PLATO DE DOS GRADOS DE LIBERTAD CON GUIADO VISUAL.

(Two degrees of freedom servomechanism ball & plate design with visual guidance)

Para acceder al Título de

GRADUADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

Autor: Juan Revilla Atienza

06 – 2023

RESUMEN

El presente proyecto se basa en la resolución de un problema de control de la trayectoria de una bola en un entorno completamente controlable el cual usaremos el deslizamiento de una esfera en un plano inclinado para producir el movimiento además se utilizarán varios recursos comerciales de bajo coste que además del control del sistema permitirán su monitorización.

Uno de los objetivos del proyecto es definir todos los bloques necesarios para pasar un movimiento simulado en el ordenador gracias al modelo en variables de estados y las variables físicas a un sistema complejo en el mundo real.

Además, se diseñarán los bloques teóricos desde la adquisición de datos mediante una cámara, hasta el movimiento de la plataforma mediante servomotores en total se usará un controlador que nos trasforme la imagen a distancias y dos microcontroladores que apliquen un control discreto y posicionen los motores en la posición deseada esto añade una parte de comunicación entre varios dispositivos en tiempo real.

The main task of the project is to resolve a ball and plate problem in a controllable environment to move the ball I will use the movement of a sphere in an inclined plane furthermore I will use various commercial low-cost resource that can control and monitor the system.

One of the objectives of the project is to define all the blocks needs to pass a simulated movement from the pc to the reality thanks to the physical variable model and the state model representation.
Moreover, it's needed to design the theoretical blocks from the data acquisition by a camera, to the movement of the platform by servomotors on the frame also we need a controller to transform the image to distance and two microcontroller that apply a discrete control and set the motors in the desired position this add a communication part between multiple devices in real time.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1. INTRODUCCIÓN

En este proyecto se usará Matlab como entorno de simulación física esto me ayudara a decir los diferentes parámetros del sistema como pueden ser el tamaño, el tiempo de paso, o el modelado del sistema con sus variables de estados.

Este proyecto tiene como objetivo poder controlar completamente un sistema que se basa en el movimiento de una bola sobre un plato, el plato se inclinara para que la bola ruede sin deslizamiento de tal forma que podamos controlar la posición de la bola, es importante tener en cuenta que la velocidad angular y su aceleración no es lineal con lo que si queremos un buen control necesitaremos un sistema lineal o una linealización en torno a un punto, en este proyecto se medirá una variable lineal como es la posición del centro en un punto del plato este punto tiene velocidad angular 0 y toda su velocidad es lineal con lo cual podremos medir y controlar esta velocidad.

Para las mediciones se pueden usar distintos sistemas desde acelerómetros hasta matrices de sensores que midan la posición estimada de la bola en el plato por costes, sencillez y resolución usare como sensor una cámara de 5Mpx esto permite al sistema tener mucha precisión aunque sacrificando frecuencia de muestreo del sistema aunque existen formas de mediante software aumentarla no pasaremos de los 100 Hz mientras que con otros sistemas podemos tener frecuencias 10 veces mayores además hará falta del uso de un ordenador que interprete las imágenes y nos de las coordenadas del centro de la bola y otros dos microcontroladores que lleven el control de los actuadores.

El documento se divide en 3 grandes bloques un bloque en el que se analiza el estado del arte de este tipo de sistemas, una simulación de la física del sistema en el que se justifica la elección de las dimensiones del sistema final y un planteamiento de la estructura del sistema, con todo esto añadiré la construcción de un sistema prototipo similar y unos posibles presupuestos para si se desease en el futuro desarrollar una plataforma comercial y no un prototipo.

1.1. Meta

Este proyecto requiere el uso de algunas competencias adquiridas durante el transcurso de todo el grado algunas que tienen que estar cubiertas se nombran a continuación:

- Capacidad de extraer las ecuaciones físicas de un sistema.
- Capacidad de diseñar y acotar un mecanismo para su futuro mecanizado.
- Capacidad de diseñar un sistema de control acorde con lo que se pide.
- Capacidad de extraer las diferentes funciones de transferencia necesarias.
- Capacidad de programar los diferentes microcontroladores necesarios.
- Capacidad de utilizar Python para aplicaciones de visión por computador.
- Capacidad de elegir los actuadores de un sistema.
- Capacidad de simular un sistema.
- Capacidad de entender la electrónica necesaria para el control del sistema.
- Capacidad de redactar un proyecto técnico.

Algunos de los puntos anteriores como la electrónica serán de forma teórica ya que para el diseño se adquieren sistemas con toda la electrónica necesaria implementada de igual manera se añadirá la información para poder diseñar un sistema desde 0 con la posibilidad de diseñar los sistemas electrónicos y así poder tener un producto más barato de fabricar y no un prototipo.

1.2. Alcance

Para la aplicación real del sistema se ha decidido hacer un proyecto propio de bajo coste donde no sea necesario diseñar y probar electrónica compleja y así poder centrar el proyecto en el diseño teórico mientras que la parte práctica será un simple prototipo que demuestre la posible aplicación del sistema para ello se ha optado por:

Construir la estructura en una impresora 3D añadiendo rodamientos a las partes móviles y dejando un marco para introducir un tablero en el cual la bola se mueva, este mecanismo está gobernado por dos servomotores que proporcionan los grados de libertad necesarios para implementar el movimiento de alabeo y cabeceo, estos servomotores estarán controlados cada uno por un microcontrolador ESP-32 se usaran módulos de desarrollo lo cual incrementan el precio del sistema y el uso de dos añade facilidad a la hora de programar aunque diseñando un sistema de comunicaciones más complejo se podría reducir su uso a uno solo ya que cada microcontrolador puede hacer dos tareas a la vez teniendo dos núcleos físicos y un sistema operativo llamado FreeRTOS capaz de gestionarlo, además si cambiamos los controladores a PIC reduciremos el precio pero también eliminaremos el PID discreto obligándonos a implementar el PID en el controlador, como controlador se ha utilizado una raspberry pi placa muy extendida en la industria y perfecta para este tipo de aplicaciones, aunque existen en el mercado soluciones más baratas como ESP-CAM la cual tiene suficiente potencia para interpretar las imágenes y mandárselo al PID , la raspberry pi permite un control mayor ya que tiene un sistema operativo el cual maneja de forma nativa los distintos protocolos de comunicación así como la cámara que se usara para manejar la bola además el uso de la raspberry pi permite experimentar con los algoritmos de Python de OpenCV de otra manera habría que usar C/C++ en todos los programas.

Como bola es necesario algo lo suficientemente grande para que la cámara pueda recoger correctamente el punto central y que tenga la menor masa posible y sea de un color distinguible del fondo para aplicar un filtro de color, al principio se valoró una bola de AIRSOFT, aunque por su mayor accesibilidad y uso se ha optado por usar una pelota de pimpón.

Finalmente, para obtener la trayectoria se programó una página web en la cual dibujas la trayectoria deseada con el ratón y genera un archivo de puntos por los que pasara el centro y la raspberry pi obtendrá esta trayectoria y la implementara gracias a los datos de la cámara y al rápido control de los microcontroladores.

2.ESTADO DEL ARTE

Empezare el proyecto analizando otras soluciones a este mismo problema tanto comerciales como prototipos llevados por otras universidades.

Los sistemas comerciales se utilizan como demostraciones para enseñar los principios de los sistemas de control las compañías venden a los laboratorios de automática un kit con la parte física y un software controlador que contiene todas las librerías necesarias para llevar a cabo el control completo del sistema existen dos compañías principales aunque otros laboratorios dedicados a la investigación tecnológica e industrial como Tekniker también tienen sus sistemas propios, las compañías principales son quanser y national Instruments.

Quanser

Quanser es una compañía canadiense líder en robótica y sistemas de control ofrecen soluciones para la educación y la investigación, más de 2500 universidades utilizan los sistemas de quanser para la enseñanza de los futuros ingenieros en control, ofrecen tanto equipos de laboratorio como software y kits de desarrollo sus soluciones son fáciles de usar además ofrecen documentación y videos educativos que ayudan a entender los sistemas.
Según la propia empresa tienen una plataforma eficiente para validar el problema del péndulo invertido además de unos microcontroladores de tiempo real con alto rendimiento actualmente se están centrando en drones además tiene una plataforma de robots telerobotics ofrecen soluciones en la rehabilitación terapéutica.
Esta empresa ofrece dos soluciones muy similares a nuestro proyecto el primero y más similar es un robot de dos grados de libertad para el balanceo de una bola mediante visión por computador para enseñar como la imagen se distorsiona en los bordes.

Ilustración 1 Sistema Quanser Ball and Plate

El módulo 2 DOF Ball Balancer consta de una placa en la que se puede colocar una bola y se puede mover libremente. Dos unidades base de servo rotativo están conectadas a los lados de la placa mediante dos cardanes que permiten un total de dos grados de libertad. La placa puede girar en cualquier dirección. Al controlar la posición de los engranajes de carga del servo, el ángulo de inclinación de la placa se puede ajustar para equilibrar la bola en la posición plana deseada.
La cámara digital montada en la parte superior captura imágenes bidimensionales del plato y rastrea las coordenadas de la pelota en tiempo real. Las imágenes se transfieren rápidamente a la PC a

través de una conexión FireWire. Los estudiantes pueden hacer que la pelota siga varias trayectorias (un círculo, por ejemplo), o incluso estabilizar la pelota cuando se lanza al plato usando el controlador proporcionado con el experimento.

Quanser ofrece también un sistema similar llamado Ball and Beam el cual únicamente tiene un grado de libertad es ideal para introducirse en los sistemas de lazo cerrado se basa en la rotación de unos servos.

Ilustración 2 Sistema Quanser Ball and Beam

El módulo Ball and Beam consta de una barra de acero en paralelo con una resistencia de alambre bobinado de níquel-cromo que forma la pista sobre la cual la bola de metal puede rodar libremente. La pista es efectivamente un potenciómetro, que emite un voltaje que es proporcional a la posición de la bola.

Cuando se acopla a la unidad base del servo rotatorio, el ángulo de inclinación del haz se puede controlar cambiando el ángulo del engranaje del servo. El módulo Ball and Beam se puede operar en modo independiente, y la posición de la bola se puede controlar a través de la interfaz de usuario.

National Instruments

National instruments es una empresa con sede en Estados Unidos que se dedica a la monitorización de sistema de control precisamente es la empresa que provee a la universidad de las tarjetas de adquisición de datos, pero además de esto se dedican a realizar un software llamado LabVIEW el cual permite simular y controlar sistemas mediante controladores virtuales es capaz de interactuar con otros lenguajes y aplicaciones como CAD o MATLAB.

Gracias a este sistema otra empresa llamada Feedback instruments con sede en el Reino Unido nos vende un sistema ball and plate completo con los módulos de LabVIEW la empresa se centra nuevamente en equipos para la enseñanza en universidades ofrecen productos de gran calidad y que se adjuntan al modelo matemáticos teóricos además añaden recursos educativos y documentación.
El sistema muestra el uso de sistemas de control para estabilizar una bola en el centro incluso lanzándola contra la superficie en 2 segundos permite cambiar los parámetros del PID y seguir la trayectoria de la bola en un sistema no lineal e inestable además incluye una aplicación de Windows para su completo control.

Ilustración 3 Sistema Feedback Instruments Ball and Plate

El Sistema de control de bolas y placas es controlado por NI LabVIEW utilizando una tarjeta de interfaz de NI y demuestra un problema de control clásico de equilibrar una esfera sobre una superficie plana y mantener su posición. Luego se puede programar para que la bola describa una trayectoria circular o de cualquier otra forma alrededor del plato. La actuación de la mesa electromagnética única permite el estudio de este sistema inestable en tiempo real usando controladores sofisticados en NI LabVIEW. La naturaleza progresiva de los ejercicios del alumno permite el estudio del problema desde los primeros principios hasta conceptos de control más avanzados. El producto proporciona una visión útil de la ingeniería de control en todos los niveles de estudios universitarios y permite a los usuarios avanzados modelar y controlar la bola y el plato utilizando su propia estrategia.

Además de empresas que usan equipos de national instruments LabVIEW es un programa con una gran comunidad de ingenieros y estudiantes de ingeniería que han decidido dedicar su tiempo a diseñar un sistema ball and plate o ball and beam, estos diferentes proyectos están recogidos en la página labview.hacksite.io y cada uno utiliza una solución parecida pero con alguna innovación en este documento añadiré la imagen de un proyecto de la universidad de Illinois que resuelve el problema de ball and beam con un motor DC y LabVIEW para el diseño de toda la electrónica.

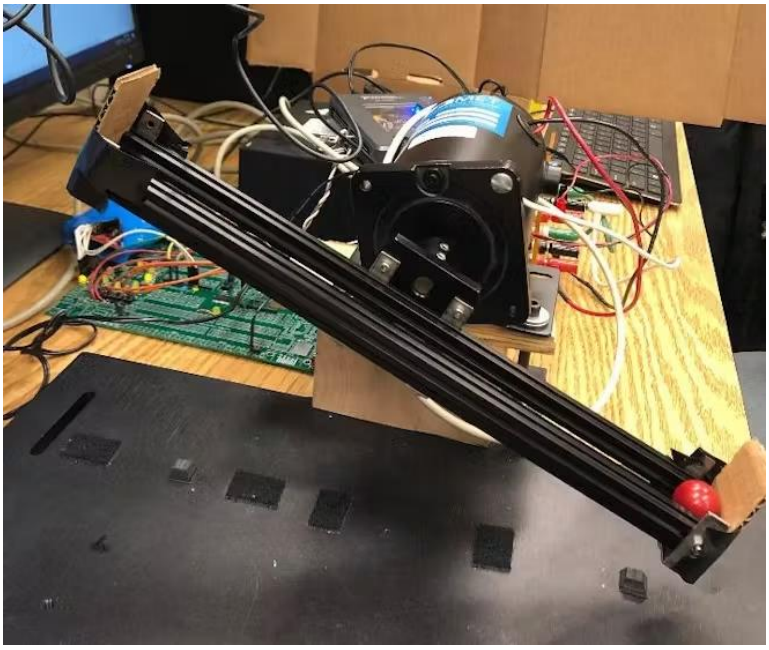


Ilustración 4 Sistema LabVIEW Ball and Beam

Soluciones no comerciales

Además de estas soluciones comerciales o de investigación con software propietario también existen una multitud de soluciones de universidades de todo el mundo como esta que realizan o realizaron la solución a un sistema ball and plate a continuación añadiré 4 ejemplos que se encuentran fácilmente en la biblioteca de otras universidades o en GitHub.

Universidad de Indonesia

El primero de los proyectos públicos que aparecen es llevado a cabo por la universidad suizo-alemana de BSD en indonesia en el utilizan un control PID y para saber la posición de la pelota utilizan un panel táctil que es movido por dos servos por la parte inferior.

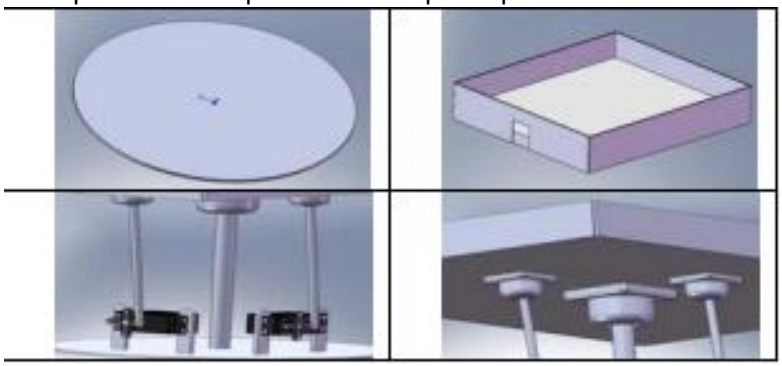


Ilustración 5 Solución Universidad de Indonesia

Universidad de Sevilla

Una solución española a este problema es el llevado a cabo por la universidad de Sevilla con autor José Francisco Quesadas y tutor Daniel Rodríguez la estructura es similar a la anterior, pero cambia el uso de una raspberry pi por el uso de una placa de NVIDIA llamada JTSON tk1, pero la estructura es similar a la anterior.

Ilustración 6 Estructura Universidad de Sevilla

GitHub

EN GitHub también aparecen proyectos similares en este caso utiliza un MSE-450 para equilibrar la bola en el centro de una pantalla táctil gracias a un PID.

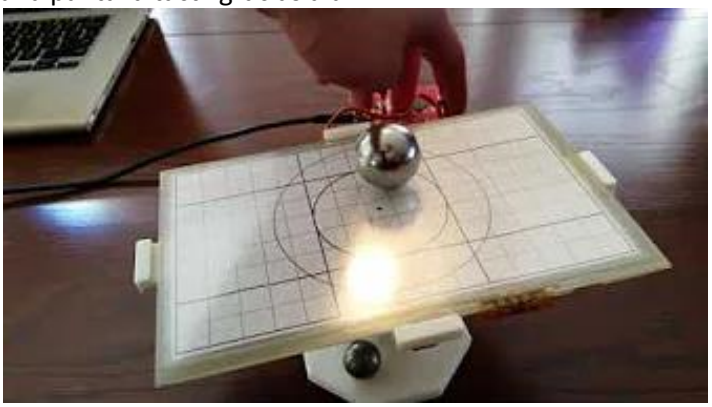


Ilustración 7 Sistema De hallado en GitHub

Proyecto personal

El último proyecto de estado del arte que incluiré esta implementado por David Thomasson el cual no pertenece a ningún proyecto fin de carrera si no a un proyecto personal para controlar un servomotor.

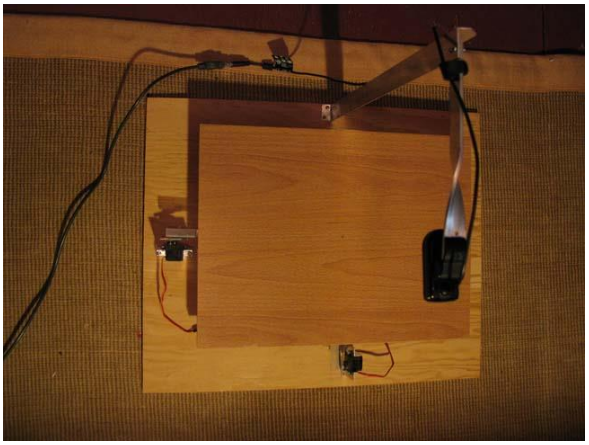


Ilustración 8 Estructura de un proyecto personal

3.DISEÑO MECÁNICO

En este apartado me centrare en resolver el primer problema que nos encontramos al intentar diseñar este tipo de sistemas, la estructura que usaremos, esta tiene que ser una solución creativa funcional, simple y ligera en esta parte se añadirán otras posibles soluciones más utilizas y se detallara con planos cada una de las piezas de la estructura desde el marco hasta los soportes de los servomotores, aunque la solución para unir los servomotores se detallara más adelante.

3.1 Concepto de la estructura

La estructura que se diseñara no es más que un plato completamente plano el cual rotara en los ejes X e Y provocando el movimiento de la bola en los mismos para esto se plantean dos soluciones tener un plato en el aire sujetado por una especie de cruz tridimensional que permita inclinarse el mismo o por una unión cónica con el mismo funcionamiento y dos servomotores en el suelo que conectados en el eje de simetría permitan el movimiento en cada eje respectivamente inclinando el plano.

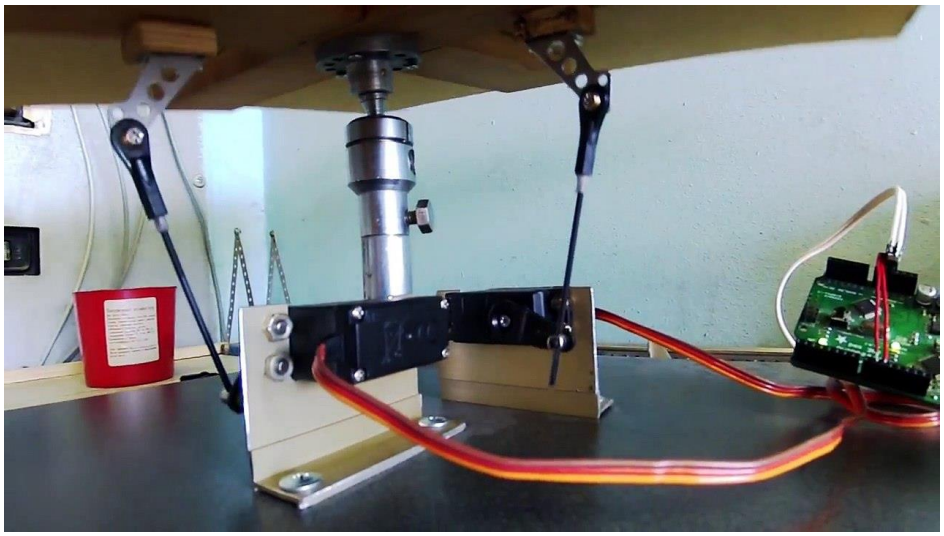


Ilustración 9 Diseño genérico sin marco

Esta solución tiene el problema de que habría que hacer un plato que sujetase otro plato para evitar distorsiones en el terreno, aunque tiene la ventaja de que el plato puede ser tan grande como se desee.

Para el diseño definitivo pensé primeramente en inventos históricos de la antigüedad una de las mayores civilizaciones antiguas y que más experimentaron en el ámbito de la ingeniería fueron los romanos uno de sus inventos fue la balanza romana que se basa en la balanza egipcia original la cual usare como concepto para la estructura.
La balanza romana se sigue usando hoy en los laboratorios cuando queremos realizar mediciones de masa muy precisas sin ningún error más que el de fondo de escala el presente proyecto no tiene como finalidad medir masa con lo cual no es la mejor configuración para empezar.
En cambio, la balanza egipcia se basa en una barra horizontal sujetada por un pilar central con un tornillo que permite el balanceo de la pieza horizontal como vemos es una estructura muy simple, pero permite medir masas y lo que nos interesa generar planos inclinados este tipo de balanzas se denominan balanzas en cruz.

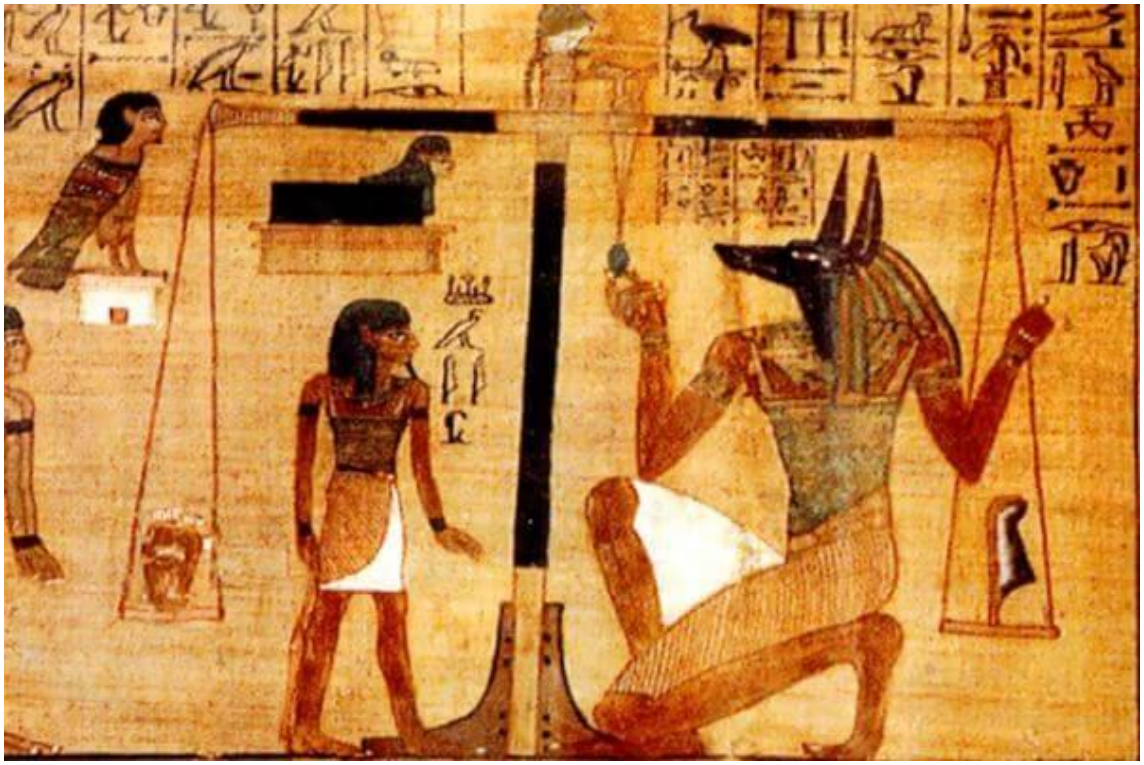


Ilustración 10 Primera aparición histórica de la balanza

Con una simple balanza podemos movernos en un eje generando un mecanismo ball and beam (bola y viga) pero lo que queremos es tener un plato para eso hace falta otra balanza que se encargue del otro eje poniendo las dos balanzas en cruz tendríamos las estructura terminada ahora solo faltaría poner la plataforma encima y aunque esto es posible y muchas soluciones se basan en este principio y funcionan correctamente o incluso unen la balanza por los extremos en mi caso quería una estructura más ágil donde las uniones con la plataforma no sean tan complicadas añadiendo rotulas prismáticas para poder encajar los mecanismos de la balanza en el mismo punto.

Se decidió optar por otra solución más compacta en vez de tener una simple balanza lo que hay es un marco exterior que conectándose a un marco interior se mueve el otro eje al situar el motor en el eje de rotación podemos controlar el ángulo de inclinación de la plataforma sin temor a que se nos rote el motor.

Una solución comercial de estructura parecida es la ofrecida por Gunt Hamburg rt123 que sitúa también dos marcos uno exterior y otro interior que tirando de sus bordes conseguimos mover la bola a la posición deseada.

Ilustración 11 Sistema Ball and Plate comercial con marco

EL problema de este sistema es que tiene una construcción demasiado compleja y no incluye una cámara además no vemos la posición de los actuadores con lo que podría estar dirigido por un sistema que no sea mediante motores.

Otro sistema similar pero realizado por el Kybernetika s.r.o es el de la universidad de Kosice.

Ilustración 12 Sistema Universidad de Kosice

Este sistema es de una construcción más sencilla con los actuadores en el marco y una caja exteriores para realizar su control además incluye una cámara para controlar la posición de la pelota, pero esta pelota esta fija.

Finalmente, el modelo que utilicé para saber que la solución empleada en este proyecto es posible fue la diseñada por Shane Wighton como proyecto personal en el año 2010 a partir de esta estructura muy simple pude definir todos los sistemas mecánicos necesarios.

Ilustración 13 Marco para inclinación de iPhone

Este sistema pese a no controlar una bola física como tal si no un juego del móvil donde inclinabas el móvil y conseguías mover la bola además incluye la cámara de forma que se mueve solidariamente con la estructura además este proyecto es movido mediante varios joysticks.

Estructura elegida

A continuación, se va a mostrar el primer prototipo de la estructura y todas las partes de la versión final los planos con la medida y el detalle de porque se eligieron las medidas se detallarán en el apartado de planos del proyecto.

La primera versión de la estructura modelada se le encontró varios fallos la estructura contenía solamente 4 piezas, el tablero, el marco y los dos soportes le faltaba una forma de introducir la cámara y si lo que hacía era unir directamente el tablero con el marco este no giraría ya que la fuerza de fricción sería demasiado grande además las dimensiones del marco me hacían imposible fabricar las piezas con la impresora 3D.

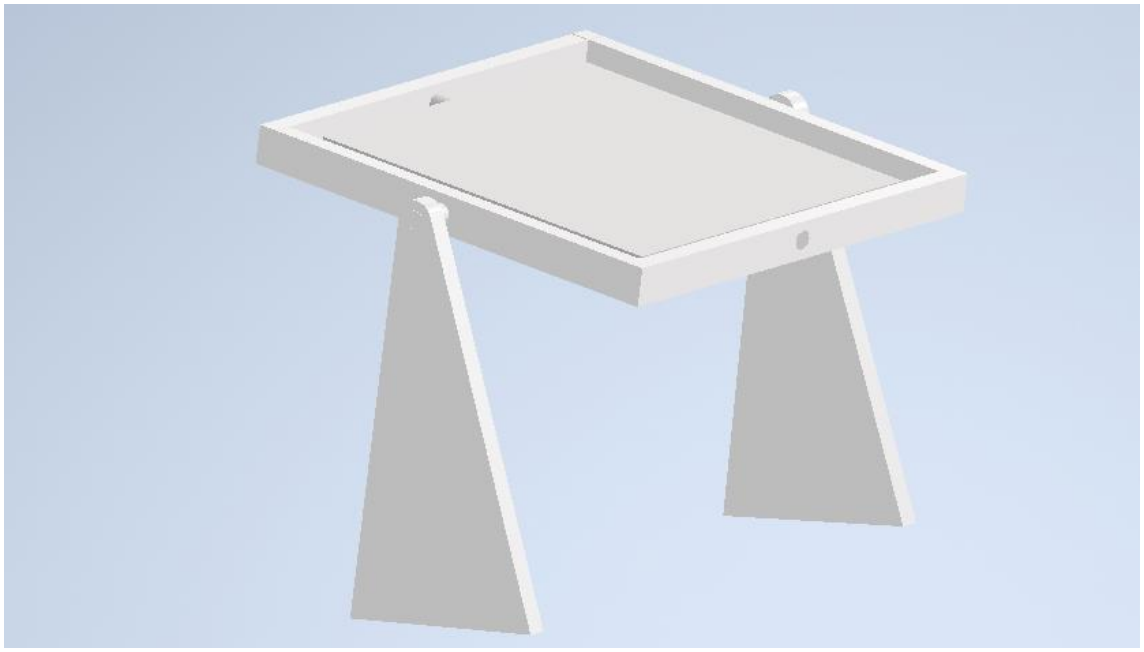


Ilustración 14 Primer prototipo CAD

Se decidió modificar ligeramente la estructura manteniendo el concepto, pero ahorrando materiales y mejorando la misma, lo primero que se hizo fue introducir unos rodamientos comerciales para los ejes giratorios reduciendo la fricción del sistema, estos rodamientos son de referencia R7 y son muy utilizados en ruedas de patinetes o de carritos de la compra con lo cual son accesibles por poco dinero.

Ilustración 15 Rodamientos

El fallo de la cámara se solucionó añadiendo un marco exterior perpendicular hacia arriba en el centro con lo cual no afecta al centro de gravedad en los ejes X e Y solo afecta en el Z el cual no controlamos con lo que mientras este se mantenga por debajo del metacentro el sistema no tendría ningún problema aun así la cámara que se usara pesa solo 10g la altura será aquella que permita a la cámara ver todo el tablero y solamente el tablero esto va a depender de la apertura focal de la cámara y el tamaño del tablero.

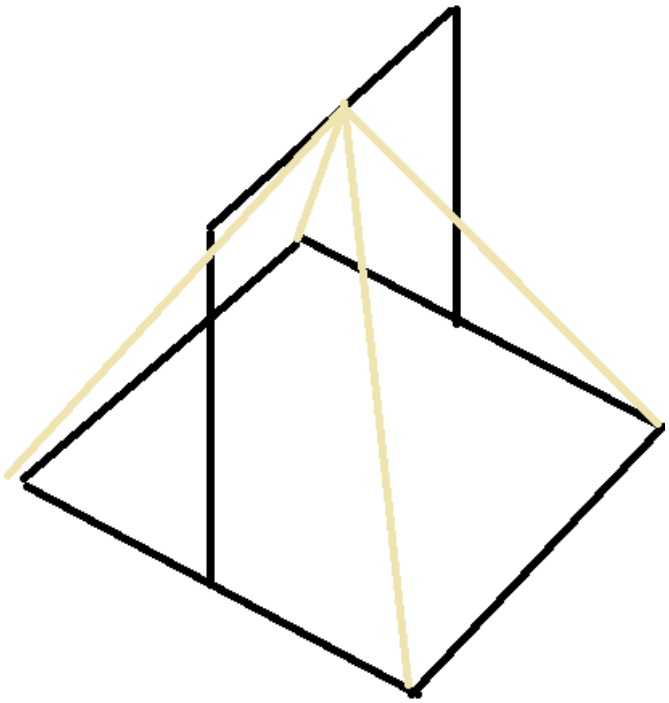


Ilustración 16 Concepto campo de visión de la cámara

Para solucionar el problema de que la estructura no se puede imprimir directamente con la impresora 3D se decidió utilizar una solución modular dividiendo toda la estructura en esquinas y barras que como si fuesen bóvedas romanas encajan entre sí para dar rigidez a la estructura además luego se puede pegar para evitar vibraciones en el sistema también se diseñaron a parte una pirámide de soporte que consigue mantener la estructura en pie, para sujetar la cámara se decidió por combinar unos pilares con cinta aislante que evite la caída de los cables sobre el campo de visión de la cámara, un modelo definitivo podría sustituir la cinta americana por bridas o por unas gomas que eviten que la cámara se caiga.

Diseño de las piezas

En esta parte mostrare todas las piezas que se han diseñado he impreso para el correcto funcionamiento del sistema más adelante del proyecto se justificarán los valores de las dimensiones y se añadirán los planos.

Torre

El primero de los elementos es los soportes verticales este sistema se compone por dos partes la primera una pirámide truncada de base cuadrada que al aumentar el área en el que se apoya la estructura conseguirá mantenerse erguida la otra parte es un soporte para el rodamiento con un marco exterior en el que quedara encajado a presión el rodamiento, en este elemento se tendrá en cuenta la altura de la estructura intentando esta ser la mínima para que no toque en el suele y el tamaño del rodamiento para que la parte del soporte no se rompa además necesitaremos un rectángulo inferior lo bastante grande para que la estructura sea estable

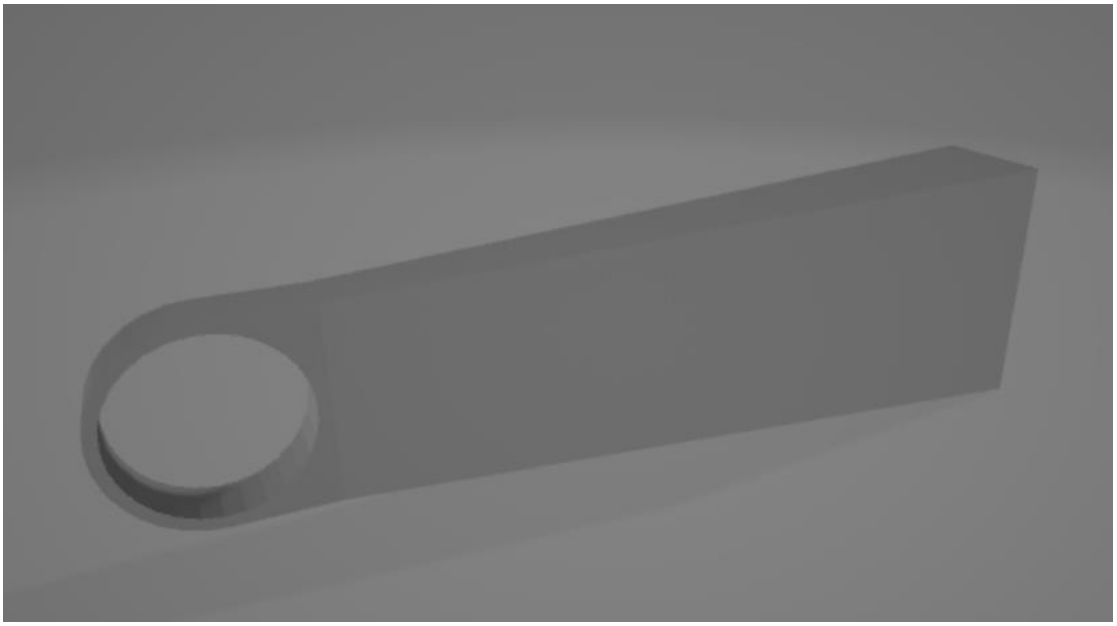


Ilustración 17 Soporte torre

Esquina

Esta es la pieza que más veces se usa en este proyecto y a su vez la más rígida es la esquina básicamente son dos barras de 5 cm a 90º donde en una tiene un cubo saliente que encaja en el negativo de la otra barra este sistema es muy resistente y es lo que da rigidez a la estructura si se hubiese decidido montarlo por vigas y columnas el sistema hubiese perdido rigidez y además sería más difícil de fabricar ya que requiere de barras más grandes y piezas más complejas.

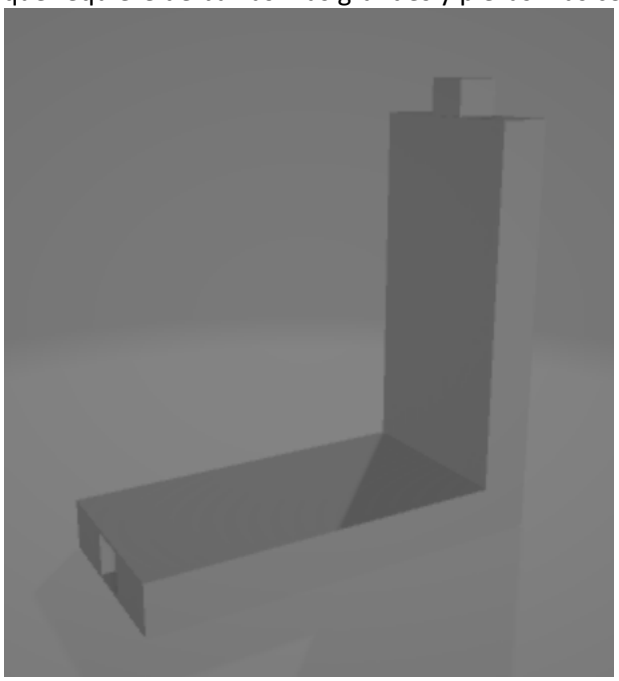


Ilustración 18 Esquina

Rodamiento externo

Para situar el rodamiento externo se diseñó una barra con un agujero en medio del tamaño del rodamiento donde iría encajado el mismo lo que permitiría la libre rotación del eje interno este rodamiento es el que decide la altura de todo el marco además a esta barra se le añadirá en un extremo un hueco de un cubo y en el otro un cubo que encajaran en una esquina por banda al ser

completamente simétrica si se le da la vuelta tiene el mismo efecto con lo cual solo necesitaremos un diseño e imprimiré dos.

Ilustración 19 Soporte rodamiento

Eje externo

El marco externo gira mediante unas barras que salen para incrustarse en la parte interna de los rodamientos, para dirigir el giro del marco se le saca una barra perpendicular que con una barra se conecte a un servomotor externo, además esta barra añade de nuevo el sistema de conexión de un cubo por un lado y un hueco con la forma del cubo por el otro.

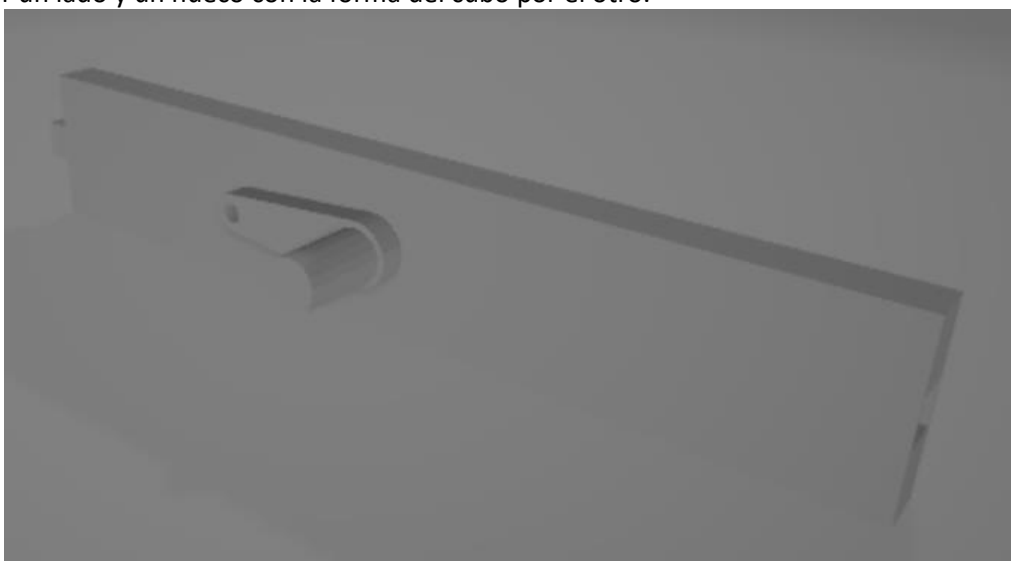


Ilustración 20 Eje externo

Eje interno

Para el eje interno se ha decidido por una solución similar a la del eje externo pero para que pueda ser contenido dentro del externo este es más corto ya que entre el marco externo y el marco interno hay que albergar el soporte para la cámara y este no tiene que rozar con nada, además se ha acortado ligeramente el eje y se ha separado la barra del marco para poder introducir una barra pasante, este diseño al colocarse no era lo suficientemente grande con la intención de arreglarlo sin gastar mucho material se ha diseñado además una especie de junta que aumentan el diámetro del eje 1 mm y le dotan de flexibilidad para un mejor agarre en el marco externo.

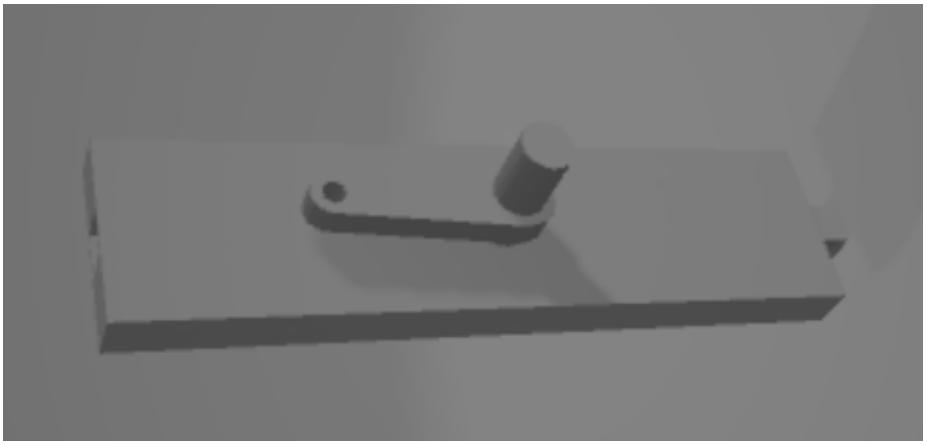


Ilustración 21 Eje interno

Soporte marco cámara

Para soportar el eje de la cámara se ha diseñado un soporte muy simple se basa en una barra igual que la anterior sustituyendo el eje con un hueco donde encajara en vertical el otro marco tanto en un lado como en el otro habrá huecos para tener que diseñar un menor número de piezas y mantener la simetría en la estructura, al igual que las anteriores tienen para unirse un cubo que sale y otro hueco donde encaja.

Ilustración 22 Soporte marco cámara

Marco cámara

Para elevar la altura de la cámara se ha optado por una barra la cual no tiene ningún hueco simplemente contiene dos cubos a 90 grados uno de ellos se encajará en el marco interno mientras que el otro se incrustara en una esquina como el sistema no es simétrico necesitaremos diseñar dos modelos de esta pieza uno con la esquina hacia fuera y otro con la esquina hacia dentro para que la barra horizontal pueda tener tanto parte saliente como hueca.

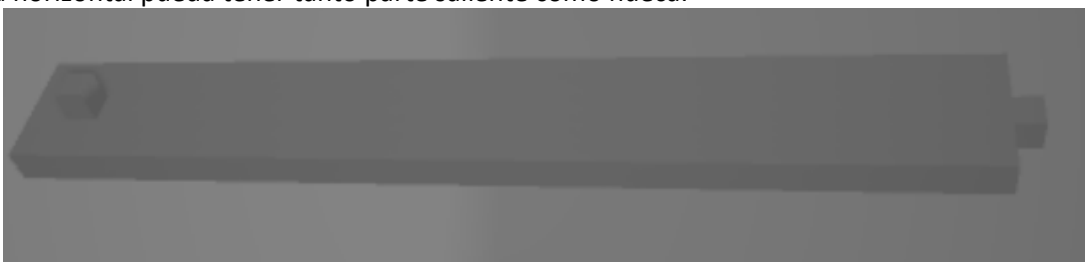


Ilustración 23 Marco cámara 1

Ilustración 24 Marco cámara 2

Montura Cámara

Para poner la cámara se ha diseñado una barra las dimensiones de esta barra tienen que ser dos veces el ancho del marco más largo ya que hay que cerrar completamente la estructura un error en esta pieza haría que el marco no encajara además nos fijaremos en la hoja de datos de la cámara para comprobar donde tiene la cámara los huecos para añadir el soporte y su tamaño si queremos situar la cámara exactamente en el centro, para mi sorpresa los soportes no están simétricos a la placa si no que están alineados al centro de la cámara y separados una distancia con lo cual hay que tener cuidado con la orientación de la pieza si no queremos tener la imagen descuadrada además al diseñar este soporte no me di cuenta que la cámara cuenta con un conector con lo que diseñe una especie de cama donde se encajaría por una parte en las aberturas anteriores y por otra en la cámara haciendo un espacio para que descanse el conector.

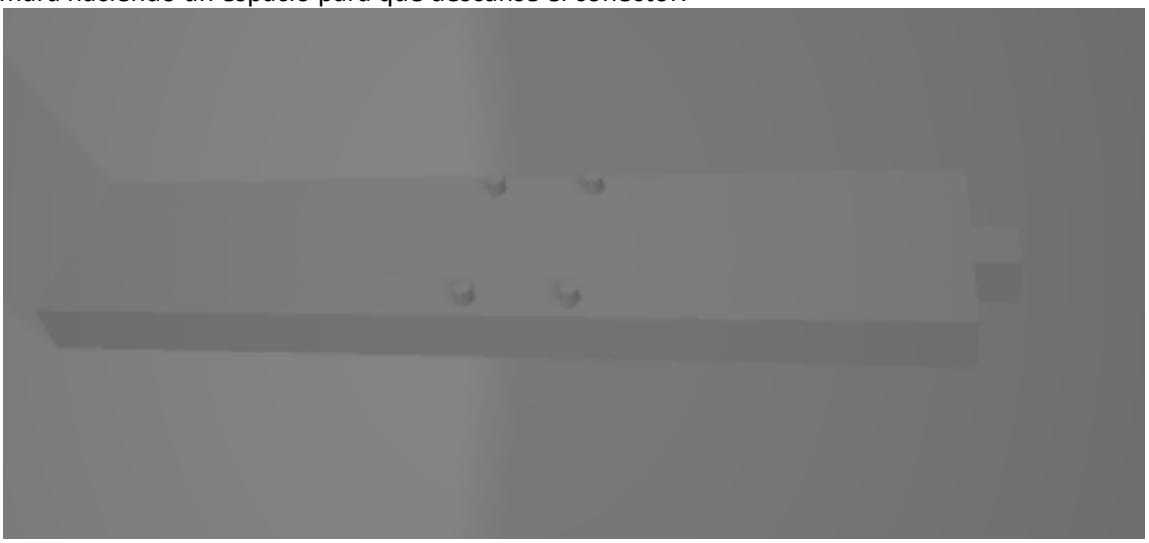


Ilustración 25 Montura cámara

Enganches motores

Para unir los motores a la estructura la solución más fácil era generar unos enganches que tienen doble o incluso triple función la primera es sujetar los motores mediante una brida lo que permite cambiar de motor si el sistema cambia además si el sistema tiene muy desplazado el centro de gravedad se le pueden añadir contrapesos que desplazan el centro de gravedad donde queramos la última función y posiblemente más importante es sujetar el tablero sobre el que se moverá la bola.
Para el motor externo se copiará la inclinación de la pirámide como si fuese su huella así solo hay que deslizar la placa hasta que esta no baje más con ese encaje será más que suficiente.

Ilustración 26 Enganche motores

Tablero y bola

La idea del tablero y la bola es que sean fáciles de conseguir por ello las ideas han ido al mundo del deporte, para la bola la idea es que sea hueca, algo grande y vistosa para la cámara durante la compra en la tienda de deportes se valoraron dos opciones una pelota verde de petanca y otra de pimpón naranja, la pelota verde era de mucho mejor tamaño y mucho más vistosa para la cámara pero al ser de petanca el peso era demasiado para el sistema con lo cual no quedo más opción que ir a por la pelota naranja que dependiendo de la iluminación podría parecer a la cámara blanca y esto arruinar el resultado pero con un buen filtro no debería de ser ningún problema. Para el tablero este tiene que ser ligero, rígido y fácil de moldear por otro beneficio como es la calibración de la cámara además de los previamente mencionados se ha elegido por un tablero de ajedrez de PVC de competición el cual sabemos exactamente lo que miden sus casillas y es perfecto para hacer contraste con la bola, aunque pueda parecer fácil comprar este tipo de tableros no lo es y son muy caros con lo cual en el prototipo se utilizó uno más simple y barato pero con las mismas propiedades.

Ilustración 27 pelota de pimpón.

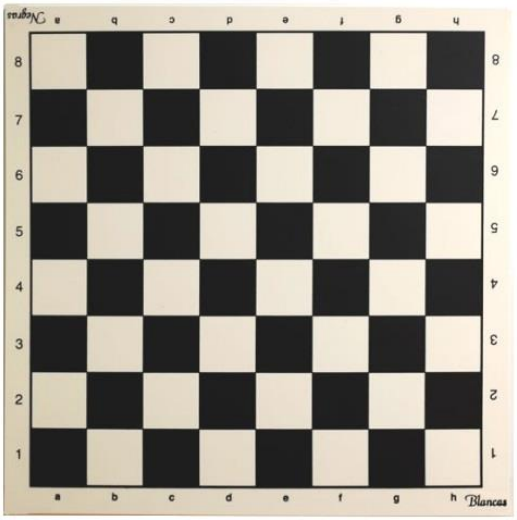


Ilustración 28 Tablero de ajedrez

4. DISEÑO ELECTRÓNICO

En este apartado se van a detallar y analizar los diferentes bloques que hacen posible el funcionamiento del sistema teniendo en cuenta las entradas y salidas del sistema, así como las variables que necesita cada bloque para funcionar.
Todo sistema de control es definido por un esquemático donde se contienen los grandes bloques del sistema incluyen los sensores, actuadores y controladores dentro de los mismos hay sistemas que pueden incluir filtros amplificadores y otros dispositivos electrónicos. Un diseño de bloques bien planificado y ejecutado es esencial para lograr un control optimo.

Introducción

A continuación, se plantearán todos los bloques necesarios, así como la realimentación del sistema para ello analizaremos en cadena directa el sistema sin realimentación simplemente conectando la entrada con la salida para ello necesitamos generar una entrada de forma discreta por un ordenador que se tratara como una única señal más adelante del proyecto se detallara en cómo se ha llevado a cabo la construcción de esta entrada, llega a un microcontrolador el cual aplica un PID para dar un valor de ángulo al motor que mueve la bola siendo este movimiento la salida el sistema final se detalla en el siguiente esquema:

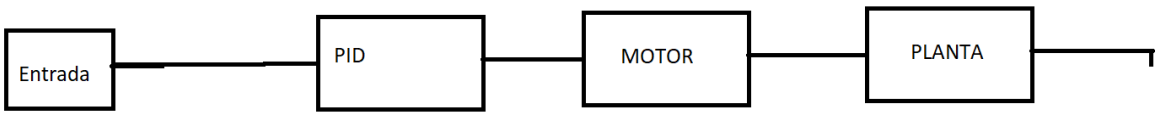


Ilustración 29 Cadena directa del sistema

En el esquema anterior la primera función de transferencia corresponde al motor y la segunda al movimiento de la bola.
Para la realimentación es necesario usar un sensor en este proyecto como sensor se utiliza una cámara digital que un controlador interpretara para localizar la bola e introducir la realimentación a la entrada.

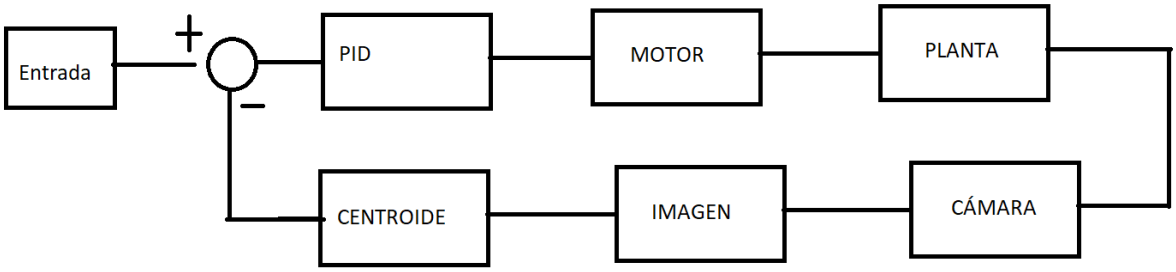


Ilustración 30 Sistema en lazo cerrado

El sistema controla tanto el eje X como el eje Y, pero el sistema de control es exactamente el mismo con los mismos elementos con lo cual en este apartado nos centraremos en un solo lazo además el control es discreto, aunque el movimiento se realizara de forma linear ya que el motor contiene el mantenedor.
La parte de entrada y realimentación se realizará con un controlador mientras que la parte de mandar la señal al motor se realizará con otro microcontrolador en medio hay que añadir un sistema de comunicación que explicare en la parte de controladores del proyecto.

Adquisición

Para la adquisición de datos solo se necesita saber la posición del centro de la bola para hacer esto hay muchas soluciones las tres más usadas es por detección óptica, panel táctil resistivo o con una cámara a continuación explicare las diferencias entre todas y como seria la implementación de cada una y los motivos que me han llevado a elegir la cámara.

Sensor laser

El sensor laser se basa en el principio fotoeléctrico, lo primero que se hace es generar un haz de luz laser mediante un fotodiodo en el punto donde este haz impacte tendremos un receptor el receptor está formado por un fototransistor el cual tendrá dos estados encendido o apagado para detectar la bola ponemos muchos fotodiodos con su fototransistor al otro lado del marco en tándem y mediante un microcontrolador sacaremos la posición estimada de la bola y podremos mandar esta posición al microcontrolador del motor. La principal ventaja de este sistema es su velocidad siendo un sistema rapidísimo y además al tener menos datos no necesitamos tanta potencia de cálculo como para meter un controlador simplemente es una transformación de señal a un número digital conteniendo la información del punto medio que en una señal de 8 bits es codificable.
El uso de este sistema pese a aumentar la velocidad perdería resolución y complicaría el diseño electrónico simplificando el sistema de control.

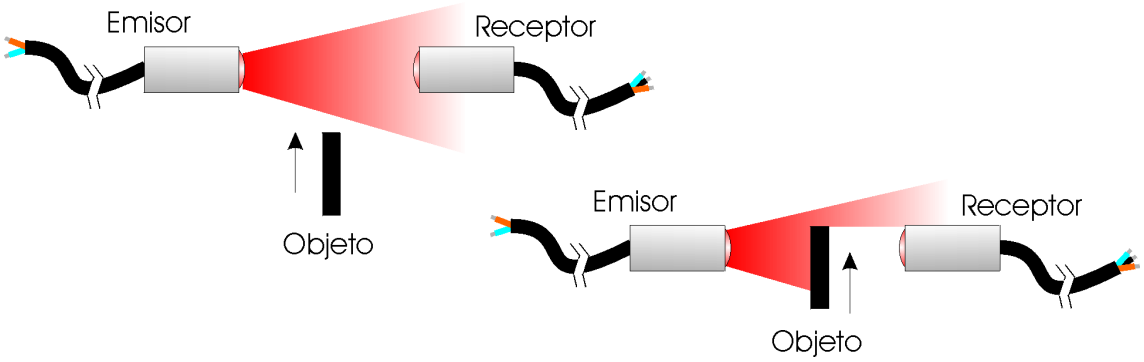


Ilustración 31 Sensor de pantalla laser

Pantalla táctil resistiva

Si queremos usar una pantalla táctil esta tendrá que ser resistiva ya que la capacidad de la bola no será suficiente para un sistema con detección capacitiva por ello nos tenemos que quedar con la pantalla resistiva, los sistemas que no utilizan cámara utilizan en su mayoría este sistema ya que por precio una pantalla resistiva es barata y su acondicionamiento es muy sencillo, se basa en un material que se estira en el punto exacto donde se encuentra la masa realizando presión es decir que la resolución sería muy grande ya que el punto que salga de la medida es directamente el centro de la bola además solo necesita 4 hilos dos para la resistencia en la dirección vertical y otros dos en la dirección horizontal alimentando el sistema tendremos una tensión proporcional a la alimentación que nos diga la posición de la bola con gran exactitud el principal inconveniente es su tamaño siendo limitado y no pudiendo hacer el panel del tamaño que necesito además para la medida necesito hacer uso de convertidores analógico digital que muchas veces inducen a error en la medida y necesitan ser compensados.

Gracias a un panel táctil también eliminaría el uso de un controlador y el error aun existiendo es controlable peor no me quito el problema del tamaño de la pantalla fabricarme mi propia pantalla resistiva con galgas es posible pero también tendría que añadir la parte electrónica de alimentación del puente Wheatstone.

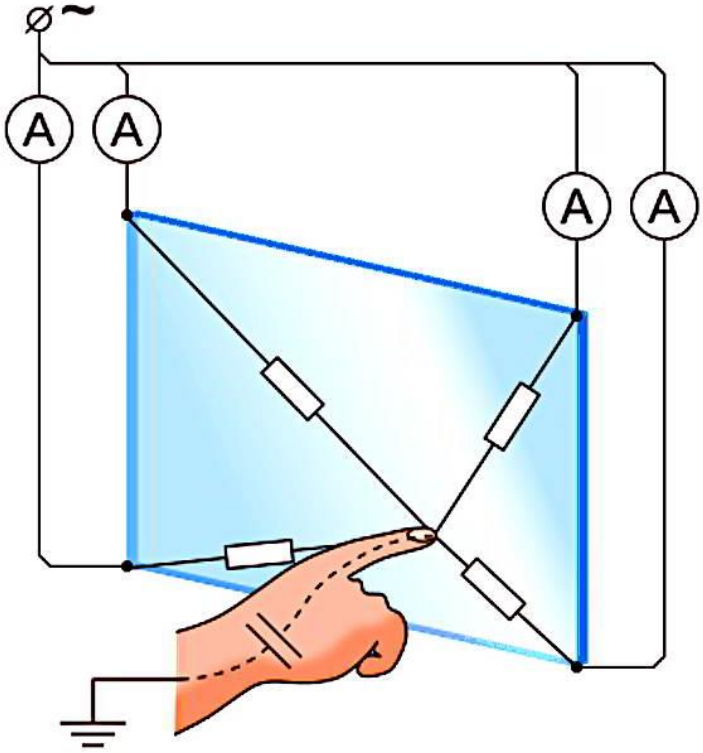


Ilustración 32 Pantalla resistiva

Cámara

Finalmente, la cámara es la solución propuesta es un sistema que se basa en grabar un video de la bola y que con un controlador potente conseguir analizar la imagen para saber dónde está el centro de la bola este sistema es el más lento de todos pero a nivel de instrumentación es simplemente conectar por el puerto serie o paralelo a la placa sin ninguna complicación adicional además es la solución más barata de todas y al forzarnos a usar el controlador podemos añadir más características como un control remoto en tiempo real.

Ilustración 33 Cámara Raspberry pi

Muestreo

Para muestrear la señal necesitamos un dispositivo digital que sea capaz de comunicarse con la cámara para interpretar los valores del sensor, para capturar datos puede ser analógico la mayoría de veces este dispositivo es analógico ya que solo necesitamos un hilo y si queremos usar varias cámaras nos ahorraremos una gran cantidad de señales, pero sen este caso solo necesitamos una cámara y la distancia que separa la cámara del dispositivo no supera el metro con lo cual usaremos un dispositivo digital que nos forzara a una configuración de tiempo de muestreo vía software, dentro de los dispositivos podemos usar de tres tipos un microcontrolador, un pc o un single board computer.

Microcontrolador

Existen diferentes tipos de microcontroladores que se denominan familias dependiendo de la compañía que lo fabriquen y los bits que sea capaz de manejar además son de muy bajo coste y ampliamente usados por la industria para todo tipo de aplicaciones de visión artificial habiendo un catálogo inmenso de cámaras preparadas para interactuar con el microcontrolador.
Las principales familias de microcontroladores que se podrían usar son:

- PIC: son de los primeros que aparecieron en el mercado se encarga de su fabricación y soporte Microchip y se pueden programar tanto en C como en ensamblador dando las instrucciones directamente al chip.

Ilustración 34 PIC32

- STM32: son de ST y además ofrecen soluciones de software personalizadas que permiten tanto la programación ensamblador como en C también dependiendo de los periféricos que contenga puede tener diferentes gamas.

Ilustración 35 STM32

- ESP32: están fabricadas y mantenidas por una empresa china llamada ESPRESIFF se puede programar tanto en C como en Python además es perfecta para aplicaciones inalámbricas ESPRESIFF vende un modelo con la cámara de visión artificial incorporada.

Ilustración 36 ESP32

- Arduino: se trata de una empresa que se dedica a desarrollar una plataforma para procesadores RISC con que se puede usar con todos los anteriores y con unas placas que de hardware libre de la propia compañía con soporte completo.

Ilustración 37 Placa Arduino UNO

Este tipo de solución es muy barata y con un consumo de potencia muy reducido pero los núcleos no tienen gran velocidad no pasando de los 300 MHz y la memoria está limitada con lo que no podemos incluir grandes programas debiendo usar o sin sistema operativo o con sistemas operativos muy ligeros dedicados a la gestión de los recursos de la placa, la idea de estos dispositivos es combinarlos entre sí para lograr aplicaciones complejas.

PC

Los PC son muy usados en la industria debido a su popularidad en la sociedad permiten una gran variedad de aplicaciones a nivel de hardware pueden estar basados en ARM como los microcontroladores anteriormente mencionados o en x86 arquitectura patentada por Intel y que es usada únicamente en algunos PC y sobremesa ya que por sus características de potencias para aplicaciones móviles son mejores los procesadores ARM.
El uso de un pc permite el uso fácil de periféricos por USB se pueden usar con un sistema operativo en el cual instalar diferentes programas y entornos de programación, el sistema operativo más popular es Windows y en este proyecto se usará para programar la entrada mediante el lenguaje de Python en VScode.

Estos dispositivos son muy potentes llegando a frecuencias de 5GHz, aunque su consumo también es elevado y su coste es muy caro, aunque los problemas de costes los podríamos solucionar y los de consumo de potencia también la principal desventaja del PC es que no puede interactuar directamente con el mundo físico si no que necesita de una placa traductora que haga el accionamiento del control como un microcontrolador.

Ordenador monoplaca

Los ordenadores monoplaca podemos decir que es un punto intermedio entre el microcontrolador y el ordenador esto es muy útil en la industria ya que para las aplicaciones no tendrías que renunciar al control con el mundo real y si queremos lo de un microcontrolador tenemos más potencia están basados en procesadores ARM y es capaz de instalar sistemas operativos y manejar periféricos USB, al igual que los microcontroladores hay diferentes familias dependiendo del fabricante y cambian propiedades como la velocidad del chip, el tamaño de las memorias RAM o el número de periféricos disponibles.

- BeagleBone se trata de una placa producida por Texas Instruments optimizada para aplicaciones de IA y aprendizaje automático de código abierto y bajo coste además hay una versión con rangos industriales de temperatura su tamaño compacto la hace ideal para este tipo de aplicaciones.

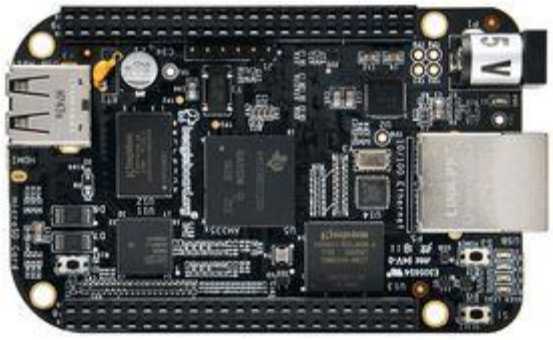


Ilustración 38 BeagleBone

- Asus Tinker Edge R es una placa fabricada y vendida por Asus donde destaca una gráfica dedicada más potente y núcleos pensados específicamente para aplicaciones de aprendizaje automático llamados NPU a nivel de hardware es mucho más compleja y requiere más potencia además tiene un costo muy elevado

Ilustración 39 Asus Tinker Edge R

- JETSON nano, haciendo competencia a la Asus NVIDIA fabrica esta placa pensada para aplicaciones de desarrollo y prototipos para IA y robótica en este caso se centra en los gráficos permitiendo una gran potencia y renderizado de videos y aplicaciones similares.

Ilustración 40 JETSON nano

- Raspberry pi están desarrollados en Reino Unido por la Raspberry pi fundación es el modelo más popular diseñado tanto para la industria como para la educación su precio es reducido además venden de forma nativa una cámara para aplicaciones de visión artificial la empresa tiene una comunidad muy grande que está en constante crecimiento y tiene muchos proyectos con una comunidad tan variada y activa cualquier duda que surja en la realización de la programación o la conexión del hardware será fácil de resolver.

Ilustración 41 Raspberry pi 4

- Banana pi debido a la pandemia mundial de 2020 y un crecimiento en el interés de la población combinado con el intereses que ya venía teniendo la industria en esta placa antes hizo que sus ventas crecieran exponencialmente lo que siendo una compañía no muy grande agoto completamente el stock de placas provocando un hueco en el mercado donde han surgido otras placas claramente similares que pretenden por un precio similar sustituir a la raspberry pi mediante un mejor sistema de producción, el principal clon es la Banana pi aunque hay muchos más con fabricación en china.

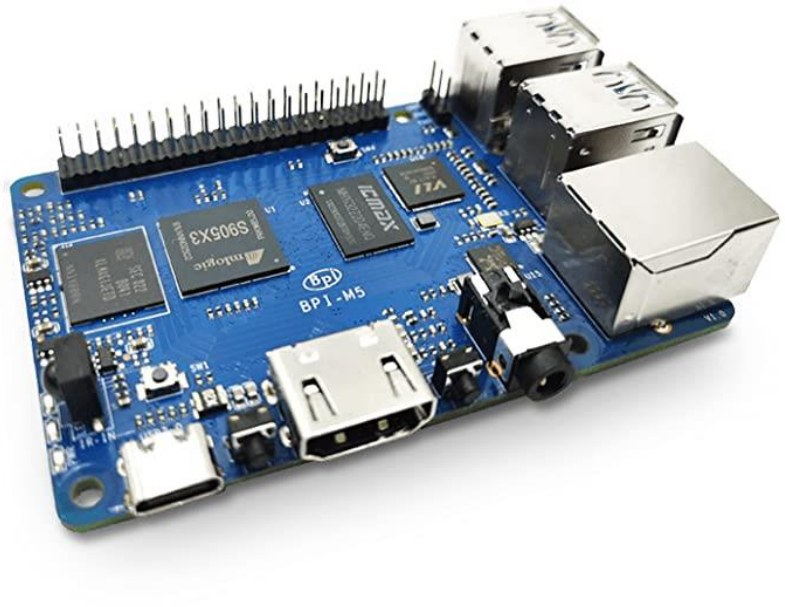


Ilustración 42 Banana pi

Aunque todas las soluciones anteriores puedes servir para realizar la aplicación en el proyecto esta última es la que más se ajusta a lo que necesito en el proyecto siendo algunas baratas y con una potencia más que suficiente optimizadas para aplicaciones similares de visión y localización de objetos mediante computador.

Esta placa se va a encargar de recoger los datos de la cámara por el puerto especifico paralelo de placa que incluye, calibrar la imagen para evitar distorsiones, localizar el centro de la bola en la imagen y calcular el error entre este y la referencia para mandar el dato por el protocolo que se desee al PID contenido en otra placa aunque esto último podríamos hacerlo en la misma placa ya que tenemos los puertos necesarios mandando el dato y no calculando la acción del regulador ganamos mucho tiempo de muestreo además gracias a las librerías de OpenCV para Python podemos obtener una imagen más pequeña suficiente para ver la bola pero a una mayor tasa de fotogramas.

La Entrada se generará mediante un array de posiciones que ira pasando la bola en línea recta este array se puede escribir de muchas formas manualmente dándole los datos con Matlab generando un programa que compile trayectorias, aunque en el apartado de códigos se va a explicar en detalle no pertenece al lazo de control y el proyecto no necesitara el generador de trayectorias para funcionar.

Regulación

Una vez que nos llega el error tenemos que encontrar la forma de modificar este dato para que el motor se mueva como deseamos y traducir de distancia a señal de motor para ello en este apartado voy a analizar las diferentes señales que se le puede dar a un actuador y los diferentes mecanismos de control que se pueden implementar.

Mecanismos de control

En el mundo de la automática existen muchos mecanismos de control y de cada uno se pueden hacer pequeñas modificaciones para mejorar sus características, en este apartado se verán las ventajas e inconvenientes de implementar uno u otro.

PID

El PID es el más simple de los mecanismos de control se trata de añadir al sistema un control proporcional K un control integral 1/s añadiendo este bloque eliminaremos el error en régimen permanente y una parte derivativa s que controla la parte transitoria del sistema esos bloques son combinados en un sumatorio que hará un bloque único de control.

- P la función de transferencia del control proporcional es Kp nos permite controlar la amplitud de la señal y amplificarla.
- PI Este control se basa en multiplicar la integral del error por una constante Ki que controle el tamaño del polo en S su función de transferencia es

  $$\frac{Ki}{s}$$

  pero si usamos la inversa nos sale ti que es más útil ya que permite controlar la frecuencia del control integral en el dominio temporal.
- PD para compensar la acción temporal del PI se añade un control derivativo mediante el producto de una constante Kd con la derivada del error en el entorno de Laplace la función de transferencia es Kd*s gracias a esta Kd podemos controlar la velocidad que tiene el sistema y su comportamiento transitorio.
- PID para hacer un PID simplemente se suman los 3 puntos anteriores en el tiempo es

  $$Kp * e(t) + Ki * \int e(t)dt + Kd * \frac{\mathrm{d}e(t)}{\mathrm{d}t}$$

  sí pasamos a Laplace nos queda

  $$Kp + \frac{Ki}{s} + Kd * s$$

  para hacer la función de transferencia multiplicamos por s a ambos lados de la ecuación y queda

  $$F(s) = \frac{Ki * Kp * s + Kd * s^2}{s}$$

  en sistemas continuos la función de Laplace es sencilla, pero si tenemos un sistema discreto como el que vamos a tener siempre esta ecuación es un poco diferente ya que en vez de integral se retrasa una muestra y en vez de derivar se adelanta una muestra, aunque normalmente se realiza un cambio de variable de s a z con la última ecuación ya que los cálculos son más sencillos.

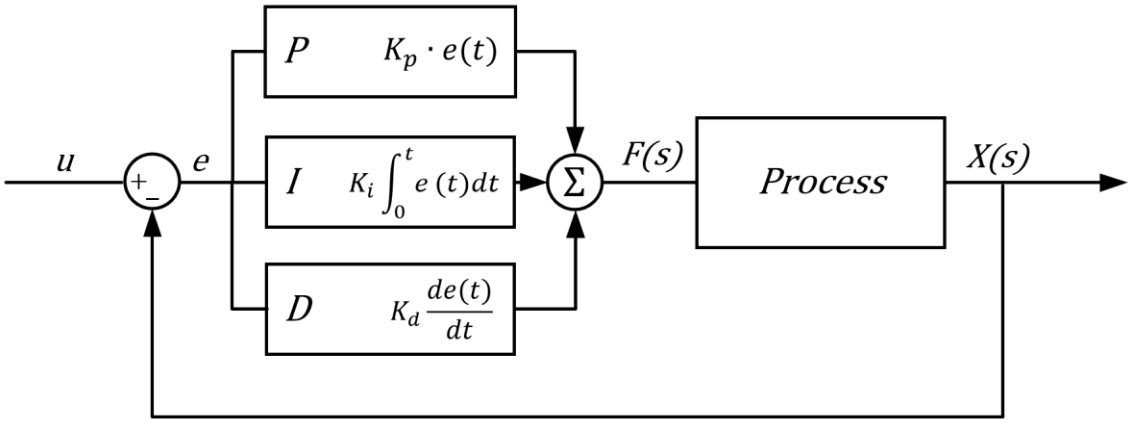


Ilustración 43 PID paralelo genérico

Realimentación de estados

Para hacer el control por realimentación de estados necesitamos conocer perfectamente el sistema no nos sirve únicamente con conocer su comportamiento para esto existen varias técnicas que van desde el diseño analítico hasta la experimentación para conseguir una representación compleja del sistema y sus variables de estado.

La ventaja es que nos permite un control completo del sistema realimentando el vector de estados en vez de la salida podemos aplicar un control mayor, este sistema se puede usar en combinación con una realimentación normal para aumentar el control del sistema.

La matriz B o matriz de ganancia conecta la entrada con el espacio de estados en la matriz A o matriz de estados se define como las variables de estado se relacionan con sus derivadas finalmente la matriz C o matriz de salida se relaciona el espacio de estados con la salida normalmente se añade una nueva matriz D que relaciona directamente la entrada con la salida, aunque en la mayoría de las ocasiones este vale 0 y no se añade al esquemático.

Al usar la matriz k nos permite situar los polos del sistema exactamente donde queramos en la parte de modelado de este proyecto se detallará un control por realimentación de estados teniendo en cuenta el equilibrio de la bola y el mecanismo de movimiento y se comparará con el control por PID.

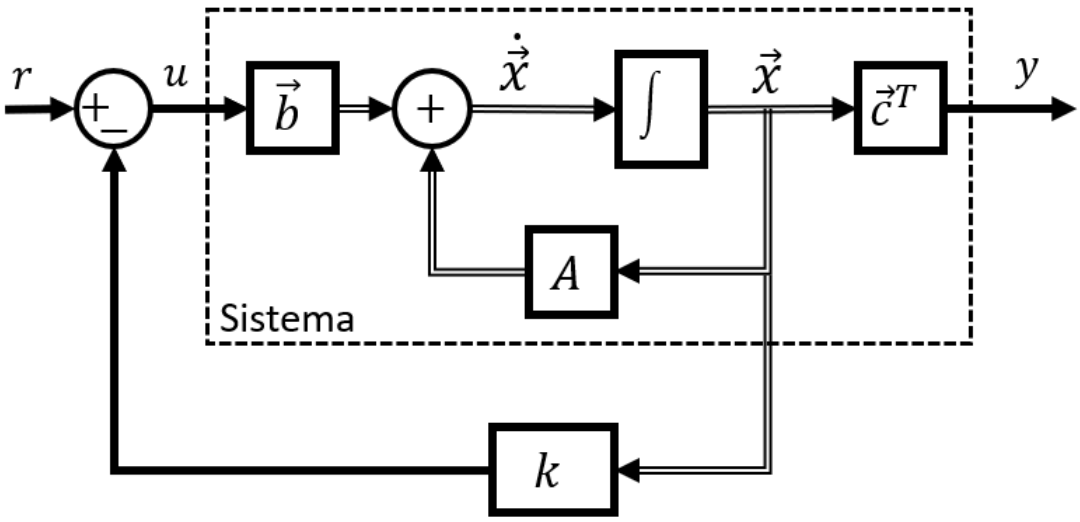


Ilustración 44 Sistema en variables de fase

Control adaptativo

El control adaptativo se basa en tener parámetros K o de PID que al igual que la planta cambien con el tiempo así evitar una degradación del sistema con el tiempo.
Hacer un control adaptativo implica realizar antes un modelo de referencia que debe ser invariante con el tiempo y conocer muy bien las salidas del controlador este controlador puede ser mismamente un PID al que una vez sabiendo la desviación que produce frente a la medida estimada vayamos ajustando sus parámetros Kp, Ki y Kd teniendo en cuenta como los cambios alteran al sistema.

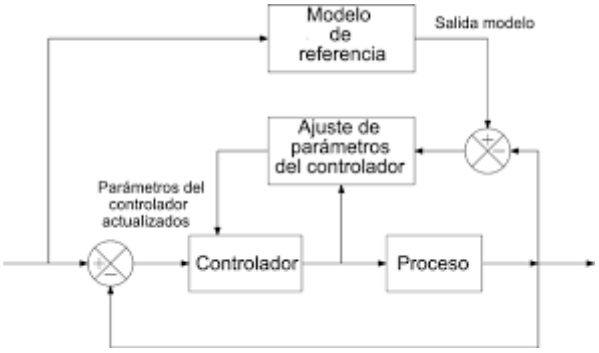


Ilustración 45 Esquema control adaptativo

Control predictivo

El control predictivo se utiliza cuando no podemos medir las variables del proceso entonces tenemos que generar un modelo que sea similar al sistema real y del cual si podamos medir sus parámetros este sistema recibe únicamente la salida de la planta real y es capaz de cambiar las variables de sí mismo para adaptar el modelo predictivo y el controlador que le entra al sistema real.
Este sistema funciona muy bien con plantas que no varíen demasiado con el tiempo si tenemos un sistema que sus cambios son grandes o cambia algún bloque del sistema el modelo no será capaz de ajustarse al sistema con lo que el sistema no funcionara correctamente.

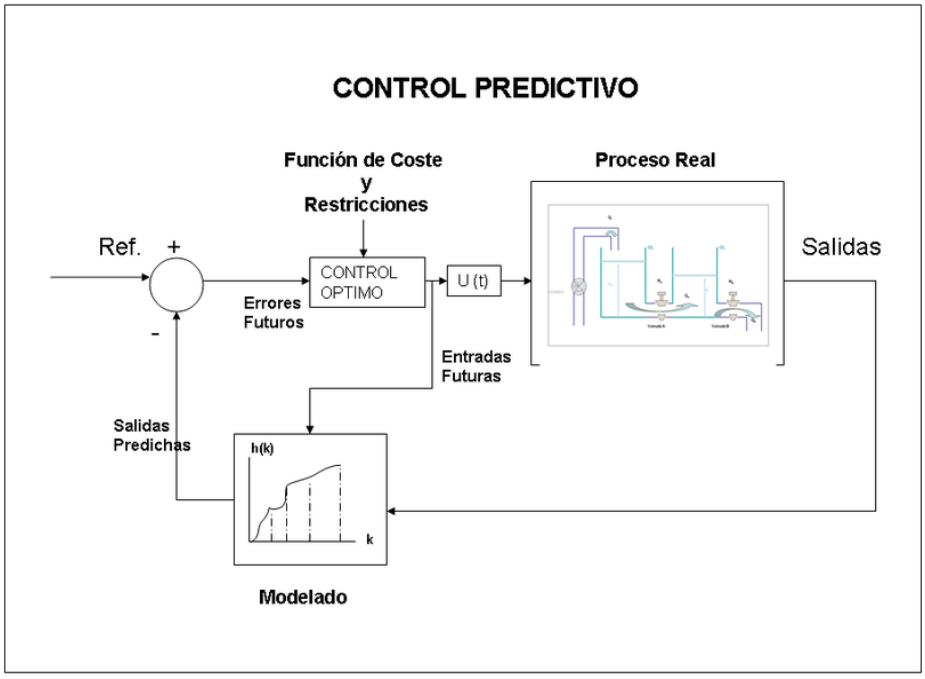


Ilustración 46 Esquema control predictivo

Señales de salida

Las señales que admite un motor en general pueden ser de 3 tipos dentro de cada tipo hay diferentes protocolos que aseguran un nivel de seguridad ante fallos de desconexión además hay dispositivos comerciales diseñados como configuraciones globales para estos sistemas que dependerán del tipo de actuador que utilizamos.

PWM

La señal más utilizada en estos ámbitos de aplicación por su sencillez y fácil codificación es la señal de ancho de pulsos se basa en generar una señal modulada a una cierta frecuencia que cambiando cuanto tiempo se encuentra en 1 o cuanto tiempo se encuentra en 0 podemos generar una referencia controlada como si fuese un nivel además a la hora de recoger la señal podemos filtrar las señales de esa frecuencia para reducir el ruido es usado en el control de motores ya que podemos controlar la velocidad o posición del mismo cambio simplemente el ciclo de trabajo con una resolución muy alta.

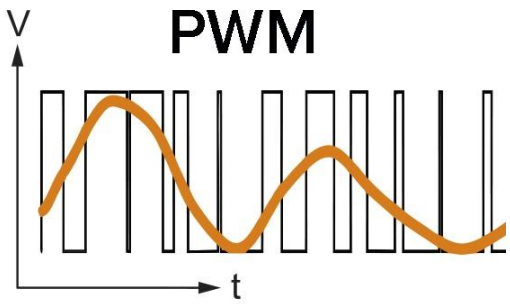


Ilustración 47 Señal PWM

Analógica

Una señal analógica tiene la ventaja de que en teoría tiene infinitos valores configurables además es capaz de en una línea llevar una gran cantidad de información se suele modular como una onda sinuosidad de una determinada frecuencia y amplitud que al alimentar las bobinas de un motor pueden hacer que este gire hay motores que para aumentar su eficiencia utilizan estas señales como entradas y necesitaremos un driver de variador de velocidad que genere las entradas con el desfase y amplitud adecuados para que el motor gire correctamente. El otro motivo por el que se usa una señal analógica es para recorrer grandes distancias donde una señal digital o PWM perdería gran parte de su energía en el trayecto. Como no necesitamos ningún variador de velocidad y la distancia que recorremos es muy corta no se utilizaran señales analógicas.

Ilustración 48 Señal analógica

Digital

Cuando queremos comunicar varios dispositivos digitales entre sí de manera alámbrica se utilizan señales digitales que se pueden generar de la misma forma que las PWM, pero se pueden combinar varias PWM para mandar datos muy rápido por protocolos recogidos por el IEEE universales además

podemos añadir varias líneas para ser completamente inmunes al ruido y desarrollar protocolos serie complejos como es el USB. Dependiendo del ámbito de aplicación se usarán unos u otros protocolos para la industria es muy usado el PROFIBUS y para control de microcontroladores el I2C o el I2S dependiendo si manejamos audio o video.
También podemos tener señales digitales más sencillas que actúen como un interruptor que simplemente hagan girar un motor en lazo abierto a un nivel constante de tensión hasta que se le deje de suministrar tensión y se vaya frenando con las perdidas internas este control no es el mejor, pero es el más sencillos y para aplicaciones muy simple puede ser útil.

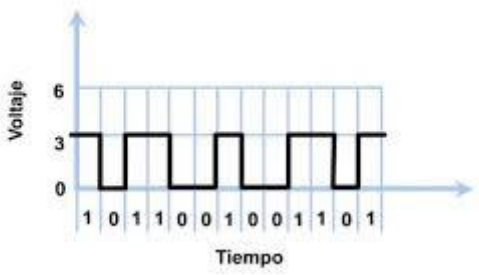


Ilustración 49 Señal Digital

Actuadores

En este apartado se describirán los diferentes actuadores que podrían hacer girar la estructura al tener que provocar un giro todos serán motores y hay de diferentes tipos:

Motor de corriente continua

Un motor de corriente continua es el más básico de los sistemas que se pueden usar los hay de dos tipos dependiendo de la velocidad y potencia que necesitemos puede ser con o sin escobillas si tenemos escobillas bastaría con darle la entrada lo que haría girar el motor pero los que son sin escobillas necesitan tres señales sincronizadas para funcionar, a la hora de elegir un motor es importante saber la tensión de alimentación, la cantidad de corriente que consume y cuanto par máximo es capaz de ofrecer para saber si tendrá fuerza suficiente para mover la estructura.
Si queremos usar este tipo de actuadores hay que añadir unos engranajes para reducir su velocidad y conseguir un alto par con una alta precisión a la hora de girar para el control necesitaremos un codificador conectado al eje que no es más que un acoplador que generara una señal de pulsos de diferente frecuencia dependiendo de la velocidad del eje.

Ilustración 50 Motor de corriente continua

Motor paso a paso

Otro tipo de motores que se puede usar son los motores paso a paso estos motores incorporan varios electroimanes que hagan girar al eje a base de apagar y encender las bobinas normalmente incorporan cuatro bobinas de las cuales están conectadas en serie sacando únicamente cuatro hilos dependiendo del orden en el que estemos a las bobinas podremos girar hacia un lado o hacia el otro estos motores pueden dar cuantos grados quieran pero mediante un incremento que tiene que venir previamente diseñado mediante una rueda dentada este incremento marcara la resolución de giro del sistema esta maneara única de construcción harán que tengan una velocidad reducida peor un gran par normalmente necesitaran un microcontrolador para transformar de ángulo a pasos.

Ilustración 51 Motor Paso a Paso

Servomotor

Un servomotor contiene en su interior un motor de corriente continua con unos engranajes acoplados, una pequeña parte electrónica que genera la entrada al motor DC, un sensor para la realimentación y un PID para el motor.
Es de muy bajo coste y de todo tipo de características en cuanto a señal de entrada, velocidad y par de salida son sistemas muy precisos y fáciles de implementar además pueden funcionar con diferentes niveles de tensión lo que le hace un sistema muy flexible, también se caracterizan por la cantidad de ángulos que giran los hay desde 45º hasta más de 360º dependiendo de las necesidades del sistema se elegirán unos u otros.
Para mover el servomotor se le añade una barra con un agujero atornillado en el eje lo que hará que se mueva de forma solidaria con el eje y cambiando la orientación de esta pieza podemos definir el 0 del sistema además esta barra viene con diferentes longitudes lo que permite una mayor configuración.

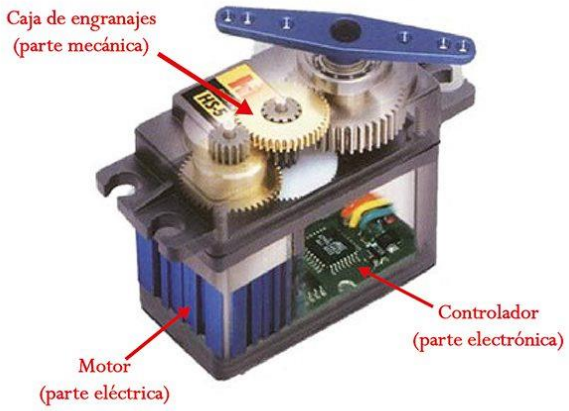


Ilustración 52 Servomotor

En este proyecto usare un servomotor como actuador tanto para el eje X como el eje Y por su fácil configuración con el microcontrolador y por su precisión para llevar el sistema al punto deseado además es un sistema muy robusto ante perturbaciones y que muy difícilmente se va a romper además las características de inclinación del plato son perfectas para usar este sistema y no otro el cual no sería de una forma tan directa.

Movimiento de la estructura

Para mover la estructura podemos optar por varios sistemas, aunque con la estructura que ya tenemos diseñada las opciones se reducen en gran medida dejando solamente a elección donde colocar los motores y como conectarlos a la estructura.
Dependiendo de donde se coloquen los motores estos tendrán que ejercer una mayor fuerza o incluso añadirán su masa al sistema que tendrá que ser movida con lo cual nos interesa tener los motores cerca de la estructura y del centro para generar el menor par posible las posibles ubicaciones son en el plano de referencia, en paralelo con el eje justo debajo, en el eje o en el marco contrario.
La solución de no fijarlo a la estructura generaría mucha distancia lo que le costaría demasiado al motor mover el eje.
Fijarlo al propio eje es a nivel de construcción posible pero no hay espacio suficiente en el marco externo para introducir el motor con lo cual solo sería posible en el marco interno.
Colocarlo en el marco contrario es posible ya que los ángulos van a ser limitados, pero va a suponer un esfuerzo mayor por parte de los motores y además alejaremos el peso del centro de la estructura para el marco externo.
Finalmente se ha decidido una solución intermedia se colocara el movimiento del eje interno en el marco interno alineado con el eje para ello del propio eje sale una barra de la misma longitud que la del servo donde se conecta para girar y el eje externo se reducirá considerablemente la distancia colocándolo en el soporte además esta última configuración no añadirá peso a la estructura y de la misma manera el eje externo sacara una barra de la misma longitud del servo para producir el movimiento del eje que al ser la misma pieza que el marco girara a la misma velocidad angular.
Para conectar los motores a la estructura necesitamos sacar la forma más sencilla y que requiera menos piezas para llevarlo a cabo el sistema designado es una barra que por un lado tendrá un hueco para conectarse al servo con un tornillo y por el otro lado tendrá un cilindro que quedara incrustado en la barra que sale del servo con este sistema se crea un sistema sin tornillos salvo el de los motores ya que esta parte es comercial e incluida con el motor lo que quiere decir que si se rompiese cualquier parte o si se quisiese comercializar el sistema no sería necesario la compra de tornillería ya que estos tornillos vienen incluidos en todos los servomotores comerciales, otra solución posible que se usa cuando quieres conectarte a huecos pasantes es el uso de unas varillas que van por un lado al servomotor mediante una L y por el otro al eje fijado mediante un sistema de tornillo y doble tuerca.

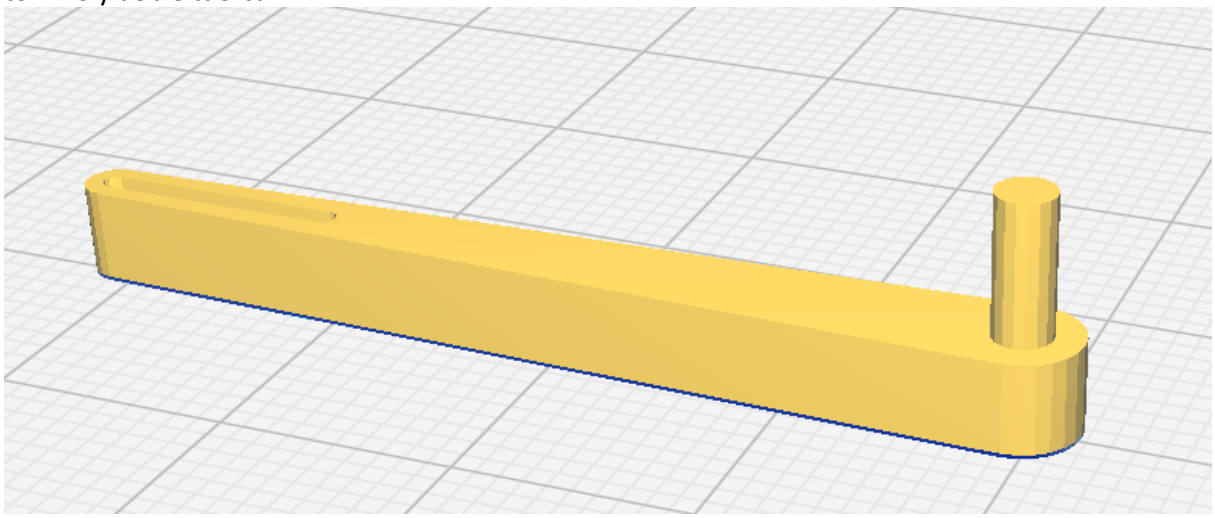


Ilustración 53 Barra servomotor

5. MODELADO DEL SISTEMA

En este apartado se llevarán a cabo todos los cálculos necesarios para llevar el proyecto a cabo mediante Matlab incluyendo los valores del PID que mejor vienen a cada motor, las medidas de la estructura, la selección de componentes y el tiempo de muestreo para que el sistema sea controlable.

Condicionantes de diseño

Dimensiones del tablero

Para calcular las dimensiones del tablero necesitamos saber la física de la bola en el plano inclinado teniendo en cuenta su velocidad máxima que depende del ángulo máximo que vayamos a aplicar estos valores son independientes del radio de la bola pues únicamente dependen del ángulo de inclinación de la estructura para ello aplicaremos las fórmulas de un movimiento rectilíneo uniformemente acelerado.

La primera formula es la de la velocidad respecto del tiempo partiendo de una velocidad inicial nula

$$V = a * t$$

Donde la aceleración de la bola en el plato viene definida por

$$a = \frac{g * sin(\alpha)}{1 + k}$$

K es el factor geométrico del cuerpo que en el caso de la esfera tiene un valor de 2/5 con esta ecuación podemos mostrar una gráfica que de la velocidad final en función de la longitud de los lados del tablero manteniendo un ángulo máximo menor que el ángulo critico es el que consigue superar la normal por el coeficiente de rozamiento estático y empieza a deslizar, la fuerza de rozamiento se calcula como

$$Fr = k * \frac{m * g * sin(\alpha)}{1 + k}$$

El ángulo critico es calculado resolviendo la inecuación Fr≤ μs·N que desencadena en la formula

$$tan(\alpha critica) = \mu s * \frac{(1 + k)}{k}$$

para calcular el valor de μ necesitaremos realizar varios experimentos y determinar cuánto tarda la bola en frenar en este caso lo que vamos a hacer es ponernos en el peor caso posible normalmente el coeficiente de rozamiento suele ser de entre 0.2 y 0.4 cuanto mayor es el rozamiento mayor es el ángulo crítico con lo cual usaremos un rozamiento de 0.15 para conseguir el mejor resultado posible lo que nos da un ángulo critico de 27.6995º, ahora podemos calcular cuánto es la velocidad final manteniendo un ángulo máximo de 25º, si consideramos una velocidad critica de 0.8 m/s podemos calcular la longitud del tablero como

$$(1 + k) * \frac{vcritica^2}{2 * g * sin(\alpha critica)}$$

Lo que da un valor de distancia de 0.2163 m que en centímetros son 21.63 cm.

Una vez tenemos la máxima longitud del borde de la bola que queremos necesitamos calcular la altura a la que tendremos que poner la cámara esta altura va a depender del tamaño del sensor y del tamaño de la plataforma esto se puede calcular sabiendo el ángulo de la cámara.

Con trigonometría sacamos que la altura es el seno de la mitad del ángulo de apertura este seno por la va a ser igual a la dimensión del tablero por la raíz de 2 así podemos calcular la altura mediante el teorema de Pitágoras

$$H = \sqrt{-(L^2/2) + \frac{(L^2/2)}{sen(\alpha/2)^2}}$$

con estos datos podemos cambiar el valor de L manteniendo un valor de Alpha de 54º.

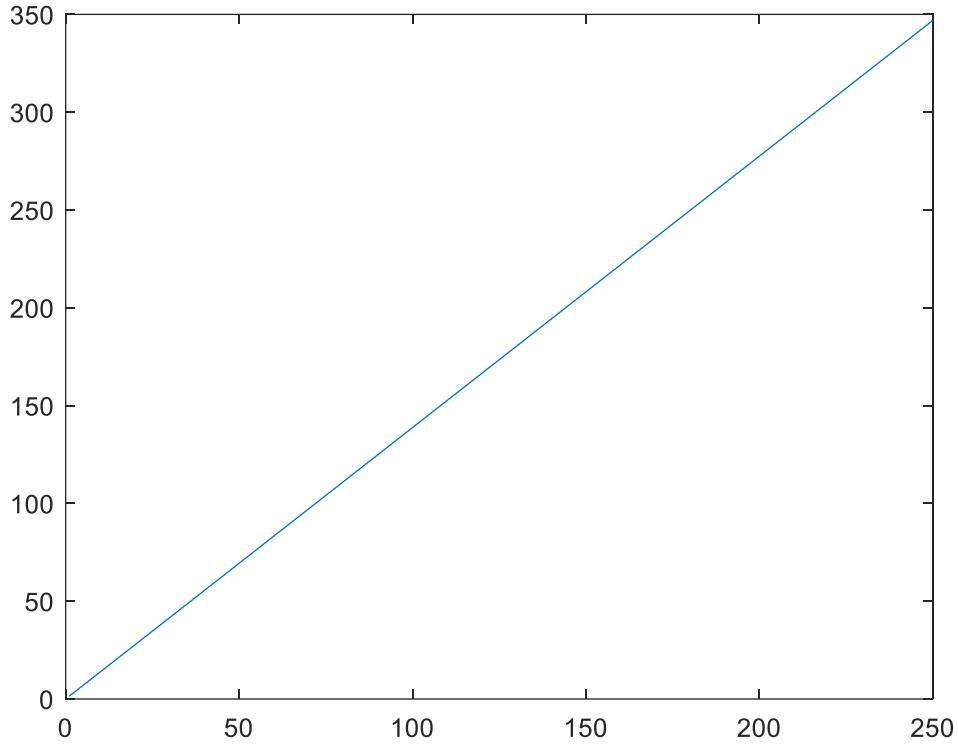


Ilustración 54 Grafica altura-lado

El valor de la altura como vemos en la gráfica es una línea recta entonces un buen valor que no pasa de los 270 mm es 190 mm de tamaño del marco lo que nos da una altura del sensor de 263.6773mm. este tamaño de altura se podrá conseguir situando el tablero en la base y haciendo un diseño modular de la estructura.

Cálculo de los grados de libertad

Una parte muy importante que se relaciona directamente con el título del proyecto es el cálculo de los grados de libertad del sistema.

Los grados de libertad son el número mínimo de sistemas para definir la posición del cuerpo respecto a un plano de referencia este concepto es el mismo que para calcular el número de actuadores necesarios y los sistemas por los que se va a desplazar.

En el espacio tridimensional en el cual se va encontrar la estructura tenemos un máximo de 6 grados de libertad dividiéndose estos en translaciones y rotaciones nuestro sistema va a estar fijo a la mesa con lo cual no va a tener ningún grado de libertad que le permita moverse en cuanto a la rotación se suelen denominar como en los aviones alabeo, cabeceo y guiñada, dependiendo de sobre que eje nos pongamos a rotar, la estructura no rota sobre el eje vertical con lo cual únicamente nos quedan dos que permiten la rotación del plano X-Y a voluntad.

Esta forma de calcular los grados de libertad es en la mayoría de las ocasiones acertada y muy útil para comprobar el resultado del análisis de los grados de libertad pero no es la correcta ya que para sistemas más complejos no nos será tan fácil imaginarnos el mecanismo, la forma correcta es usando el criterio de Chebychev donde dividimos toda la estructura en barras y pares, las barras no

permiten movimiento mientras que los pares si y dependiendo de su grado pueden permitir diferentes grados de libertad.
La fórmula considera primero el máximo número de grados de libertad de un sistema también M=6*(N-1) donde N es el número de cuerpos en movimiento más el cuerpo fijo y luego va aplicando las diferentes restricciones de los pares que lo unen.

$$GDL = 6 * (N - 1) - \sum_{i=1}^{j} (6 - j)$$

Donde j es el número de grados de libertad que permite la unión en este sistema tenemos un total de 10 barras correspondientes a N que quitando una siendo esta fija nos queda 6*9=54 luego pares tenemos dos tipos las esquinas que son de clase 2 es decir permiten tanto la rotación en un eje como en el otro y los rodamientos los cuales solo permiten la rotación en una dirección.

$$GDL = 6 * (10 - 1) - 4 * 5 - 8 * 4 = 2$$

Un diseño esquemático donde se muestra la estructura sobre la que se ha realizado el estudio de los grados de libertad es la formada por dos marcos con cuatro esquinas cada uno y cuatro barras y externamente dos pares de primer grado cada uno que son los que permiten mantener los dos grados de libertad deseados por la estructura.

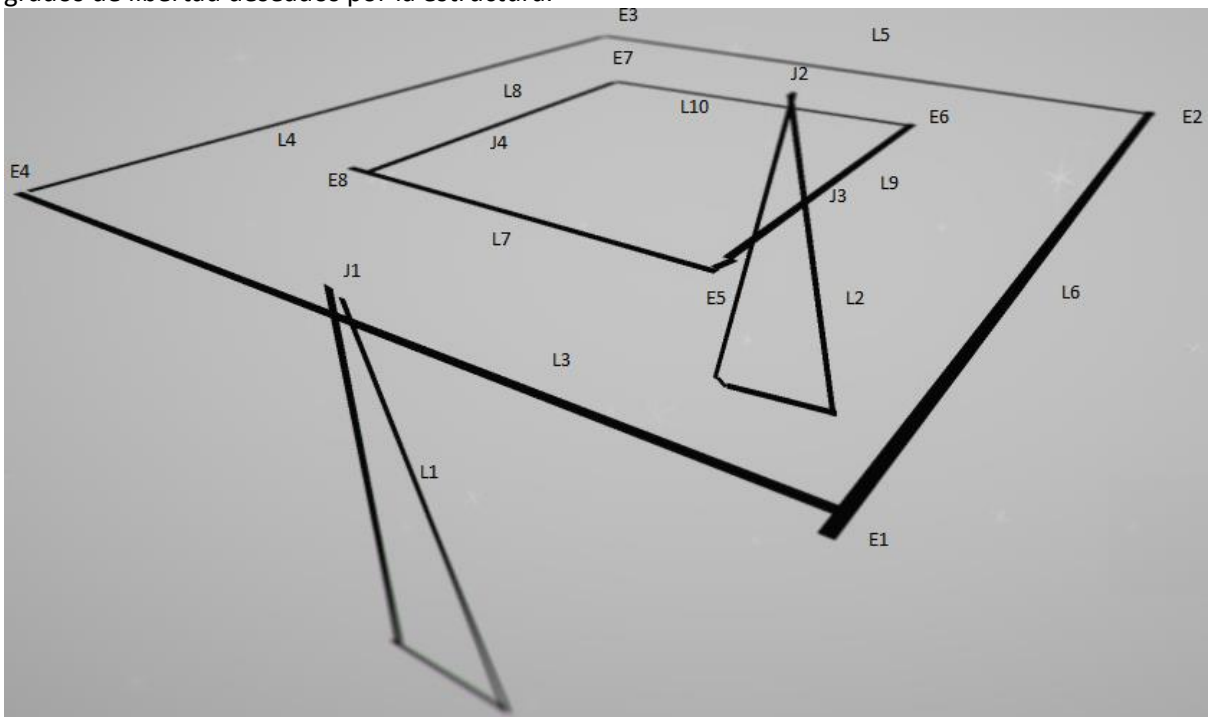


Ilustración 55 Calculo de grados de libertad

Mecanismo de barras

En esta parte del proyecto necesitare saber cuánto cambia el ángulo de la estructura con el cambio del ángulo del servomotor este cálculo es importante ya que nos ayudada a decidir cuál es el ángulo máximo que debe girar el servomotor y cuanta resolución del servomotor es pasada a la estructura.
A este proceso se le considera cinemática inversa y se puede resolver de dos formas diferentes mediante matrices de transformación o mediante un método geométrico en este proyecto se llevará a cabo el método geométrico el cual se basa en las coordenadas de los puntos de unión de las articulaciones, el sistema seguirá un esquema como el de la siguiente figura:

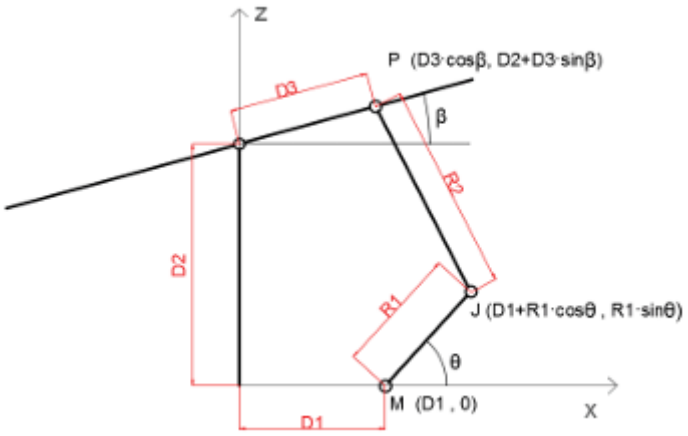


Ilustración 56 Mecanismo de barra

En él se considera el origen como el centro de la estructura en x y la altura del motor en z la idea de este análisis es dar un ángulo conocido normalmente el de la estructura ya que el servo puede rotar 180º y la estructura esta más limitada y luego mediante la inserción de circunferencias en los puntos M y P podemos sacar las ecuaciones para resolver la relación angular.

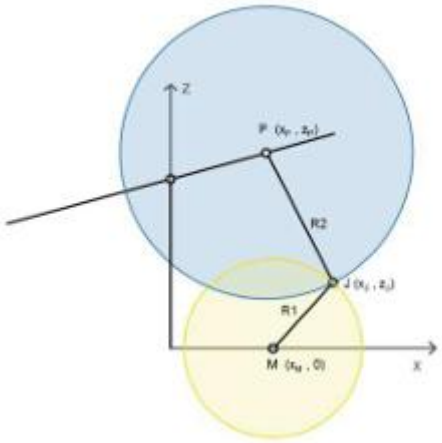


Ilustración 57 Círculos del esquema

La circunferencia azul tiene de radio R2 y origen P(D3*cos(β), D2+ D3*cos(β)) y la amarilla radio R1 y origen M(D1,0) aplicando la ecuación de la circunferencia obtenemos para la circunferencia amarilla.

$$(x - D1)^2 + (z - 0)^2 = R2^2$$

Simplificando y ordenando la ecuación queda:

$$z^2 + x^2 - 2 * x * D1 + D1^2 = R2^2$$

Para la circunferencia azul aplicamos las mismas formulas cambiando los puntos.

$$(x - D3 * cos(\beta))^2 + (z - D2 + D3 * cos(\beta))^2 = R1^2$$

simplificando queda

$$z^2 + x^2 - 2 * x * D3 * cos(\beta) - 2 * z * (D2 + D3 * cos(\beta)) + D1^2 + (D2 + D3 * cos(\beta))^2 = R1^2$$

Ahora tenemos dos ecuaciones con dos incógnitas con lo cual podemos resolver el sistema, para resolver la ecuación restamos las ecuaciones anteriores eliminando los valores al cuadrado de z y x ahora tenemos que sacar factor común a las variables x y z con la x queda 2*(D3*cos(β)-D1) y para z 2* D2+D3*cos(β) ahora simplemente agrupamos todos los términos que no tienen ni x ni z así nos queda un término independiente

$$A = R2^2 - R1^2 - D1^2 + (D3 * cos(\beta))^2 + (D2 + D3 * sin(\beta))^2$$

La ecuación final que relaciona x con z se puede poner como

$$(D2 + D3 * sin(\beta))z = (D1^2 - D3 * cos(\beta)) * x + 2A$$

que si simplificamos aún más queda

$$z = B * x + A'$$

Donde A y B vienen definidas como:

$$A' = \frac{(R2^2 - R1^2 - D1^2 + (D3 * cos(\beta)^2 + (D2 + D3 * sin(\beta)^2)}{2 * D2 + D3 * sin(\beta)}$$

$$B = \frac{D1^2 - D3 * cos(\beta)^2}{D2 + D3 * sin(\beta)}$$

Ahora tenemos que sustituir z en la primera ecuación que es la que menos términos tiene nos queda una ecuación de segundo grado.

$$x^2 * (1 + B^2) + x * (2AB - 2D1) + (A^2 + D1^2 - R2^2) = 0$$

Con esta ecuación sabemos todas las variables y podemos sacar cuánto vale la salida para los distintos motores considerando un ángulo de la plataforma de 0º.

Las dimensiones para el eje X y el eje Y vienen reflejadas en la siguiente tabla, así como su resultado:

Eje	D1	D2	D3	R1	R2
X	0	30	24	24	40
Y	10	30	24	24	40

Tabla 1:Medidas aproximadas mecanismo de barras

Para simplificar los cálculos haremos un código en Matlab que lo resuelva para ello sacaremos el valor de x con la fórmula de ecuaciones cuadráticas definiendo un polinomio con la ecuación de segundo grado y sacando sus raíces luego calcula los valores de z y pasa estas posiciones a ángulo con el arco tangente luego mostraremos los resultados en una gráfica, este código da dos soluciones de la cual solo una será posible en la realidad.

El código en Matlab para resolver el ángulo es el siguiente:

```
D1=0
D2=30
D3=24
R1=24
R2=40
t=1:1:90;
phi=zeros(2,90)
for Be=1:1:90
B=(D1-D3*cosd(Be))/(D2+D3*sind(Be));
A=(R2^2-R1^2-D1^2+(D3*cosd(Be))^2+(D2+D3*sind(Be))^2)/(2*(D2+D3*sind(Be)));
p=[(1+B^2) (2*A*B-2*D1) (A^2+D1^2-R2^2)]
x=roots(p)
z=B*x+A
angle=atand(z/x)
phi(:,Be)=angle(:,1)
end
figure
plot(phi,t)
```

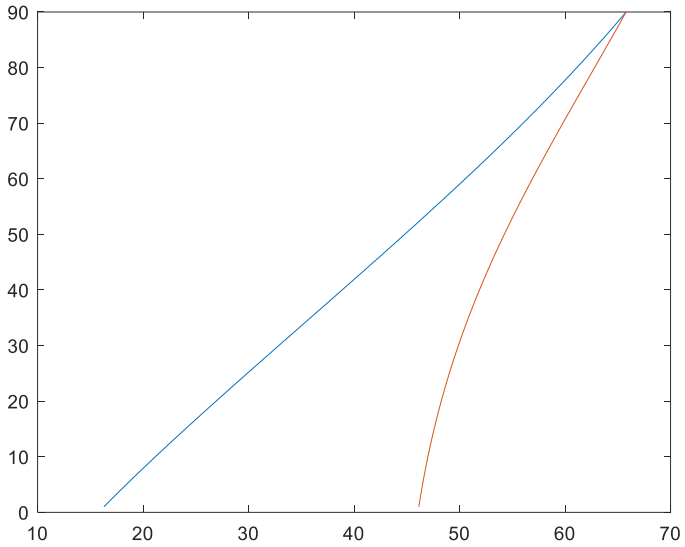


Ilustración 58 Solución primer Eje

Para el primer eje hay una recta y otra grafica no recta, se descartará la gráfica no recta y queda únicamente la azul lo primero que vemos es que en la posición original tendremos un ángulo de offset de 16.2866 la gráfica es lineal con lo cual sacaremos el ángulo con una interpolación lineal entorno al punto de trabajo sabiendo que X es cuanto se mueve el motor y el eje Y cuanto se mueve la estructura sacamos una relación de que por cada 10º de la estructura los motores se mueven 21.1607-26.9172=5.7565º es decir que si movemos el motor entre 0 y 10º teniendo como posición inicial el offset de 16º podremos mover la estructura entre 0º y 17.3716º.
Para el otro eje simplemente cambiamos los datos de distancias del código añadiendo el offset de D1 y volvemos a ejecutar el programa.

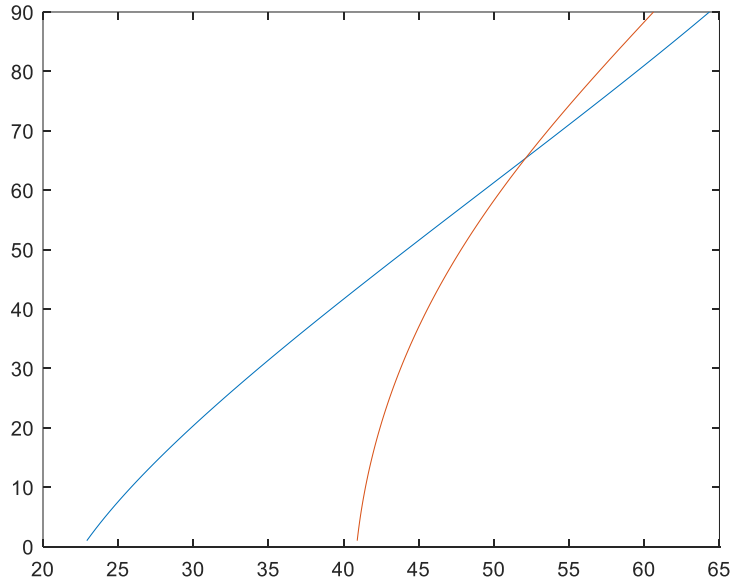


Ilustración 59 Solución segundo eje

De nuevo hay que descartar la solución naranja y únicamente queda la azul que se puede aproximar a una recta luego al desplazarnos subimos el offset a 22.9146 º una buena zona para calcular su pendiente es entre 10 º y 20 º 25.8638-29. 8492=3.9854º en este caso nos moveremos entre 0 y 25º o lo que es lo mismo cada grado del servo serán 2. 5º de la estructura.

Par de la estructura

Para saber el par que tiene que aportar el motor usaremos el montaje en inventor y miraremos los resultados del programa.
Para montar el sistema iremos con el place introduciendo cada parte de la estructura y para los rodamientos podemos introducirlos buscando en la librería propia de inventor por bearing y seleccionando la medida que encaje completamente con las piezas.
Una vez están todos los elementos del sistema podemos unirlos con la función contener esto unirá los elementos entre si permitiendo el movimiento en la dirección que queramos nótese que para una buena solución deberemos introducir tantas uniones entre los elementos como grado de la unión que tengamos para dejar la pieza bien fija lo que significa que para los rodamientos solo haremos dos uniones la del eje externo con la estructura externa y la del eje interno con la estructura interna sin embargo para las esquinas necesitaremos cuatro uniones dos para cala ladoy así dejar las esquinas bien definidas y colocadas para ello es prácticamente obligatorio usar el ratón moviendo el cursor por encima de los diferentes elementos aparecerá dibujado en color verde en la pantalla las uniones deseadas primero pulsaremos el elemento que queremos que se desplace y después donde queremos que se desplace con lo que uniendo todas las piezas tendríamos un modelo en ordenador de la estructura.

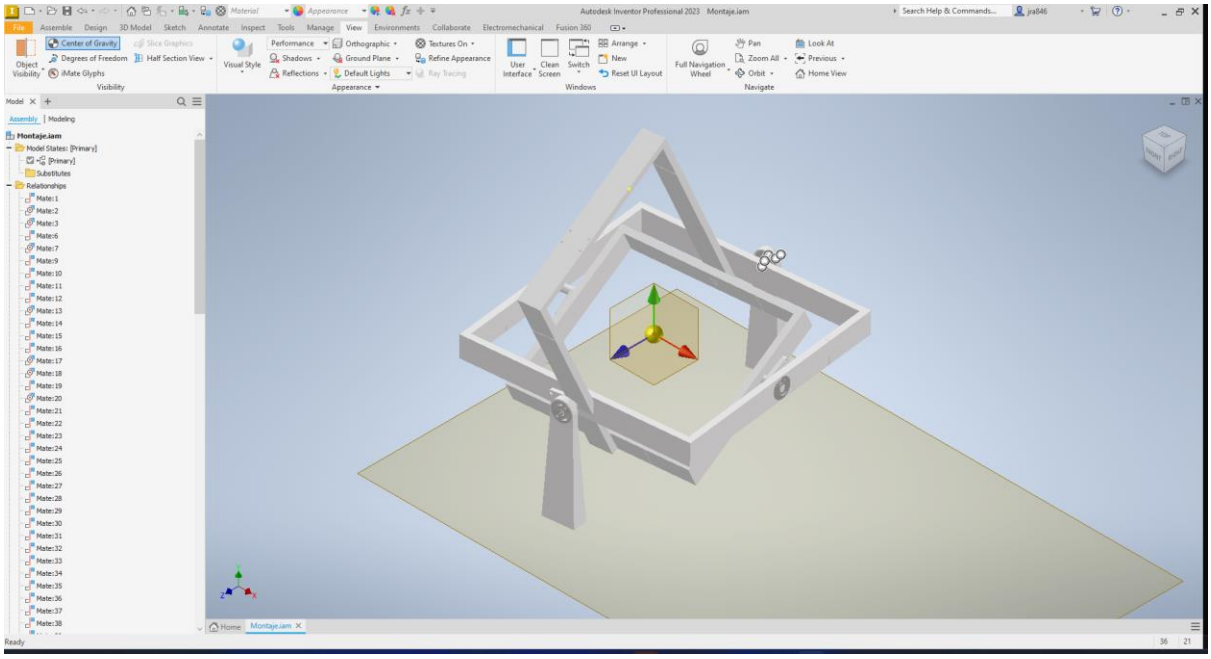


Ilustración 60 Montaje en inventor completo

Este modelo sirve para comprobar que todas las piezas estan bien diseñadas si no fuese asi el sistema daria error a la hora de hacer la union, además podemos controlar variables como el tiempo de vida de los rodamientos o el analisis dinamico del sistema.

En la pestaña de ver nos permite el programa comprobar que el centro de gravedad es el adecuado como se ve en la ilustración esta exactamente en el centro tal como se pretende y las uniones encajan correctamente disponemos de otra opcion para ver los grados de libertad del sistema directamente sin realizar el analisis analitico anterior.

En este punto no podemos aun realizar una simulacion ya que inventor necesita saber los materiales de la estructura para ello podemos seleccionar todas las barras del sistema y en la barra horizontal superior podemos poner como material PLA/ABS otra opcion es añadir nuestros propios materiales ya que cada rollo de PLA tiene unas propiedades distintas y en distintas aplicaciones puede llegar a afectar el resultado de la simulación.

Ahora simplemente llendo a la pestaña de entorno añadimos la fuerza gravitatoria en el centro de gravedad ya que el proposito de este analisis es detectar cuanto es el par minimo para mover la estructura una vez tengamos nuestra fuerza tenemos que añadir los pares del sistema para que la simulacion lo tome como referencia estos pares seran los distintos ejes que salen de las barras del marco para inciar la simulación en el entorno de simulacion le damos a simular inventor al ser un programa muy avanzado creará una malla de puntos correspondiente a la estructura y asi poder realizar un analisis de elementos finitos el problema que se presenta en la redacion del proyecto es al tener una estructura modular donde las piezas se introducen unas en otras y no ser interpretadas por inventor como tornillos el programa crea unas 20 mallas entrelazadas y calcula como todos esos puntos interracionan entre si lo que se resume en un tiempo de compilacion enorme pese a estar usando un ordenador bastante potente inventor practicamente no toca la tarjeta grafica especializada en este tipo de calculos si no que lo carga todo al procesador y sobretodo a la memoria ram ocupando un 20% del procesador y un 80 % de la ram.

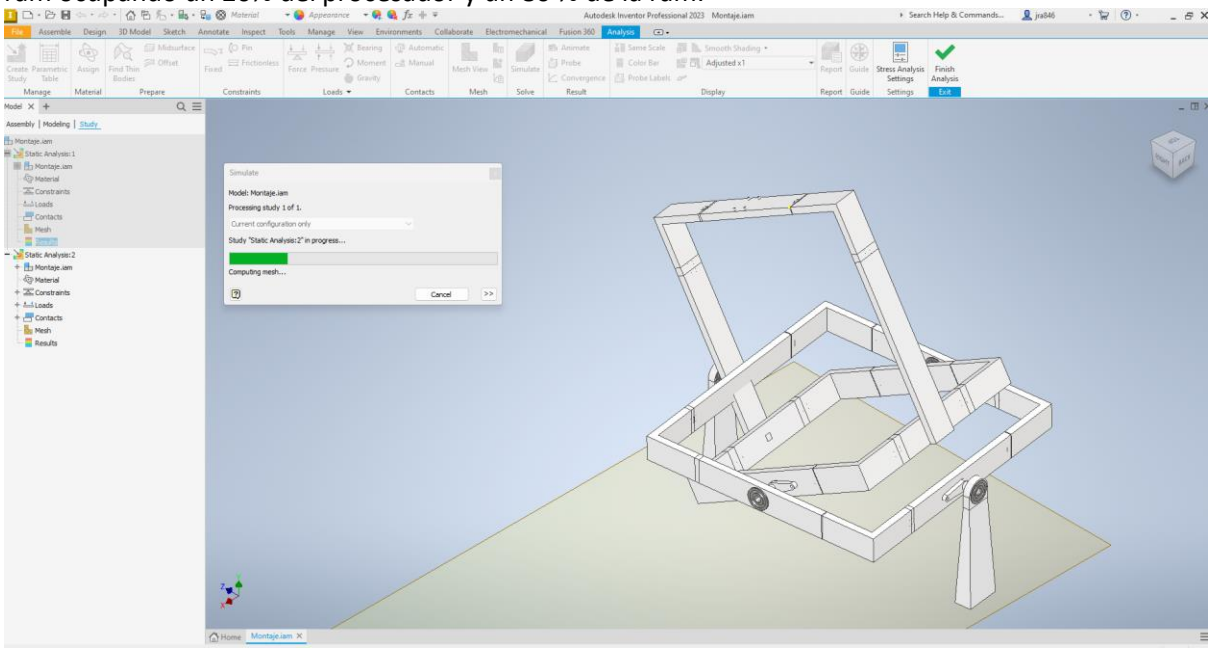


Ilustración 61 Resolución par de la estructura

Cuando esta simulación acabe pulsaremos en resultados y sobre las esquinas estará la información del par de la barra, el tiempo que el programa iba a tardar en realizar esta operación era inviable con lo cual se tuvo que evaluar dos alternativas reducir el tamaño del sistema y volverlo a hacer de cero o aplicar un análisis mucho más sencillo donde no tengamos toda la información pero podamos sacar una conclusión sobre la fuerza que tienen que tener los servomotores, para ello se utilizó el programa working model.

Montando un simple mecanismo de barras podemos obtener el par máximo de la estructura de forma muy rápida.

Ilustración 62 Mecanismo para calcular el par

El par que ejerce el motor es de 0.23 Nm que pasado a cm es de 23 N/cm lo que nos queda que necesitamos un par de 2.3 kgf.cm, aunque al añadir los apoyos que generan los rodamientos seguramente el sistema sea manejable con motores de la mitad de fuerza.

Para hacer un sistema más completo tendríamos que construir el esqueleto de la estructura y sobre ese esqueleto iniciar una simulación dinámica completa en ese apartado inventor no incluye de momento soporte para un esqueleto diseñado en PLA y tendremos que introducir medidas de las barras comerciales para que tenga todos los parámetros necesarios para hacer el análisis lo que lo hace un sistema muy complejo para este proyecto que es usado en todo tipo de industrias para la fabricación y diseño de mecanismos complejos.

Función de transferencia

Para hallar la función de transferencia de un ángulo con su aceleración angular consideraremos un sistema más parecido al Ball and Beam además linealizaremos el sistema entorno a un punto de equilibrio para así obtener esta ecuación.

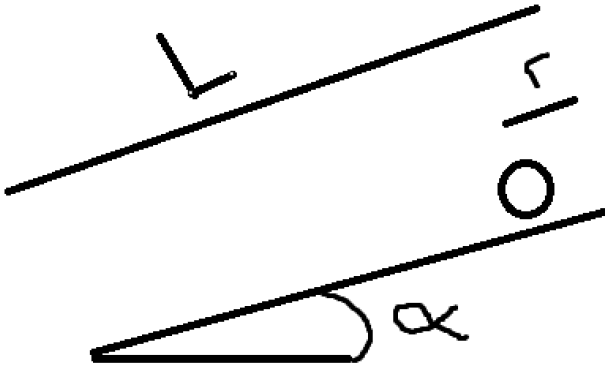


Ilustración 63 Esquema sistema de movimiento

Primero tenemos que hacer el langreano del sistema:

$$F * a = mgsin(\alpha) - mr\dot{\alpha} + J\frac{a}{R^2}$$

Sustituyendo a como la segunda derivada de la posición y ordenando la ecuación queda:

$$0 = \left(\frac{J}{R^2} + m\right)\ddot{r} + mg\sin\alpha - mr\dot{\alpha}^2$$

Esta ecuación tiene un seno con lo que nos es lineal, pero si consideramos variaciones muy pequeñas podemos linealizar la ecuación entrono a un punto quedando para alfa=0 el seno vale 0 con lo que nos eliminamos el seno quedando la ecuación:

$$\left(\frac{J}{R^2} + m\right)\ddot{r} = -mg\alpha$$

Ahora pasamos la ecuación anterior a Laplace nos queda:

$$\left(\frac{J}{R^2} + m\right)R(s)s^2 = -mg\frac{d}{L}\Theta(s)$$

EN nuestro caso D/L=1 y si hacemos la relación entre α(s) y R(s) con la ecuación linealizada nos queda:

$$P(s) = \frac{R(s)}{\Theta(s)} = -\frac{mgd}{L\left(\frac{J}{R^2} + m\right)}\frac{1}{s^2} \qquad [\frac{m}{rad}]$$

Con la ecuación anterior podemos conseguir la respuesta del sistema y así calcular los valores del PID consideraremos una masa de la bola de 2.7 gramos una d/L de 1 y un radio de 0.02 ahora nos falta encontrar el valor de la J que no es otra cosa que el momento de inercia o lo que es lo mismo 2/3*M*R^2 al ser una esfera hueca nos queda un valor de 7.2*10^-9 kg m^2 la función de transferencia resultante es:

$$G(s) = \frac{9.735}{s^2}$$

Este tipo de sistemas se conocen como sistemas de doble integrador y se pueden controlar con un PID, al tener todos sus polos en el origen el sistema es oscilador para evitarlo diseñaremos un PID que haga al sistema más estable para ello lo primero es saber su respuesta al escalón siendo esta una sinusoidal de amplitud 2 y frecuencia 2 cuando aplicamos una retroalimentación unitaria.

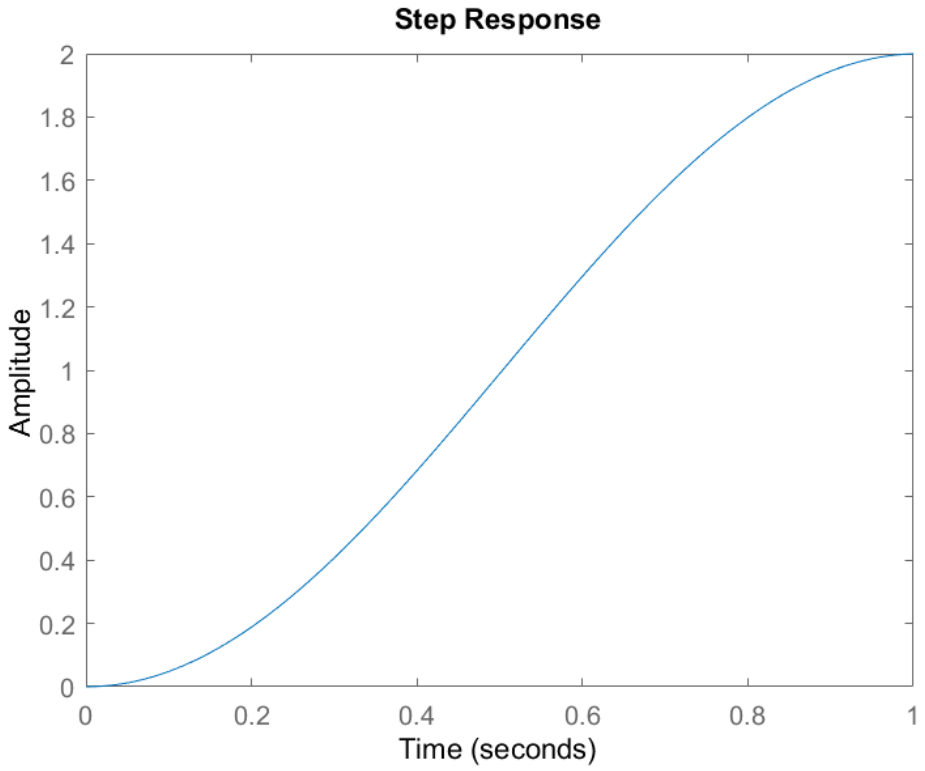
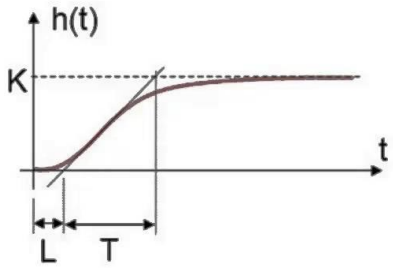


Ilustración 64 Respuesta al escalón de la función de transferencia

Si consideramos la relacion entre los angulos del servo y los angulos de la estructura como una ganancia estamos adelantando el sistema manteniendo su amplitud pero aumentando su frecuencia, para aplicar un control PID se ha optado por usar el metodo de Ziegler-Nichols.



	K_P	K_I	K_D
P	T/L	0	0
PI	0.9 T/L	0.3/L	0
PID	1.2 T/L	0.5/L	0.5L

Ilustración 65 Método Ziegler-Nichols

El valor de K=2 para hacer la recta pendiente cogemos dos puntos en la gráfica por ejemplo 0.8 y 1.2 donde la salida nos queda 0.278 y 0.36 respectivamente lo que nos da una pendiente de 4.878 que usando la ecuación de la recta.

$$y = mx + mx0 - y0$$

Y=4.878x-0.56

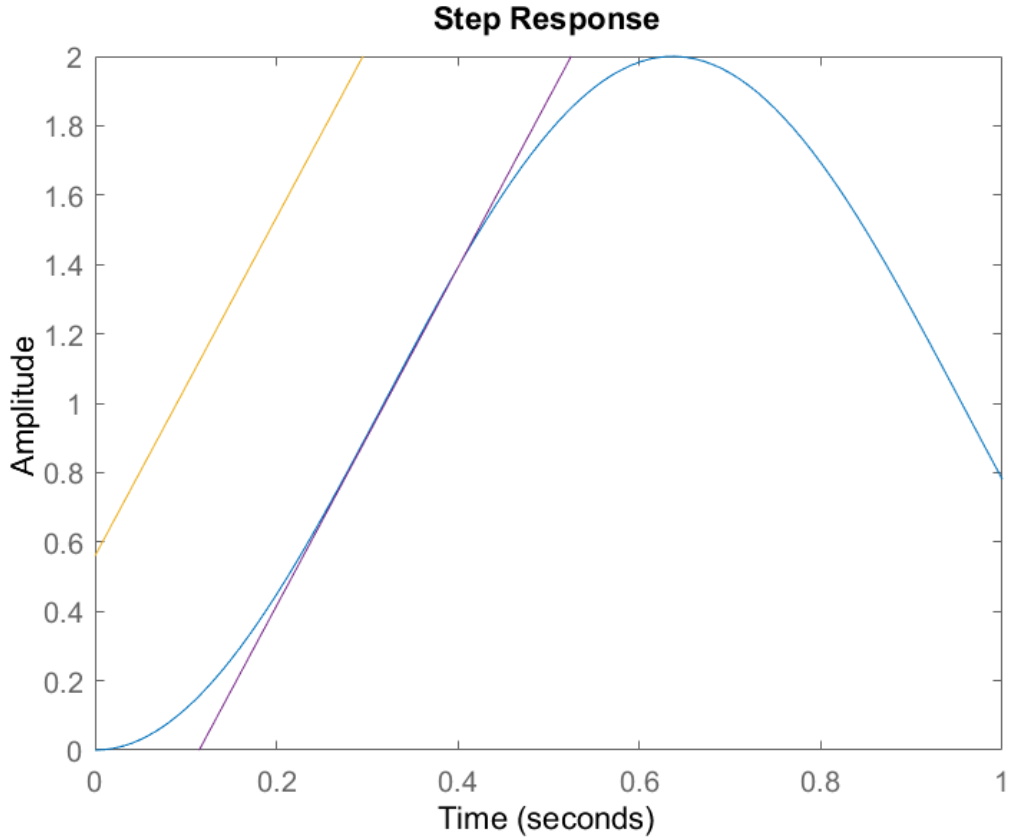


Ilustración 66 Recta de calibración

La linea que nos interesa es la morada que se junta en 0 en 0.12 segundos y llega a un valor de 2 en 0.53 segundos lo que nos da unos valores de L=0.12 y T=0.41 la sintonizacion del PID nos quedara con una salida de Kp=4.1 Ki=4.16 y Kd = 0.06.
Ahora montamos el PID y lo conectamos a la planta de forma

```
PID=kp+kd*s+ki/s
LC=2.5*PID*G/(1+2.5*PID*G)
```

Si mostramos la respuesta al escalón el sistema mejora mucho ya que ahora no es oscilador.

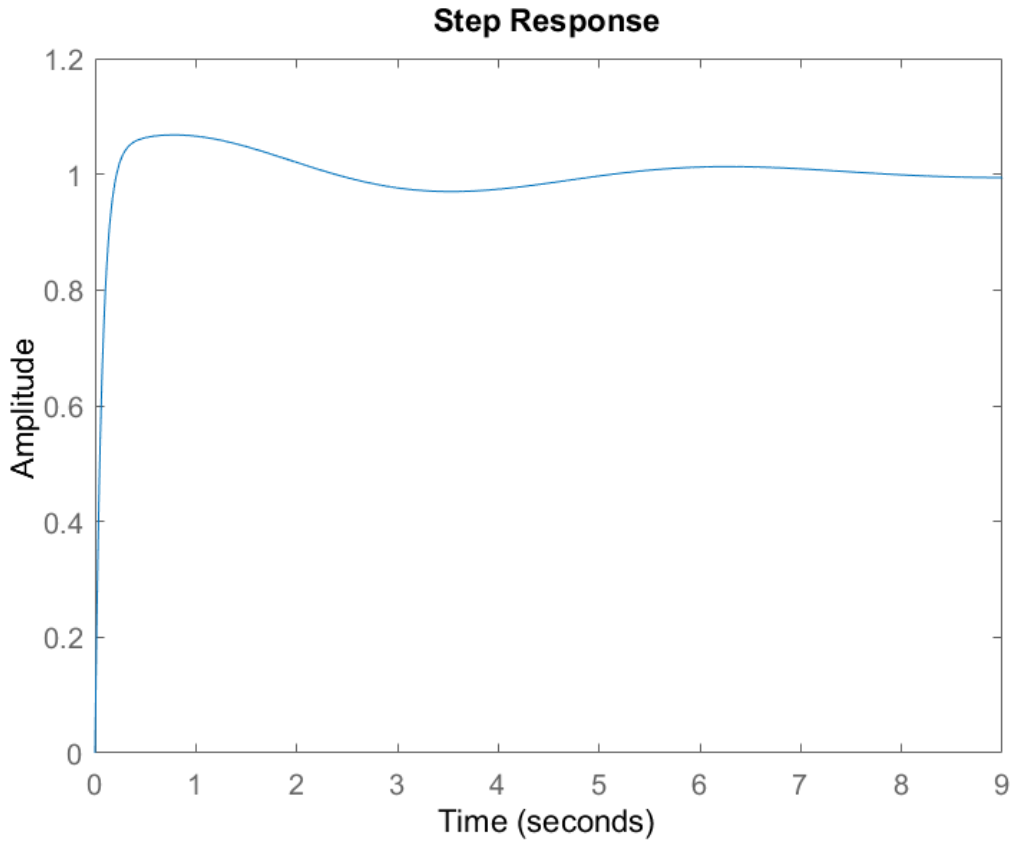


Ilustración 67 Sistema con PID

Con la informacion del escalon podemos ver muchas cosas lo primero que la respuesta es mucho mejor teniendo el pico del sistema muy cerca del 1 en 1.068 y el tiempo de subida de 0.1322 s y un sobreimpulso de unicamente el 6.8% además pasados 4 segundos llegaremos a un valor final pese a que tenga muchas obsilaciones la subida que realiza en los primeros 0.1322 s es bastante rapida y llega de 0 manteniendose todo el rato muy proximo a 1 lo cual es un gran indicador de que con el controlador le modemos dar el angulo al servomotor y sera lo que gire la bola.
Ahora podemos hacer la localizacion de los polos para saber como queda este nuevo sistema.

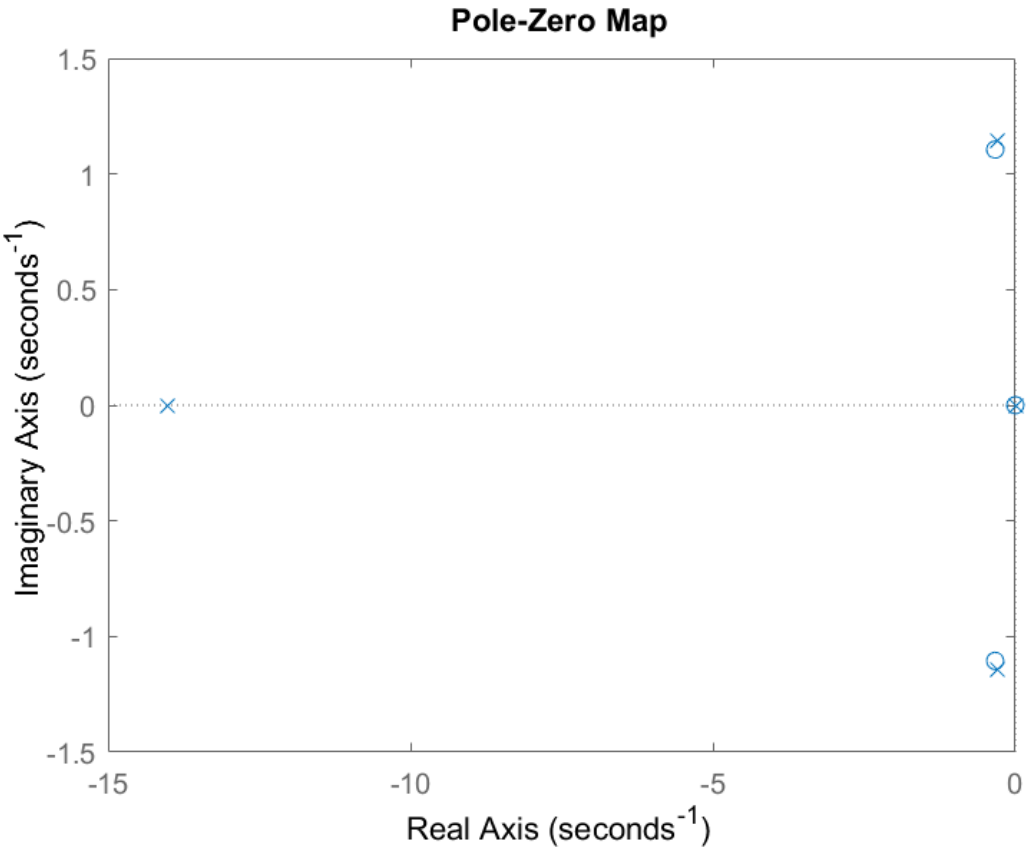


Ilustración 68 Localización de ceros y polos

Nos encontramos con un sistema que gracias al PID tiene 3 polos en el origen con sus 3 respectivos zeros cuenta tambien con un par de polos complejos conjugados con sus respectivos ceros complejos conjugados en -0.3333 ± 1.1055i y los polos complejos en -0.2979 ± 1.1407i finalmente el sistema tiene un polo no dominante en -14,0069.
La ubicacion de los polos y zeros que añade la introduccion del PID hacen al sistema controlable además para el otro eje usaremos el mismo PID ya que las ecuaciones son iguales cambiando un poco la ganancia pero eso no cambia la ubicacion de los polos del sistema que es la adecuada para realizar un buen control.

Sistema de variables de estado

Para resolver el sistema en variables de estado lo primero es elegir las variables de estado que queremos medir una forma de nuevo de simplificar el sistema es quitándonos una de las entradas y analizando únicamente un eje la función de transferencia es la misma pudiendo representar el sistema como:

$$\begin{bmatrix} \dot{r} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{mgd}{L\left(\frac{J}{R^2}+m\right)} \end{bmatrix} \theta$$

En esta representación A=[0 1 0 0] B=[0; 9.735] C=[1 0] y D=[0] si hacemos la matriz de observabilidad de este sistema nos queda la identidad de rango 2 con lo cual es sistema es observable además la matriz de contabilidad queda también una matriz diagonal lo que nos dice que el sistema también es controlable.

Está representación no nos es muy útil entonces haremos otra mejor en la cual en vez de medir el ángulo de salida vemos el par que hay que ejecutar a la estructura para ello tenemos que tomar como variables de estado la velocidad de la bola y su aceleración y la velocidad del ángulo de

inclinación de la estructura y su derivada y tomaremos como entrada la última variable de estado y de salida la posición de la bola.
Para construir el espacio de estados partiremos de la ecuación:

$$\left(\frac{J}{R^2}+m\right)\ddot{r} = -mg\alpha$$

Y rellenaremos el resto de la matriz A basándonos en ella con lo que nos queda un espacio de estados:

$$\begin{bmatrix} \dot{r} \\ \ddot{r} \\ \dot{\alpha} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-mg}{\left(\frac{J}{R^2}+m\right)} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ \dot{r} \\ \alpha \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ \dot{r} \\ \alpha \\ \dot{\alpha} \end{bmatrix}$$

La matriz A es [0 1 0 0; 0 0 9.735 0;0 0 0 1;0 0 0 0], la matriz B nos queda [0 0 0 1]' y C queda [1 0 0 0] las matrices de observabilidad y controlabilidad quedan de rango 4 con lo cual el sistema sigue siendo controlable y observable.
La respuesta del sistema corresponde a la de un integrador doble y es:

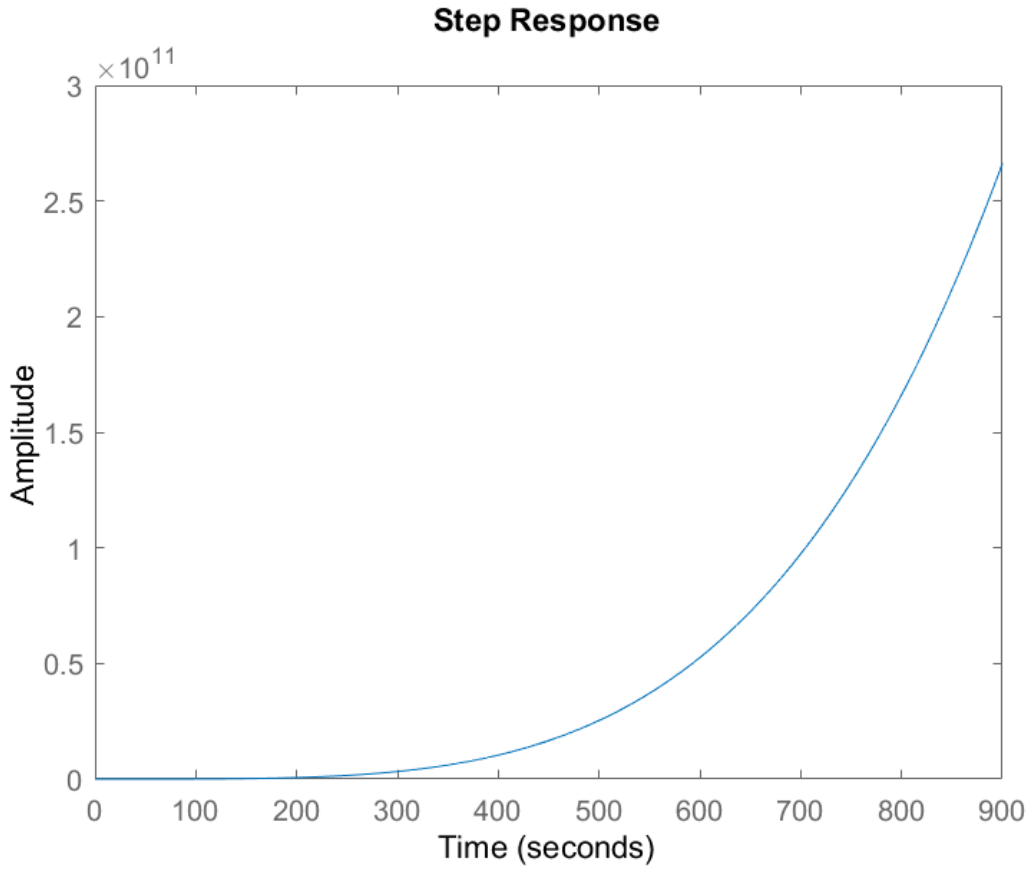


Ilustración 69 Respuesta sistema con variables de estado

El sistema responde igual ante el escalón que el calculado anteriormente en la función de transferencia, pero antes no podíamos saber si este era controlable u observable y gracias a la representación en variables de estado sabemos que lo podremos hacer mejor o peor, pero lo podemos controlar y observar.

6.DISEÑO DE CONTROLADORES

Anteriormente se ha explicado los bloques del sistema y nos hemos visto con el problema de que hay muchas formas de llegar a la solución y dentro de esas formas hay una gran variedad en el mercado de sistemas en este apartado se va a dedicar a explicar más en concreto que sistemas se han elegido y que se ha tenido que hacer para que todos los sistemas funcionen entre sí.
En este punto se analizarán cada bloque del apartado 4 donde vimos todas las posibilidades para los distintos bloques y se elegirá en concreto que sistema se usará gracias a todos los datos del apartado anterior.

Controlador Principal

Este controlador se encargará de recoger la entrada y la retroalimentación para calcular el error y mandarlo al regulador aunque un microcontrolador sea capaz de hacer esto se necesita más potencia para calibrar y localizar la bola en la imagen por ello se ha optado por un ordenador monoplaca entre todos los disponibles en el mercado se ha elegido una raspberry pi 4 de 4 4gb de RAM memoria más que suficiente para tratar las imágenes el mercado actual hace imposible adquirir una pero cualquiera de sus clones como la banana pi funcionaria bien.

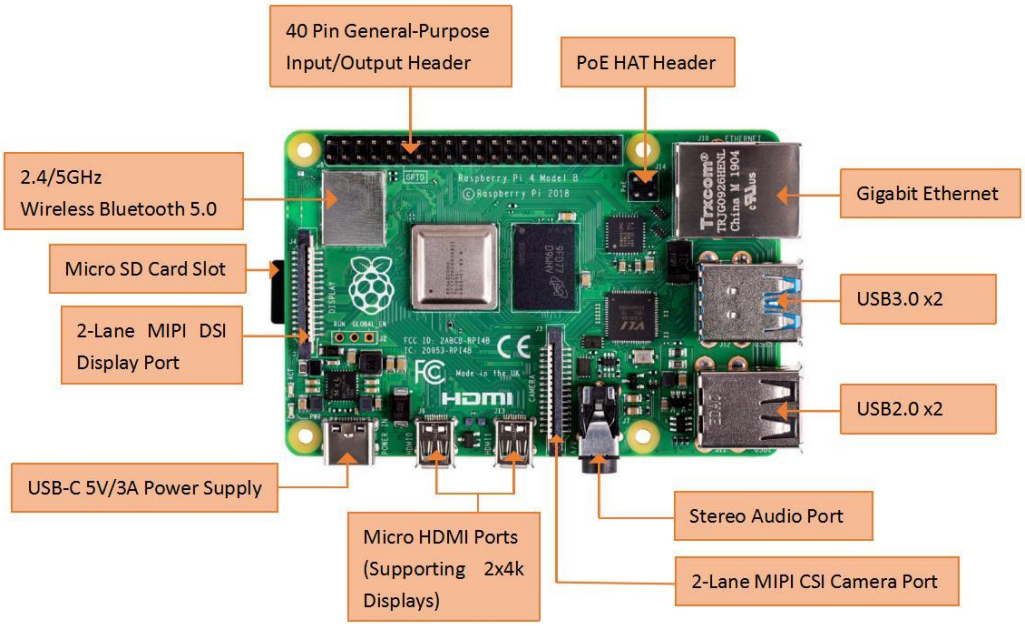


Ilustración 70 Puertos Raspberry pi 4

Analizando su hardware nos encontramos con una parte para introducir una memoria en la cual poder guardar la señal de entrada y si es necesario una imagen de calibración así como el sistema operativo y todos los programas necesarios para llevar a cabo el control además contiene 4 puertos USB donde podremos conectar el teclado y el ratón para manejar el sistema o incluso una webcam también cuenta con una salida HDMI lo que nos permitirá conectarnos a cualquier pantalla y así ver las imágenes que se están tomando y las posibles correcciones que hay que realizar en el código para mejorar el resultado, cuenta con conexiones inalámbricas wifi y bluetooth así como salida de audio.
Lo que realmente usaremos para adquirir y manejar los datos son el puerto de cámara donde adquiriremos los datos con una cámara CSI y la parte de GPIOS pins en ellos podemos hacer muchas cosas podríamos directamente dirigir los motores o mandar la señal a un microcontrolador externo que maneje la señal esto último es lo que se propone en el proyecto.

Microcontrolador y protocolo de comunicación

Antes de elegir el microcontrolador a utilizar es importante elegir correctamente el protocolo de comunicación que se va a usar para mandar el dato de la raspberry pi 4 al microcontrolador este protocolo puede ser inalámbrico o alámbrico dependiendo de las características del sistema si queremos un sistema controlado a mucha distancia donde nos permita mucha libertad para colocar los dispositivos usaremos un protocolo inalámbrico como puede ser WIFI o bluetooth dentro de esos sistemas hay diferentes protocolos como mqtt que hacen la recepción de los datos con poca latencia y alta seguridad, en este proyecto la distancia es mínima y nos interesa que la velocidad sea prácticamente instantánea que aunque sabemos que no lo va a ser se hará todo lo posible para reducirlo siempre en cuanto no afecte a las diferentes soluciones.

Se tiene que usar un protocolo alámbrico debido a su rápido tiempo de respuesta y a una distancia reducida que podemos cumplir además normalmente al ser protocolos con redundancia son inmunes al ruido y nos permite una frecuencia de hasta 3.4 MHz mientras que en un protocolo inalámbrico estas velocidades son imposibles.

Dentro de las comunicaciones alámbricas las podemos separar en dos grandes bloques por su conexión eléctrica son serie o paralelo la ventaja de una comunicación paralelo es que nos permite una mayor distancia de transmisión y podemos mandar varios bits a la vez lo que lo hace mucho más rápido que la serie el problema es que necesitaremos muchos pines para llevar a cabo la transmisión y si tenemos varios dispositivos tendremos que tener varios pines en la raspberry pi para cada dispositivo, con la comunicación en serie podemos conectar varios dispositivos en la misma línea y simplemente sabiendo su dirección podemos mandar y recibir datos esto nos permite con una sola línea de datos controlar múltiples dispositivos a coste de ancho de banda pero ya habíamos visto que esto en las conexiones alámbricas no es un problema y si lo serie tener que conectar 20 cables para poder mandar un único bit de información, también existen sistemas híbridos que combinan una conexión paralelo con una comunicación serie que hacen una aplicación más robusta como es el puerto de la cámara. Antiguamente con los primeros ordenadores todas las conexiones eran en paralelo esto forzaba a tener grandes salas con cables para interconectar la información hoy en día existen una multitud de sistemas estándar con los que todos los dispositivos se entienden.

A continuación, se enunciarán varios protocolos serie que son muy usados y sus aplicaciones:

- RS232: es de los primeros protocolos que salieron permiten la comunicación de varios dispositivos Primero está el bit de inicio, que le permite al dispositivo receptor saber que está a punto de recibir datos. Esto es importante porque RS-232 es un protocolo asíncrono, por lo que "prepara" el dispositivo receptor para leer los siguientes datos con la sincronización correcta luego vienen los bits de datos, que pueden tener entre 5 y 9 bits, aunque lo más común es 8 bits después de los bits de datos viene el bit de paridad, que es una línea de defensa pequeña pero falible para comprobar si hay errores. El bit de paridad especifica si el número de bits de datos es par (0) o impar (1).
- RS485: es la evolución del RS232 este protocolo es muy utilizado en la industria para aplicaciones de automatización permite una mayor transferencia de datos y una mayor distancia visualmente el conector es igual pero internamente no funciona igual ya que el rs485 incorpora un par trenzado que le hace inmune a la comunicación a largas distancias.
- CAN: Este protocolo fue creado por Bosch en la década de los 80 para controlar todos los periféricos del coche y poder diagnosticar todos los sistemas utiliza un único cable y existen un conjunto de subsistemas del can utilizados para diferentes aplicaciones.
- USB: es posiblemente el sistema más conocido ya que la mayoría de dispositivos como móviles o impresora a día de hoy utilizan este sistema tiene la ventaja de ser universal con lo cual puedes diseñar un dispositivo con este protocolo y cualquier dispositivo lo va a saber leer e interactuar con él se basa en dos hilos uno con D+ y el otro con D- que a medida de los años han ido mejorando en las diferentes versiones en velocidad y distancia se conectan a unos concentradores para poder añadir una gran cantidad de dispositivos.

- Modbus: El sistema serie más popular en entorno industrial es el Modbus permite una comunicación de maestro a esclavos se utiliza en los PLCs y puede ser usado también para comunicaciones TCP para comunicarnos posteriormente a internet.
- SPI: Este protocolo es utilizado en electrónica ya que permite una comunicación muy rápida teniendo una línea adicional para la señal de reloj que permite la correcta sincronización del sistema, fue desarrollado por Motorola en 1985 para comunicar dispositivos a corta distancia. SPI usa un máximo de cuatro líneas de señal (Figura 1). El dispositivo maestro, por lo general un procesador o controlador, suministra y controla el reloj (SCK) y líneas de selección de chip (CS). La operación multiplexor completa se maneja a través de las líneas de datos Máster Out Slave In (MOSI) y Máster In Slave Out (MISO).
- I2C: este protocolo es muy utilizado en la electrónica se basa en una comunicación máster Slave donde el Slave puede envían datos al maestro normalmente se utiliza un maestro que controla diferentes esclavos se basa en 2 líneas de transmisión SCL y SDA I2C se usa a menudo con periféricos que solo requieren una comunicación sencilla y ligera con un microcontrolador, como configurar controles, interruptores de alimentación y sensores.

La raspberry pi está preparada para usar SPI I2C y USB pero por estar diseñado para aplicaciones similares a la que se va a implementar se ha decidido usar el protocolo I2C ya que necesitamos pocos datos que transmitir (únicamente el valor del error) con un maestro a varios esclavos al usar este protocolo añadiremos 2 resistencias de polarización a la línea de 3.3 V de alimentación de valor 3.3 kΩ llevando una comunicación MOSI donde el master Raspberry pi mandara al esclavo microcontrolador el dato del error que este tenga que computar. Para utilizar este protocolo los diferentes dispositivos traen librerías especializadas para usar el protocolo de forma sencilla únicamente mapeando la dirección y diciendo en que parte del código recibir el dato.

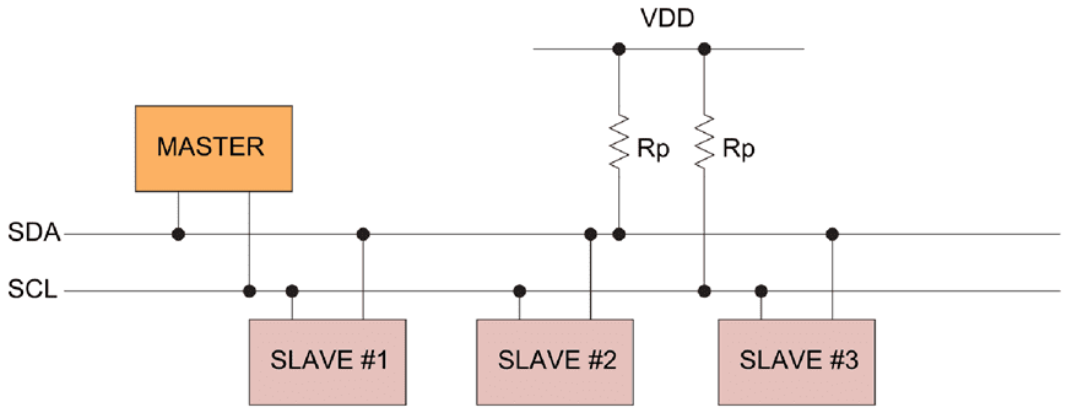


Ilustración 71 Comunicación I2C

Para el microcontrolador se plantearon dos opciones la más popular placa Arduino UNO o la ESP32, aunque la ESP32 está más pensada para aplicaciones inalámbricas esta placa es más barata que el Arduino y su entorno de programación es más profesional que el del Arduino además es más compacto y es más potente además permite en un futuro diseñar una placa personalizada para usar el chip cosas que no el Arduino no es posible.
Dentro de la ESP32 introduciremos un PID programado vía software el cual tendrá un valor de saturación de salida de 10 º para evitar mandar cambios abruptos de ángulo al motor a si mismo este microcontrolador transformara los datos de pixeles a ángulo, recibiendo datos como esclavo por el puerto I2C.

Motor

Para el motor se ha decido usar un servomotor con par suficiente para mover la estructura esto va a depender de la tensión de alimentación que le suministremos al motor y la calidad de sus componentes siendo en algunos los engranajes de plástico y en los de alto par de metal.

Concretamente se va a usar un micro servomotor todos estos son de bajo coste y son controlados mediante una señal PWM de determinada frecuencia donde cambiando el ciclo de trabajo podemos decir al motor que ángulo exacto poner la resolución dependerá de cuantos decimales puedas introducir en el ciclo de trabajo normalmente se trabaja con una precisión del ángulo ya que se modula con milisegundos y no nos permite más precisión pero modulando la señal con nanosegundos podríamos obtener precisiones de 0.01º.
Algunos de los servomotores que se pueden adquirir para esta aplicación son:

- SG90: Es el servomotor más popular del mercado contiene los engranajes de plástico se suele utilizar porque es muy pequeño y así reducimos el peso de nuestras aplicaciones pesando únicamente 9g, se suele usar alimentado a 5V tensión donde daría 1.2 kg/cm de par con su velocidad podríamos pasar de +10 a -10 en 0.033s y posee un ángulo de rotación de 180 º con este servomotor podríamos mover correctamente la estructura siempre que el peso con la cámara y el tablero no supere los 120g 1.2kg/10cm=120g.

Ilustración 72 SG90

- Hs422: Este servomotor es más caro que el anterior y mucho más pesado de 45g se define como servo de lujo, tiene también 4 veces más par es decir podemos mover objetos 4 veces más pesados lo que compensaría el peso extra del motor también es alimentado a 5 V su principal novedad para permitir tanto par es unos engranajes de hierro lubricados con aceite lo que le otorga una gran fiabilidad además permite una mayor resolución que el SG90 la parte negativa es que sacrificamos velocidad y aumentaría considerablemente el precio del proyecto.

Ilustración 73 HS-422

- MG995: Este modelo tiene un peso de 55g y un par de 8.5 kg/cm lo que permite mover masas de 850g, su ángulo de giro es menor de tan solo 120º y tiene una velocidad aún más lenta que el anterior que nos llevara realizar la operación máxima el doble de tiempo además tiene menor resolución que se compensa con un costo menor.

Ilustración 74 MG995

- MG996R: Es una versión del anterior de igual dimensiones y peso pero que permite mayor par de 9.4 kg/cm y una velocidad también un poco mayor lo que descartara el MG995 además su precio es menor haciéndolo una muy buena opción.

Ilustración 75 MG 996

- AX-12: Este servomotor es de características industriales y su alto precio como su ámbito de aplicación fuera de la industria además consume 12 A lo que añade un reto extra al proyecto para poder moverlo la ventaja es su resolución de 0. 29º se utiliza en brazos robóticos industriales para mover las diferentes uniones.

Ilustración 76 AX-12A

- MG90S: MG significa engranaje de metal es decir que es el mismo que el primero peor con el engranaje metálico lo que permite mantener la velocidad del primero que era el más rápido aumentando su par a 1.8 kg/cm que nos permite mover 180 g aumentando ligeramente el peso a 13.4 g que merece la pena por el aumento de par que nos proporciona además su precio es muy parecido al SG90.

Ilustración 77 MG90S

Finalmente se decidió por usar un servomotor MG90S que por prácticamente el mismo precio nos llevamos más potencia si nos quedásemos justos por un peso excesivo del sistema o que la velocidad no es la adecuada ya que se ha reducido con el peso podemos alimentar el motor hasta 7.2 V para ello podemos alimentar un operacional a más tensión de los 7.2 V de salida y crear un amplificador no inversor mediante 2 resistencias iguales que dando una señal de entrada de 3.3 V tendremos 6.6 V de salida que moverán más rápido la estructura.

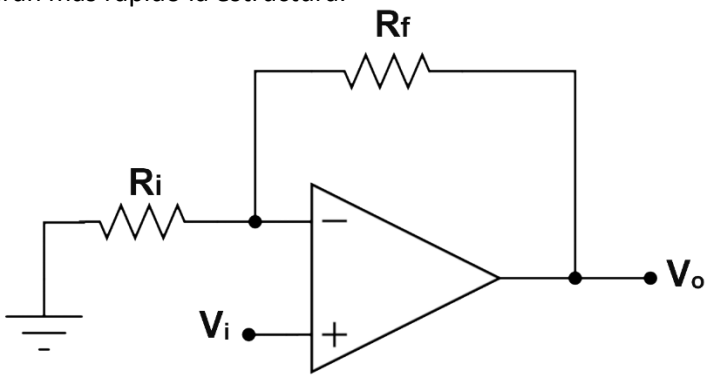


Ilustración 78 Amplificador no inversor

Cámara

Como cámaras se nos abren dos opciones una cámara USB o una cámara CSI el problema de las cámaras USB es que deben incluir si o si un microcontrolador que trasforme la señal esto no sería ningún problema si colocamos la cámara fuera de la estructura ya que nos daría igual cuanto pese pudiendo instalar cámaras con mayor velocidad y más pesadas que hagan una mejor aplicación, pero otro punto negativo es su alto coste.
La placa Raspberry pi tiene un puerto específico para usar cámaras tipo CSI estas cámaras normalmente no contienen carcasas únicamente teniendo un microcontrolador muy sencillo con muy poco peso que al estar contenido en una placa podemos fácilmente fijar a la estructura.

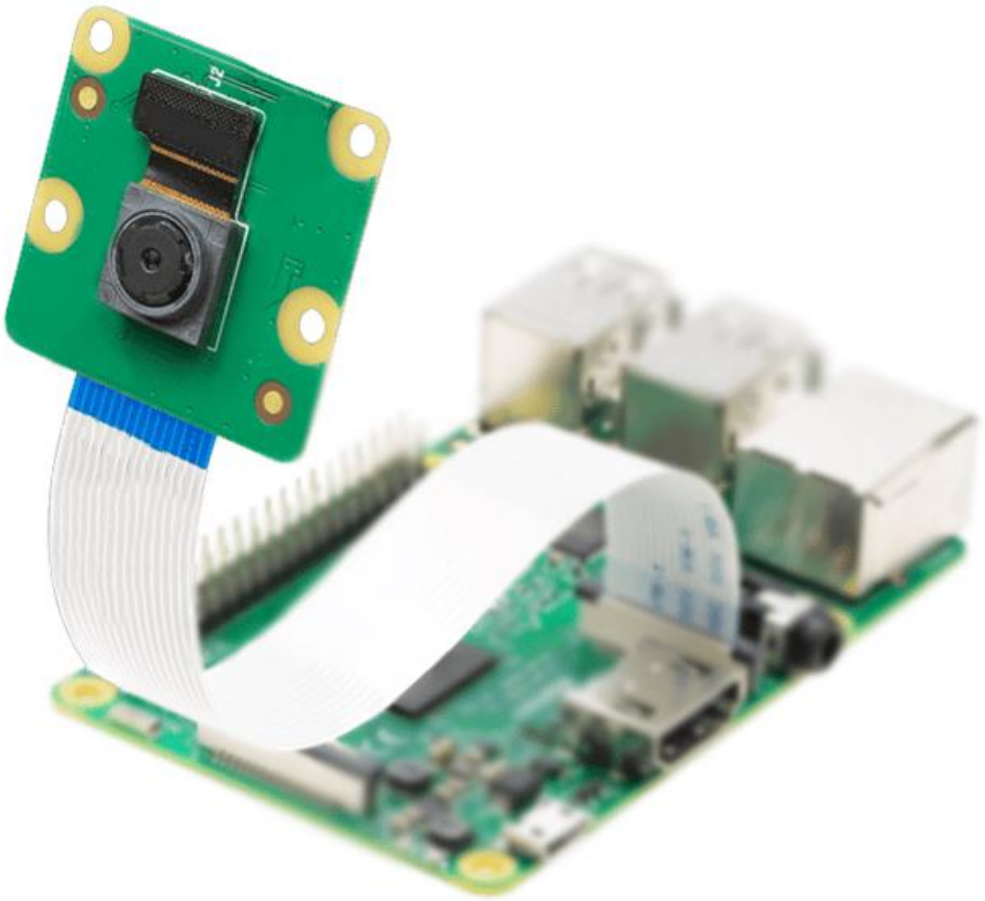


Ilustración 79 Conexión cámara controlador

Para poder usar esta cámara necesitamos saber si cumple con la velocidad para conseguir un tiempo de muestreo suficiente para la cámara oficial tiene una resolución de 5 megapíxeles usando un sensor Sony IMX 219 PQ de tecnología CMOS además su peso es de tan solo 3 g imposible para cámaras USB además si reducimos el tamaño de la imagen a 640x480p podemos grabar a 90 FPS lo que es igual a una imagen cada 0.01 segundos, con soporte oficial del sistema operativo. Existen varias versiones de esta placa, pero por su alto coste y bajo stock se ha elegido la v1.3 que la diferencia con la v2 y la v3 es la calidad del sensor de la cámara subiendo hasta los 12 megapíxeles frente a los 5 megapíxeles de esta que por otra parte es mucho más barata y no necesitamos tanta resolución del sensor para esta aplicación con lo que 5 megapíxeles es más que suficiente y con poder tener una velocidad de 90 Hz más que suficiente para conseguir un buen resultado.

7.PROGRAMACIÓN

Una gran parte del proyecto fue el uso de los diferentes lenguajes de programación para adquirir los datos y generar la señal de control, en este apartado se incluirán todos los códigos con una explicación de todos los programas necesarios, así como los valores de las variables puestos.

Estructura de los programas

En grandes bloques el sistema presenta 3 programas dos de ellos escritos en Python y uno en C al ser un prototipo la generación de la entrada se ha hecho en el ordenador principal pero el mismo programa podría perfectamente correr en la raspberry pi, pero por potencia me es más cómodo programar fuera de la placa y después clonar los programas de la nube.
Antes de empezar a explicar los programas es necesario saber en qué parte del sistema de control este cada programa y como afectara al control.

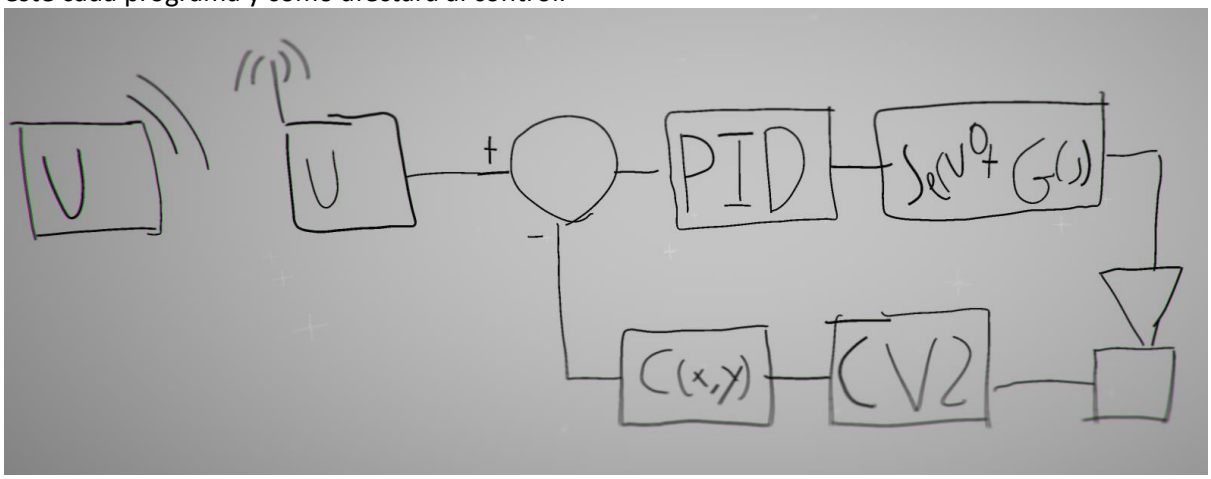


Ilustración 80 Esquema de programas

El esquema de control con los programas se puede divir en seis bloques controlados por 3 dispositivos uno siendo un servidor que unicamente transmitirera los datols de la entrada al sistema dibujado en rojo, otro dispositivo es la raspberry pi 4 que se encarga de trs grandes blolques uno que recoge la señal enviada por el ordenador otra siendo la señal de recogida de datos mediante OpenCV y los compararemos para mandarlos a la salida pero en vez de llegar directamente a la salida pasan a un microcontrolador que tendra dos tareas generar un PID para regular la salida del sistema y otra generar la salida del sistema para mover el motor, este esquematico es exactamente igual para la variable x que para la variable y.

Generación de la entrada

La entrada será una serie de puntos ordenados que la bola tendrá que recorrer para ello se realiza una especie de Paint en línea donde pintaremos la trayectoria manteniendo pulsado en un cuadro de las dimensiones de la imagen que luego corresponderá a la posición de la bola para ello usare la extensión de Python para VScode este programa está disponible para la mayoría de los sistemas operativos y es de uso libre.
Para instalarlo nos iremos a code.visualstudio.com y pulsaremos en descargar luego ejecutaremos el ejecutable y dándole a siguiente tendremos el programa instalado posteriormente nos vamos al apartado de extensiones en la barra horizontal y buscamos en el buscador Python nos descargaremos la versión que tenga más de 87000000 de descargas oficial de Microsoft.

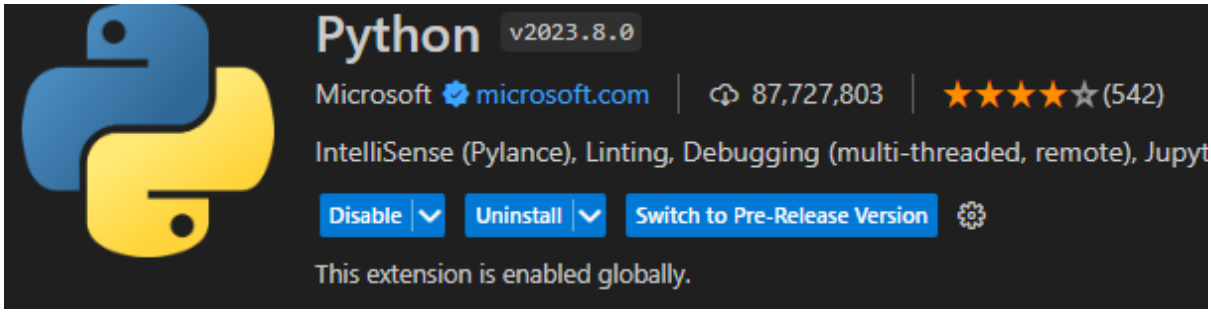


Ilustración 81 Extensión de Python

En este punto ya tenemos el programa para empezar a programar ahora lo que hace falta es generar el proyecto para eso pulsamos f1 y pulsamos python create enviroment vend esto creara una carpeta con el entorno virtual de python y todas las librerias necesarias si necesitasemos una librería extra podemos usar el comando pip.

Una vez tengamos la carpeta .venv necesitamos otras dos carpetas una static para guardar datos y otra de templetes donde meteremos el codigo del html asi mismo tenemos que asegurarnos que en el directorio principal hay un archivo llamado app.py.

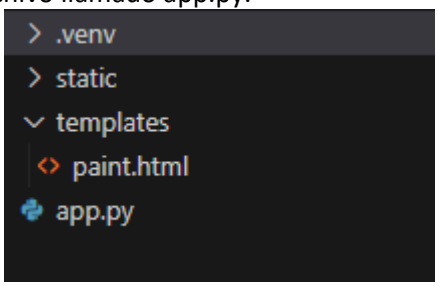


Ilustración 82 Estructura página web

El código de la entrada se separada en dos partes el front en HTML y el back en Python el back será esencial para sacar los datos que redactemos en la página.

El código en Python es:

```python
from flask import Flask, request, jsonify, render_template, redirect,
Response,json,request,url_for,make_response, send_file
import io
import os
import json
app = Flask(__name__)

@app.route('/positions', methods=['POST'])
def get_positions():
    data = request.get_json()
    positions = [(pos['x'], pos['y']) for pos in data['positions']]
    return jsonify(positions)
@app.route('/download', methods=['POST'])
def download_positions():
    positions = request.get_json()
    filename = 'positions.txt'
    with open(filename, 'w') as file:
        json.dump(positions, file, ensure_ascii=False)
    with open(filename, 'r') as file:
        content = file.read()
    os.remove(filename)
```

```
    return Response(content, mimetype='text/plain', headers={'Content-
Disposition':'attachment;filename=positions.txt'})


@app.route('/')
def index():
    return render_template('paint.html')

if __name__ == '__main__':
    app.run()
```

Al principio del código importamos las librerías necesarias para realizar la descarga de los datos y tener la aplicación corriendo además cuando ponemos from Flask import estamos importando únicamente las funciones específicas de la librería Flask lo necesitaremos para iniciar la aplicación así mismo para realizar peticiones necesitamos la función request y response, para guardar los datos en una función necesitamos las librerías json y jsonfile además hará falta la función render_template para mostrar el HTML y la función os para manejar el sistema.

Las funciones que definiremos en el api son tres la primera se llama índex se muestra nada más conectarnos a la página y lo que hace es mostrar el front del código HTML que explicare posteriormente esto lo hace con la función render_template y el HTML tiene que estar en la carpeta template.
Otra de las funciones se encarga de pasar los datos del ratón a formato json esto lo tiene que hacer en un formato que luego podamos leer por ello utiliza un string con las posiciones x e y en dos variables diferentes.
La última de las funciones crea un archivo de texto donde escribe el valor de la función anterior con el comando json.dump luego el programa lee los datos del archivo de texto y los guarda en otro json, borra el archivo de texto porque ya no es necesario y si no cada vez que descargaríamos tendríamos un contenido basura en el programa y finalmente manda las variables leídas en texto plano al dispositivo que se lo haya querido descargar.

El código HTML será el que muestre el navegador al acceder a la página para dibujar usare las instrucciones canvas de JavaScript para ello generamos un script en el cual definamos diferentes funciones y cuando termine el script haremos los cuadros necesarios para usar el ratón y los botones para poder descargar las posiciones.

```
<html>
                <head>
                    <title>Mouse Positions</title>
                    <script src="https://code.jquery.com/jquery-
3.6.0.min.js"></script>
                    <script>
                        $(document).ready(function() {
                            var canvas = $('canvas')[0];
                            var context = canvas.getContext('2d');
                            var positions = [];
                            var isDrawing = false;

                            canvas.addEventListener('mousedown',
function(event) {
                                isDrawing = true;
```

```
                                positions = [];
                                context.clearRect(0, 0, canvas.width,
canvas.height);
                                context.beginPath();
                                context.moveTo(event.clientX -
canvas.offsetLeft, event.clientY - canvas.offsetTop);
                            });

                            canvas.addEventListener('mousemove',
function(event) {
                                if (isDrawing) {
                                    context.lineTo(event.clientX -
canvas.offsetLeft, event.clientY - canvas.offsetTop);
                                    context.stroke();
                                    positions.push({'x': event.clientX -
canvas.offsetLeft, 'y': event.clientY - canvas.offsetTop});
                                }
                            });

                            canvas.addEventListener('mouseup', function(event)
{
                                isDrawing = false;
                                $.ajax({
                                    url: '/positions',
                                    type: 'POST',
                                    contentType: 'application/json',
                                    data: JSON.stringify({'positions':
positions}),
                                    success: function(data) {

$('#positions').text(JSON.stringify(data));
                                    }
                                });
                            });
                            $('#download').click(function() {
                                $.ajax({
                                    url: '/download',
                                    type: 'POST',
                                    contentType: 'application/json',
                                    data: JSON.stringify({'positions':
positions}),
                                    success: function(data) {
                                        var blob = new Blob([data]);
                                        var link =
document.createElement('a');
                                        link.href =
window.URL.createObjectURL(blob);
                                        link.download = 'positions.json';
```

```
                                    link.click();
                                }
                            });
                        });
                    });
                </script>
            </head>
            <body>
                <canvas width="480" height="480" style="border: 1px solid
black;"></canvas>
                <div>
                    <label>Positions:</label>
                    <span id="positions"></span>
                    <button id="download">Download Positions</button>
                </div>
            </body>
        </html>
```

El script empieza configurando las funciones que van a tener que estar activas todo el rato son 3 cuando hacemos clic dentro del cuadro nos ponemos a dibujar luego si movemos el ratón con el botón pulsado nos ira dibujando la línea en la pantalla pero en el momento que levantemos el dedo del botón izquierdo dejara de dibujar y pasara a formato json las posiciones que hemos recorrido con el ratón dejando el patrón dibujado en la pantalla que se reiniciara si volvemos a pulsar sobre el recuadro.
La parte final del script es una llamada a la función de descargas para poder generar el archivo que nos descargaremos y mandaremos al cliente de esta aplicación.
Una vez hemos salido del script si ejecutamos la página aparecerá únicamente con el título necesitamos bloques donde representar las funciones y poder interactuar con ellas la primera de estas es un bloque en canvas con la función de canvas donde el tamaño tiene que ser exactamente el mismo que el de la cámara con la que sacamos la imagen finalmente añadiremos un div con un botón de descarga que llamara a la función del script de descargar.

Cuando le damos a ejecutar entrando en el código principal en Python, dándole a la flecha en la parte superior derecha del código, aparece en la terminal la información que estamos corriendo una aplicación en venv de Flask sin debug nos da una advertencia que es un servidor de desarrollo y no una versión comercial lo cual no es ningún problema, las dos últimas líneas son la dirección del servidor y la forma de apagar el servidor.

```
(.venv) PS C:\Users\jrevi\Documents\web> & c:/Users/jrevi/Documents/web/.venv/Scripts/python.exe c:/Users/jrevi/Documents/web/app.py
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

La página aparece en el navegador pulsando en la dirrecion que se muestra en la terminal es la siguiente:

Ilustración 83 Página web

La pagina es muy basica ya que lo unico que tiene que hacer es dibujar en una pantalla y devolver el vector con la posiciones del raton a la vista de la imagen los datos positions solamente contienen los datos de los puntos por los que pasa el raton que sera por los que pasara el centro de la bola al final de todos esos puntos aparece un boton que al pulsarse nos descargara todos los puntos anteriores en formato json.

Para compactar el programa la placa raspberry pi es capaz de mantener de forma nativa este mismo servidor pero hacerlo de forma continuada quitaria recursos muy necesarios para realizar las operaciones de vision por ello se ha decido implementarlo a parte.

Código Raspberry pi

La cámara la controlaremos con tres códigos independientes escritos en el editor de Python que contiene la raspberry pi Thonnie con lo cual no necesitaremos ningún entorno virtual de Python hay dos maneras de introducir la entrada o bien hacemos una carpeta con una serie de posiciones previamente calculadas o accedemos a la página de la entrada y lo descargamos para ello deberíamos estar en la misma red y saber la dirección específica del ordenador.
Para una mejor solución crearemos 4 programas con diferentes funciones uno que lea la entrada otro que lea la salida que será el principal y luego uno que calcule el error y otro lo mande a los motores.

El primero de los códigos es el que lee la entrada para ello utilizamos la librería json y al igual que a la hora de mandar el archivo a descargar lo abrimos y creamos una variable intermedia llamada data luego extraemos a otras dos variables las coordenadas x e y devolviendo este valor tendremos dos cadenas de números con la posición en x y en y que recorreremos.

```
import json
def extraexy(filename):
    with open(filename, 'r') as f:
        data = json.load(f)

    # Extract x and y coordinates into separate lists
    x = [pos['x'] for pos in data['positions']]
    y = [pos['y'] for pos in data['positions']]

    return x,y
```

Este código funciona de forma idéntica al usado para generar la entrada y si queremos llamarlo lo importaremos con from leejson import extrexy esta función al tener que estar generando la localización de memoria no es una solución rápida, pero al no estar dentro del bucle de control no afectara al rendimiento siendo extraídos estos datos nada más empezar y así quedan durante todo el programa alojados en la RAM.

El segundo programa es el que calcula el error y se lo manda a los motores o por lo menos a lo que piensa que son los motores, aunque en realidad es un microcontrolador que controla los motores.

```
import numpy as np
from mandaangulo import mandaAngulo

def muevebola(cx, cy, x, y, paso):
    # Movemos el valor de las posiciones anteriores
    errorx= x-cx
    errory= y-cy
    if(abs(errorx) & abs(errory) ==0):
        return paso+1
    mandaAngulo(errorx,errory)
    return paso
```

La función tiene muy pocas líneas básicamente recoge la posición y la posición esperada recoge el error y calcula si este es 0 si es 0 pasamos al siguiente valor de entrada en el siguiente bucle pero si este no es cero mandamos el valor del error si por vibraciones es muy complicado que la bola pase exactamente por el punto y siempre tiene un error pero bajo este es el punto del código donde

podemos cambiar la sensibilidad del sistema seguramente para conseguir una solución rápida es necesario subir el valor del error a 0.1 veces el radio de la bola siendo igualmente una solución bastante buena.
Si no estamos aun en el punto que consideramos como entrada mandamos el error a los motores y nos quedamos en ese valor de entrada.
Luego manda ángulo por i2c antes de entrar en el código necesitamos activar el protocolo i2c en la placa para ello entramos en la configuración del dispositivo con el comando sudo raspi-config y en interface opción activamos el P5 correspondiente al i2c.

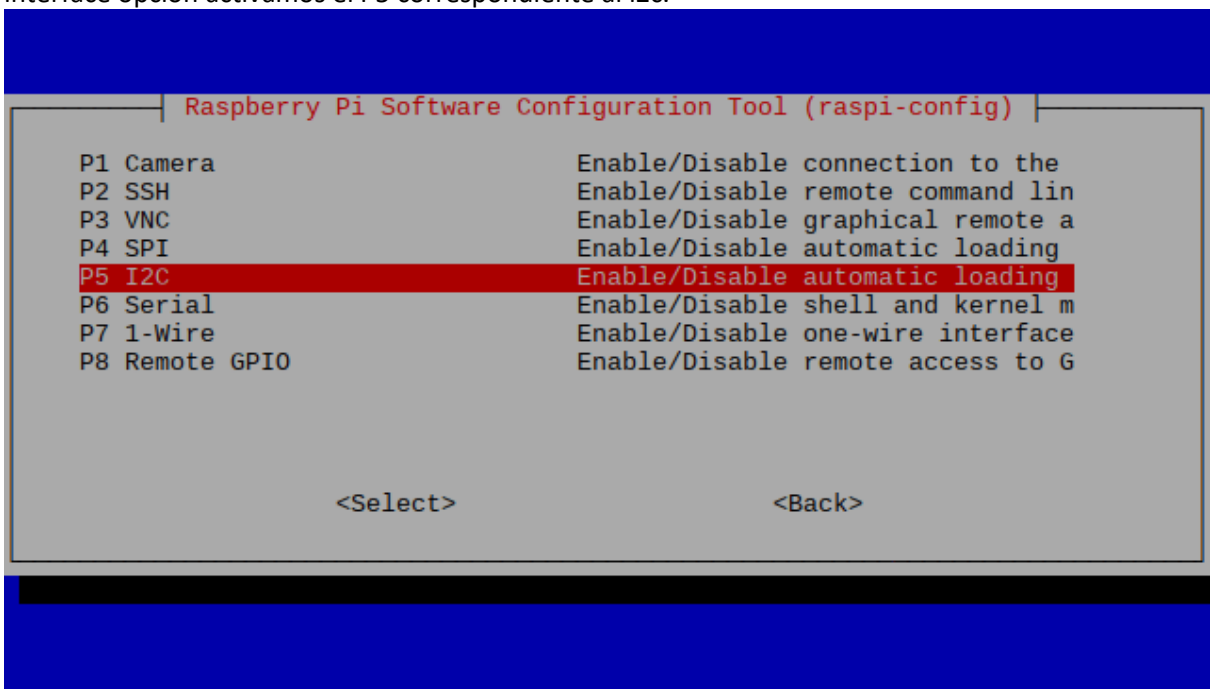


Ilustración 84 raspi-config

El programa utiliza la librería smbus para usar el protocolo i2c necesitamos saber la dirección de los esclavos que definiremos en hexadecimal después crearemos una instancia del bus en el canal 0 de la placa y con la función write_byte en la que mandaremos la posición en X al primer dispositivo y la dirección en y al segundo dispositivo.

```
import smbus

# Definir las direcciones de los dispositivos esclavos
direccion_dispositivo_1 = 0x12
direccion_dispositivo_2 = 0x34

# Crear una instancia del bus I2C
bus = smbus.SMBus(0)  # El número indica qué bus I2C se está utilizando (0 o
1)

# Enviar el primer entero al primer dispositivo
def mandaAngulo(Ax,Ay):
    bus.write_byte(direccion_dispositivo_1, Ax)
# Enviar el segundo entero al segundo dispositivo
    bus.write_byte(direccion_dispositivo_2, Ay)
```

Al no tener que definir ninguna variable al llamar al código el programa será bastante rápido y nos servirá para abstenernos de direcciones cuando calculamos el error teniendo una solución ordenada.
Finalmente, el main llamado así por ser el programa principal y más importante comunica los diferentes programas entre sí y recoge los datos de la posición de la bola este código utiliza OpenCV

para realizar la visión por computador, aunque para obtener los valores de la cámara se puede usar la librería interna de raspberry pi para el control de la cámara, pero con OpenCV puede funcionar el código para cualquier tipo de cámara.
Podemos separar el código en varias partes la primera es la definición de librerías y de variables del sistema luego pasamos a una parte de calibración de la cámara para entrar en el bucle donde tomaremos las imágenes localizaremos la bola en la misma con un filtro de color y finalmente sacaremos el valor del centro y mostraremos el resultado de la medida.
EL código entero es:

```
import cv2
import numpy as np
from leejson import extraexy
from cotrol import muevebola
paso = 0
x, y = extraexy('positions.json')


# define the minimum and maximum HSV values for the ball color
# you can adjust these values based on your ball color
lower_HSV = np.array([0, 50, 150])
upper_HSV = np.array([30, 100, 255])

# set up the camera
camera = cv2.VideoCapture(0)
camera.set(cv2.CAP_PROP_FRAME_WIDTH, 480)
camera.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

# load the chessboard image and define the size of the chessboard
chessboard = cv2.imread('chessboard.png')
chessboard_size = (4, 4)

# find the corners of the chessboard
ret, corners = cv2.findChessboardCorners(cv2.cvtColor(chessboard,
cv2.COLOR_BGR2GRAY), chessboard_size, None)

# set up the calibration parameters
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 30, 0.001)
obj_points = np.zeros((np.prod(chessboard_size), 3), np.float32)
obj_points[:, :2] = np.mgrid[0:chessboard_size[0],
0:chessboard_size[1]].T.reshape(-1, 2)
img_points = corners.reshape(-1, 2)

# calibrate the camera
ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera([obj_points], [img_points],
chessboard.shape[:-1], None, None)

# start capturing frames from the camera
while True:
    # read a frame from the camera
    ret, frame = camera.read()
```

```
    # undistort the frame using the calibration parameters
    undistorted = cv2.undistort(frame, mtx, dist, None)

    # convert the frame to HSV color space
    HSV = cv2.cvtColor(undistorted, cv2.COLOR_BGR2HSV)

    # threshold the frame to get the binary mask for the ball
    mask = cv2.inRange(HSV, lower_HSV, upper_HSV)

    # find contours in the binary mask
    contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

    # if there is at least one contour, track the ball
    if len(contours) > 0:
        # find the largest contour
        contour_sizes = [(cv2.contourArea(contour), contour) for contour in
contours]
        largest_contour = max(contour_sizes, key=lambda x: x[0])[1]

        # compute the centroid of the largest contour
        M = cv2.moments(largest_contour)
        cx = int(M['m10'] / M['m00'])
        cy = int(M['m01'] / M['m00'])
        paso=muevebola(cx,cy,x[paso],y[paso])

        # draw a circle at the centroid of the ball
        cv2.circle(undistorted, (cx, cy), 10, (0, 255, 0), -1)

    # display the frame
    cv2.imshow('frame', undistorted)

    # wait for a key press and check if it's the ESC key
    if cv2.waitKey(1) == 27:
        break


# release the camera and close the window
camera.release()
cv2.destroyAllWindows()
```

Primero importamos las librerías necesarias siendo OpenCV, numpy para hacer operaciones y las que hemos escrito anteriormente leejson y control posteriormente definiremos una variable índice llamada paso, extraeremos los valores de la entrada llamando a la función extraexy.
Como la identificación de la bola se hará por color necesitamos iniciar un filtro HSV con el valor del matiz, saturación y valor se puede definir este espacio de color como un cono donde la matriz va de 0 a 360, la saturación de 0 a 255 y el valor de 0 a 255.

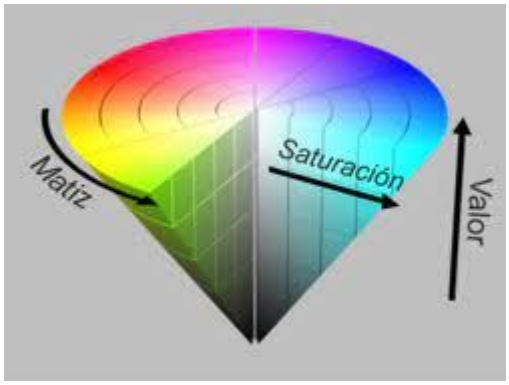


Ilustración 85 Espacio de color HSV

Si queremos quedarnos con el trozo de la pirámide que únicamente tenga el naranja de la pelota eliminando el resto de la imagen nos quedaremos entre le rojo y el amarillo de 0 a 30 grados, luego para la saturación necesitaremos una saturación entre el 20 % y el 40 % finalmente necesitamos un valor muy alto de entre un 60 % y un 100 % con este filtro en la imagen únicamente habrá pixeles de la pelota.

Una vez tenemos los límites del filtro tenemos que activar y configurar la cámara con cámara=cv2.VideoCapture esta parte se puede sustituir con la librería especifica de la raspberry pi además nos permite una mayor configuración.

Después abrimos una imagen de calibración del tablero, aunque también podríamos sacar una foto sin la bola y que esta imagen sea la de calibración así adaptarse al paso del tiempo, con esta imagen de calibración sabiendo el tamaño del tablero de 4 por 4 casillas podemos buscar las esquinas del tablero y con esos puntos sabiendo que su unión es una línea recta en la realidad podemos sacar los parámetros de calibración con cv2.cablibrateCamara

En este punto ya tenemos todos los parámetros para iniciar la adquisición de datos entramos en un bucle infinito donde primero recogemos las imágenes de la cámara con el comando camera.read() luego tenemos que calibrar la imagen para ello usamos la función undistort con los parámetros extraídos anteriormente luego aplicamos el filtro HSV pasando la imagen calibrada a HSV y usando una marcara una vez tenemos la máscara buscamos los bordes de la máscara binaria para localizar la bola si hay algún borde en la imagen significa que esta la bola lo que hacemos ahora es encontrar el centroide de la bola y separamos en coordenadas cx y cy para llamar a mueve bola y mostrar el resultado mostrando un punto en el centroide finalmente si queremos terminar la aplicación pulsaremos la tecla escape y se cerrara la imagen y el sistema mantendrá la posición.

Antes de ejecutar el código necesitaremos habilitar el puerto de cámara en la placa raspberry pi para ello entramos de nuevo al menú de configuración y habilitamos la primera opción de camera reiniciando la placa para que los cambios se hagan efectivos haciendo esto cuando ejecutemos el programa aparecerá una ventana con la imagen de la cámara y la localización de los centroides si encuentra una bola.

Código de los actuadores

Para programar los microcontroladores que manejen los motores mediante PWM para ello necesitamos descargarnos todas las librerías necesarias para ello usaremos VScode y en la sección de complementos buscaremos Espressif IDF y nos descargaremos la extensión con poco más de 382000 descargas una vez tenemos la extensión pulsamos f1 y configure esp-idf extensión y pulsamos en express seleccionando una carpeta para las herramientas y otra para esp-idf, cuando acabe la instalación podemos iniciar un nuevo proyecto con f1 show example Project y créate simple Project.
El código que explicaré en la memoria es simplemente el principal ya que el resto de las opciones vienen dadas por el espacio de trabajo, el main del programa es:

```
/*
 * SPDX-FileCopyrightText: 2022 Espressif Systems (Shanghai) CO LTD
 *
 * SPDX-License-Identifier: Apache-2.0
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_log.h"
#include "driver/i2c.h"
#include "driver/mcpwm_prelude.h"
#include "driver/mcpwm_timer.h"
#include "pid_ctrl.h"


const static char *TAG = "i2c";
#define SERVO_MIN_PULSEWIDTH_US 500  // Minimum pulse width in microsecond
#define SERVO_MAX_PULSEWIDTH_US 2500  // Maximum pulse width in microsecond
#define SERVO_MIN_DEGREE        -90   // Minimum angle
#define SERVO_MAX_DEGREE        90    // Maximum angle

#define SERVO_PULSE_GPIO             0        // GPIO connects to the PWM
signal line
#define SERVO_TIMEBASE_RESOLUTION_HZ 1000000  // 1MHz, 1us per tick
#define SERVO_TIMEBASE_PERIOD        20000    // 20000 ticks, 20ms

#define ESP_SLAVE_ADDR  0x11
#define DELAY_MS        100
#define I2C_SLAVE_SDA_IO 21
#define I2C_SLAVE_SCL_IO 22
static inline uint32_t example_angle_to_compare(int angle)
{
    return (angle - SERVO_MIN_DEGREE) * (SERVO_MAX_PULSEWIDTH_US -
SERVO_MIN_PULSEWIDTH_US) / (SERVO_MAX_DEGREE - SERVO_MIN_DEGREE) +
SERVO_MIN_PULSEWIDTH_US;
```

```
}

void app_main() {
    int error=0;
    float errorPID=0;
    int angle=0;
    float convFact=0.1;
    uint8_t rx_data[5];
        ESP_LOGI(TAG, "Create timer and operator");
    mcpwm_timer_handle_t timer = NULL;
    mcpwm_timer_config_t timer_config = {
        .group_id = 0,
        .clk_src = MCPWM_TIMER_CLK_SRC_DEFAULT,
        .resolution_hz = SERVO_TIMEBASE_RESOLUTION_HZ,
        .period_ticks = SERVO_TIMEBASE_PERIOD,
        .count_mode = MCPWM_TIMER_COUNT_MODE_UP,
    };
    ESP_ERROR_CHECK(mcpwm_new_timer(&timer_config, &timer));

    mcpwm_oper_handle_t oper = NULL;
    mcpwm_operator_config_t operator_config = {
        .group_id = 0, // operator must be in the same group to the timer
    };
    ESP_ERROR_CHECK(mcpwm_new_operator(&operator_config, &oper));

    ESP_LOGI(TAG, "Connect timer and operator");
    ESP_ERROR_CHECK(mcpwm_operator_connect_timer(oper, timer));

    ESP_LOGI(TAG, "Create comparator and generator from the operator");
    mcpwm_cmpr_handle_t comparator = NULL;
    mcpwm_comparator_config_t comparator_config = {
        .flags.update_cmp_on_tez = true,
    };
    ESP_ERROR_CHECK(mcpwm_new_comparator(oper, &comparator_config,
&comparator));

    mcpwm_gen_handle_t generator = NULL;
    mcpwm_generator_config_t generator_config = {
        .gen_gpio_num = SERVO_PULSE_GPIO,
    };
    ESP_ERROR_CHECK(mcpwm_new_generator(oper, &generator_config, &generator));

    // set the initial compare value, so that the servo will spin to the
center position
    ESP_ERROR_CHECK(mcpwm_comparator_set_compare_value(comparator,
example_angle_to_compare(0)));

    ESP_LOGI(TAG, "Set generator action on timer and compare event");
```

```
    // go high on counter empty
    ESP_ERROR_CHECK(mcpwm_generator_set_actions_on_timer_event(generator,
                    MCPWM_GEN_TIMER_EVENT_ACTION(MCPWM_TIMER_DIRECTION_UP,
MCPWM_TIMER_EVENT_EMPTY, MCPWM_GEN_ACTION_HIGH)));
    // go low on compare threshold
    ESP_ERROR_CHECK(mcpwm_generator_set_actions_on_compare_event(generator,
                    MCPWM_GEN_COMPARE_EVENT_ACTION(MCPWM_TIMER_DIRECTION_UP,
comparator, MCPWM_GEN_ACTION_LOW)));

    ESP_LOGI(TAG, "Enable and start timer");
    ESP_ERROR_CHECK(mcpwm_timer_enable(timer));
    ESP_ERROR_CHECK(mcpwm_timer_start_stop(timer, MCPWM_TIMER_START_NO_STOP));

    i2c_config_t conf_slave = {
        .sda_io_num = I2C_SLAVE_SDA_IO,           // select SDA GPIO specific
to your project
        .sda_pullup_en = GPIO_PULLUP_ENABLE,
        .scl_io_num = I2C_SLAVE_SCL_IO,           // select SCL GPIO specific
to your project
        .scl_pullup_en = GPIO_PULLUP_ENABLE,
        .mode = I2C_MODE_SLAVE,
        .slave.addr_10bit_en = 0,
        .slave.slave_addr = ESP_SLAVE_ADDR,       // slave address of your
project
    };
    i2c_param_config(I2C_NUM_0, &conf_slave);
    ESP_ERROR_CHECK(i2c_driver_install(I2C_NUM_0, I2C_MODE_SLAVE, 0, 0, 0));
    //PID Configuration
        pid_ctrl_parameter_t pid_runtime_param = {
        .kp = 4.1,
        .ki = 4.6,
        .kd = 0.02,
        .cal_type = PID_CAL_TYPE_INCREMENTAL,
        .max_output   = 100,
        .min_output   = -100,
        .max_integral = 1000,
        .min_integral = -1000,
    };
    pid_ctrl_block_handle_t pid_ctrl = NULL;
    pid_ctrl_config_t pid_config = {
        .init_param = pid_runtime_param,
    };
    ESP_ERROR_CHECK(pid_new_control_block(&pid_config, &pid_ctrl));
    while (1) {
        error=i2c_slave_read_buffer(I2C_NUM_0, rx_data, 5, 1000 /
portTICK_PERIOD_MS);
        pid_compute(pid_ctrl,(float)error,&errorPID);
        angle=(int)(errorPID*convFact);
```

```
        ESP_ERROR_CHECK(mcpwm_comparator_set_compare_value(comparator,
example_angle_to_compare(angle)));
        ESP_LOG_BUFFER_HEX(TAG, rx_data, 5);
        ESP_LOGI(TAG,"%i",angle);
        vTaskDelay(DELAY_MS/portTICK_PERIOD_MS);
    }
}
```

El programa empieza definiendo las librerías estándar de c como stdio.h y string.h luego añadimos las librerías del sistema operativo FreeRTOS, también necesitaremos una librería para imprimir la información por pantalla de eso se encarga esp_log.h, también hay que incluir los drivers de i2c y PWM para controlar correctamente el motor y el PID necesario.
Una vez tenemos todas las librerías necesitamos definir todos los parámetros constantes del sistema que dependerán del motor que estamos usando y los puertos a los que nos conectemos además definiremos un retraso de 100 ms tiempo en el que tendremos otro dato listo para computar.
Antes de entrar en el main necesitamos crear una función que sea capaz de pasar de ángulos a ciclo de trabajo haciendo un factor de conversión de grados a tiempo con las constantes.

$$T = \alpha * \frac{Tmax}{\alpha max}$$

Una vez tenemos toda la cabecera podemos entrar en el main donde primero definiremos las variables del sistema que son el error y el ángulo así como un buffer para guardar los datos del canal i2c, lo primero que hacemos es definir una señal PWM en frecuencia y fuente física del contador además crearemos un operador que conectaremos al contador y un comparador y generador partiendo del operador posteriormente ponemos el ángulo del servomotor a 0º para que siempre que se inicie empiece en esa posición.
En la siguiente parte del código diremos a la placa que tiene que hacer cuando se genere un evento en el contador o en el comparador finalmente iniciamos el contador y configuramos el esclavo de i2c con la dirección y el modo de comunicación iniciando el controlador para poder usar el i2c en modo esclavo.
Posteriormente configuramos los parámetros del PID Kp Ki y Kd, así como el tipo de calibración y sus valores máximos para la salida, creamos el bloque PID en la placa en este punto tenemos todo el sistema iniciado para entrar en el bucle.
Dentro del bucle leemos el valor del error del buffer del i2c y metemos el error en el PID con un puntero a una dirección donde queremos que se guarde la salida para pasar de pixeles a ángulo multiplicamos por 0.1 porque si el PID va de -100 a 100 nuestro motor va de -10 a 10 º lo que es un factor de 0.1 para pasar de valor PID a ángulo este valor se lo damos al comparador para seleccionar el ángulo del motor finalmente mostramos por pantalla el valor de error recibido y el ángulo que estamos mandando añadiendo un retardo al bucle de 100 ms.
Como vemos el código en C tiene muchas más líneas que en Python, pero su explicación no es mucho más larga esto ocurre porque en C tenemos que configurar e instalar todos los diferentes bloques que usemos no podemos importar directamente todo y que con una función se ejecuten tareas muy complejas pero gracias a esta configuración tan amplia podemos hacer aplicaciones más rápidas y tenemos un control sobre el guardado de la memoria gracias al uso de punteros y así podemos llegar en conjunto al tiempo de 0.1s en todo el sistema cosa que con todo el código en Python es complicado, además cualquier fallo es mucho más fácil de localizar y reparar.
Para el PID hace falta descargarse una librería extra con algunos protocolos que el fabricante de la ESP32 no considero que tengan tanta popularidad y su tamaño era importante se decidió dejar en apartado a parte se puede descargar desde GitHub este componente tendremos que incluir su dirección en el compilador y en el buscador del editor para que no nos lo detecte como fallo.
Otra opción del PID era implementar las fórmulas en una función y luego mediante un condicional conseguir los diferentes valores de saturación de las variables del PID lo que pasa es que el PID que incluye la librería es más avanzado que el tradicional y nos ahora líneas de código.

Para el PID que incluimos puede ser incremental o posicional también llamado sin retroalimentación de estados se basa en el valor del error y la suma acumulativa de este.

$$Salida(i) = Kp * error(t) + Ki * \sum error$$

Un PID posicional es menos preciso y versátil que un PID estándar, pero puede ser adecuado en situaciones donde se requiere un control básico y los requisitos de rendimiento son menos estrictos.
En esta aplicación usamos el PID incremental o con realimentación de estados siendo este un PID estándar al que le sumas su valor anterior esto le permite ajustarse mejor a la respuesta del sistema en las variaciones del error. Proporciona un control más robusto y preciso en sistemas de tiempo discreto y es menos susceptible a errores sistemáticos en comparación con el PID posicional, lo que le hace una mejor solución para este sistema.

Este proyecto está pensado para ser versátil a los mecanismos de control como los distintos bloques de PID posibles estándar mediante el diseño con operacionales en una PCB personalizada y propia, o un PID en cascada para mejorar la señal añadiendo polos extras al sistema o de una forma más avanzada un control adaptativo que vaya cambiando el PID conforme cambia el sistema.

Durante la fase creativa del presente proyecto se pensó en cambiar el PID por un control mediante redes neurales el problema es que este tipo de sistemas es muy complicado que funciones en entornos de simulación sobre todo si tenemos componentes no lineales por ello habría que construir una estructura prototipo conectarla unos joysticks y controlarlo manualmente hasta que tengamos suficientes datos para entrenar una red neuronal y sea esta la que imitando al operador consiga mantener la trayectoria deseada de una forma más natural y a partir de ese modelo pueda aprender solo durante la investigación del proyecto se descubrió que hay librerías específicas para este controlador para implementar redes neuronales que han sido entrenadas previamente con lo cual no sería demasiado complicado pasar de este proyecto a uno donde se implementen estas técnicas de control que actualmente están tan de moda y no hay ningún otro sistema como este que lo implementen añadiendo un factor diferencial al proyecto.

8.PLANOS

En este apartado se adjuntarán los diferentes planos para llevar a cabo el montaje del sistema y además se incorporaran planos de componentes externos pero que forman parte de la estructura como es la cámara y los motores así poder justificar los diferentes componentes modelados aunque lo que más importancia ha tenido en nivel de diseño y que ha definido todo el sistema es el rodamiento con lo que será la primera pieza sobre la que comentaremos las dimensiones todas las dimensiones estarán en mm aunque la escala a la que se representa en los diferentes planos no es la misma ya que se ha intentado tener una escala en el dibujo lo más grande posible sin que se salga para poder ver bien las piezas.

Rodamiento

En general todos los rodamientos se acotan igual lo que cambia es el tamaño del mismo en este proyecto se usó unos ABEC 7 lo que significa que está formado por 7 bolas y consta de un diámetro interior d de 8 mm muy importante para decidir el diámetro del eje que será también de 8mm además cuanta con un ancho T de 7 mm que será el factor determinante para decidir el ancho de toda la estructura y un diámetro exterior D de 22 mm que de nuevo definirán el alto del marco ya que este caso deberemos dejar un poco de margen para que el rodamiento tenga un poco de espacio donde asentarse.

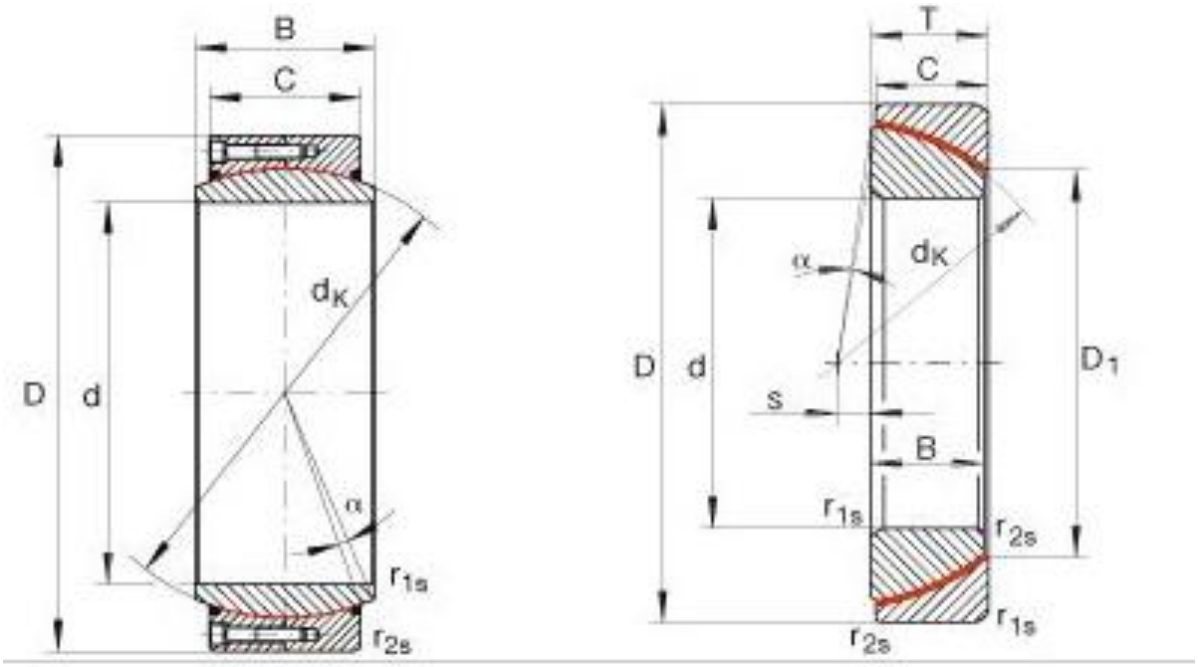


Ilustración 86 Plano rodamiento

Elegir un rodamiento de gran calidad es esencial si lo que queremos es lograr una buena solución ya que es el elemento sobre el que realmente se realizara la fuerza y debe tener la menor fricción posible además estos cojinetes nos sirven para separar el giro de la parte interna de la estructura del que se produce por la parte externa sin estos elementos cada vez que quisiéramos girar la estructura esta se caería ya que giraría solidariamente el soporte con el marco externo.

Servomotor

Como plano de servomotor usaremos dos elementos la carcasa del servomotor que tendremos que enganchar a la estructura de alguna forma y la barra que se conecta al eje la cual tiene que tener la misma longitud que la distancia del centro de la estructura al punto del que se va a tirar.
Para la estructura lo podemos buscar en la hoja de datos del modelo SG90S en la primera página aparece unas medias muy simplificadas del mismo pero que son suficiente para crear un soporte al que engancharse

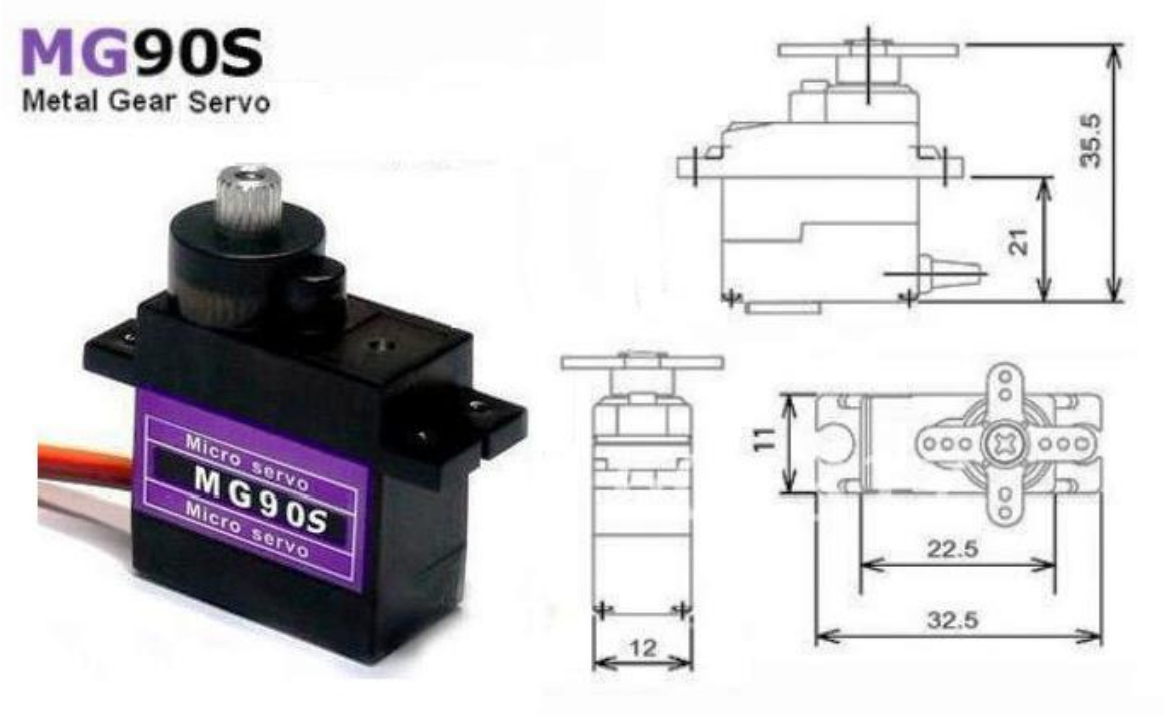


Ilustración 87 Plano servomotor

Las dimensiones importantes son la altura del servo de 35.5 mm el ancho del servo de 22.5 mm y su altura de 12 mm también tiene en la carcasa una forma de meter uno tornillos para asegurarlo en una posición fija esta plataforma se encuentra a 21 mm de altura disminuye el ancho en 1 mm quedándose en los 11 mm y tiene una anchura total de 32.5mm.
La otra parte llamada brazo es una pieza más genérica que podemos imprimir en 3D se va a usar la última dimensión lo que da una distancia de 30 mm con agujeros tabulados para tornillos M2.5 al estar colocado al ras del servomotor sin añadir más altura al montaje.

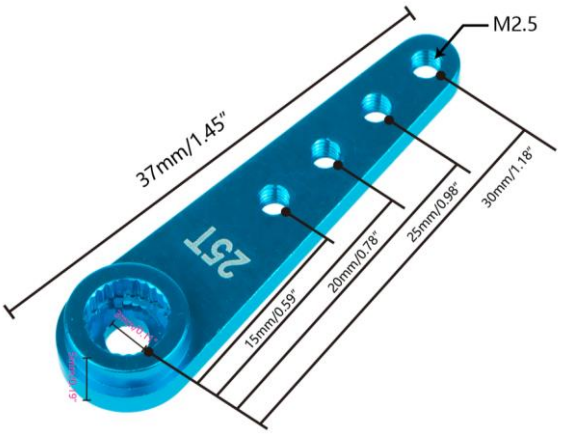


Ilustración 88 Plano barra servo

Cámara

La estructura estará preparada para usar una cámara raspberry pi v1.3 aunque todas las cámaras de este fabricante son del mismo tamaño para encajar la placa tiene una anchura de 9mm y la cámara otros 8mm de altura que deberemos tener en cuenta para crear un margen de altura para poder ver toda la plataforma además vemos como la placa pese a tener el sensor en el centro no tiene los soportes en las esquinas si no que hay dos en el centro y otros dos en las esquinas.
La placa tiene unas dimensiones de 24 mm x 25 mm todos los huecos son de 2mm de diámetro y están de forma horizontal separados del marco 2mm en altura los huecos de la esquina están a 2mm y el más centrado con el sensor está a 9.5mm, el sensor contenido en el centro de la placa tiene unas dimensiones de un cuadrado de 8 mm situado a 8.5 mm del borde y a una altura de 5.5 mm de la base finalmente la separación entre los huecos es de 12.5 mm a lo ancho y de 21mm a lo largo.
El conector que tendremos que canalizar es un cable de 1 metro de largo por 16mm de ancho que permitirá colocar la placa principal a largas distancias además su tamaño permite ocultar el cable en la estructura.

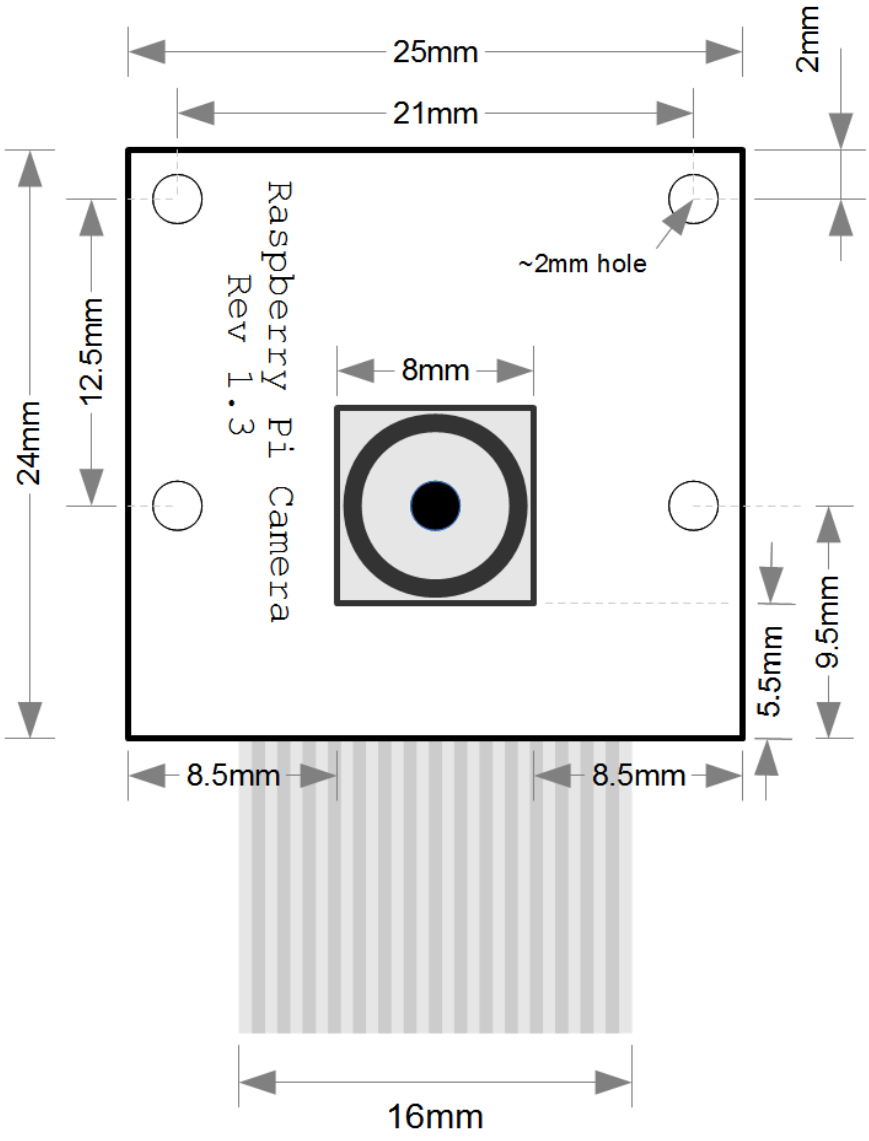


Ilustración 89 Plano Raspberry pi cámara

Aunque el plano de la placa anterior puede parecer muy simple tiene muchas medidas que permiten con facilidad diseñar un soporte para la cámara que encaje en nuestra estructura de forma sencilla y en el punto exacto para mantener la cámara en el centro.

Soporte

Lo primero que hay que diseñar es un soporte que contenga el cojinete y ponga una base más ancha que la parte superior para que al igual que los pies antropomórficos consigan una mayor estabilidad el esquema que se adjunta está a escala 3:1 y contiene únicamente las medidas necesarias para su construcción.

El diseño se realizó en dos partes primero un cubo de altura 100 mm con una anchura de 25mm y un fondo en la parte superior de 7 mm que se une a una base de 10 mm más grande en la base con lo que nos queda una base de 35 mm de ancho por un fondo de 17mm que conectaremos todas las esquinas en línea recta creando una pirámide truncada de base rectangular.

La otra parte que se acopla arriba es un soporte para el servomotor cuyo centro se localizará a 12.5 mm y necesitaremos un hueco exterior de 22 mm de diámetro al que crearemos un pequeño borde de 1.5 mm de margen lo que nos queda un medio arco de 12.5mm lo que dará muchísima rigidez.

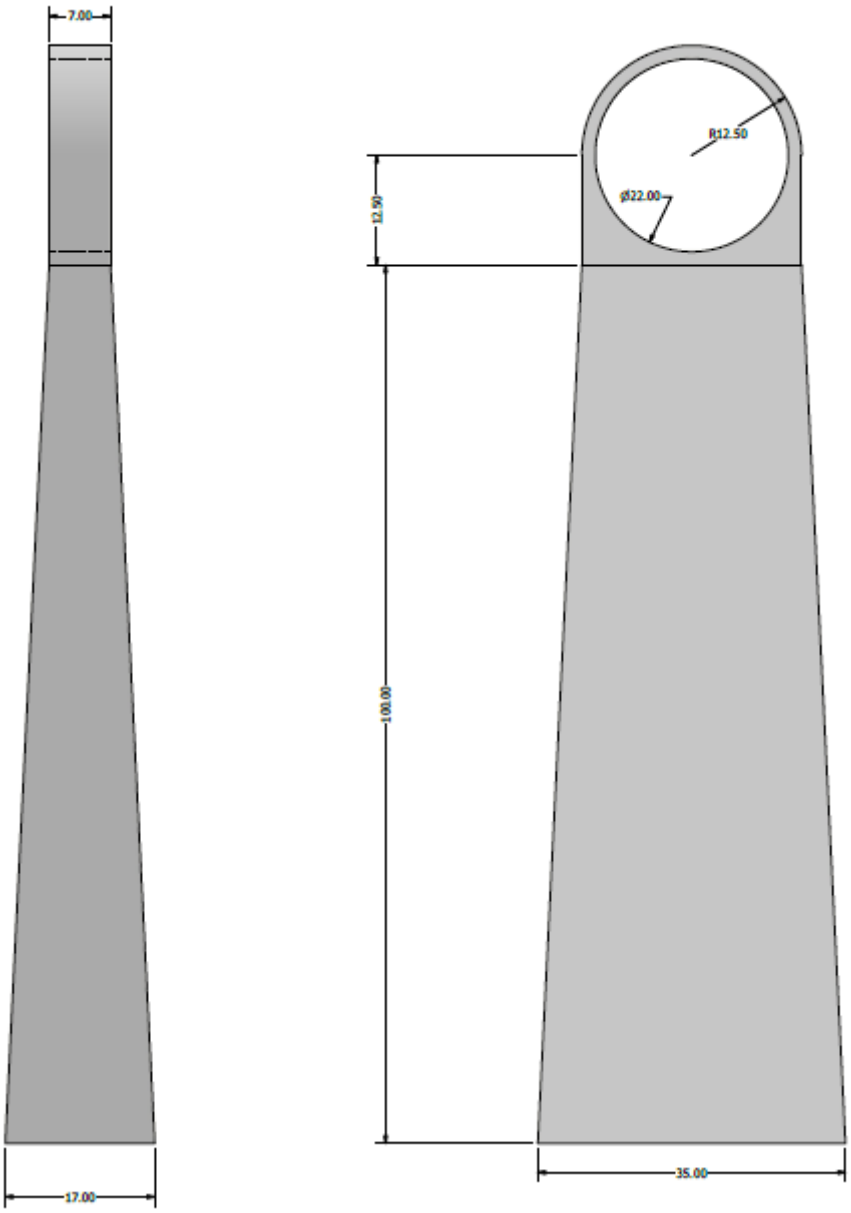


Ilustración 90 Plano soporte

Una vez fabricada la pieza para encajar el rodamiento se hizo mediante la aplicación de presión con una maza que mantiene la parte exterior fija y permite el correcto movimiento del eje, además, aunque una mayor área de la base hubiese sido mejor esta dimensión funciona adecuada.

Rodamiento externo

La siguiente parte en la cual se describirán las medias es el soporte para los otros dos rodamientos el cual se puede imaginar como la parte superior del soporte estirado para formar un marco exterior este marco con 2 esquinas tiene que tener un tamaño de marco de 240 mm proporcionando las esquinas 100 mm y la parte del rodamiento los 140 mm restantes además esta parte tiene la unión con la esquina de forma adecuada esta pieza tiene un eje de simetría vertical que permite a la estructura rotar 180º solamente tendríamos que diseñar esta pieza ya que funciona para ambos lados.

La escala del dibujo es de 4:1 y de nuevo se han puesto las medidas justas para poder fabricar la pieza, la altura total de la pieza es de 25mm y a los lados hay en una parte un cubo saliente de 5 mm de altura con una cara visible de 6 mm lo que hace una separación del borde de 9.5 mm por lado, el hueco es igual lo único que el objeto en vez de salir 5 mm entra 5 mm manteniendo una cara de 6 mm en forma cuadrada. La anchura de la pieza es de 7 mm para encajar de forma exacta el rodamiento en la anchura del borde separaremos medio milímetro el encaje de la parte de conexión con la siguiente pieza este saliente con la estructura de soporte genero la necesidad de hacer una parte de mecanizado ya que es muy complicado de encajar.

Finalmente, la estructura contiene un hueco para meter a presión el resto de los rodamientos con un diámetro exterior de 22 mm el centro de este está separado del borde de mayor tamaño 70 mm comprobando que está en el centro del marco que se fabricara.

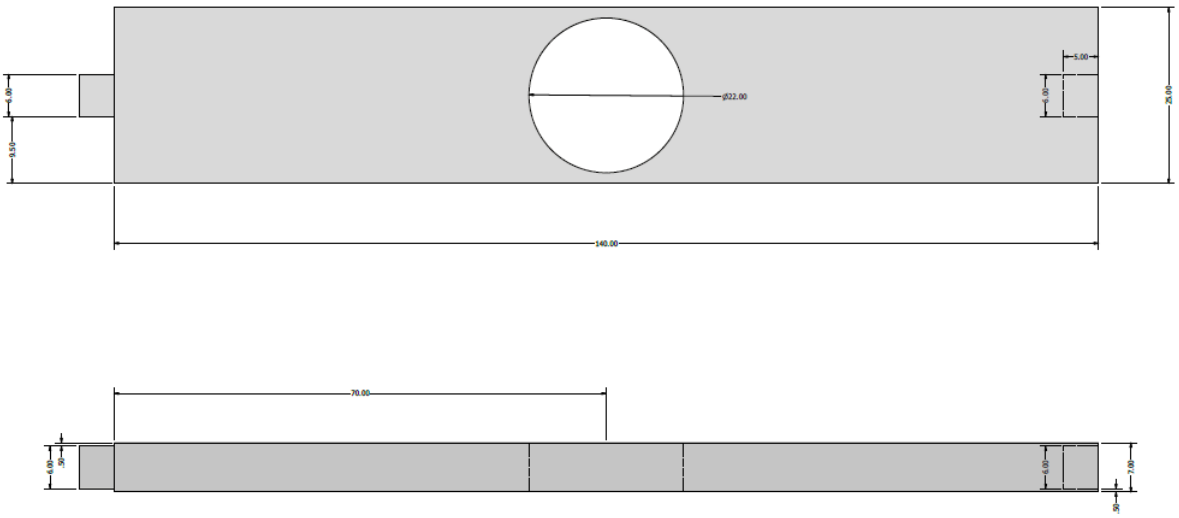


Ilustración 91 Plano soporte rodamiento externo

Esta parte es la base del resto de marcos del sistema manteniendo la misma forma de unirse con el resto de las piezas y las dimensiones de la anchura y altura del rodamiento aquí se comprueba efectivamente como la única pieza comercial que incluye la estructura es la que manda al resto ya que me es imposible cambiar esta pieza

Eje externo

La siguiente pieza que veremos es la que saliendo del marco externo encaja en la parte interna del cojinete ajustado al soporte de la estructura el saliente debe ser del tamaño justo para mejorar su resistencia y que no se genere un par muy alto en la unión con el marco que la haga partirse.
El dibujo está representado en escala 4:1 con lo que veremos la pieza más grande manteniendo las cotas del dibujo real de nuevo esta pieza se puede separar en dos la primera una barra de forma similar a la parte anterior lo único que manteniendo únicamente el hueco para encajar con la esquina rellenando la parte donde debería ir el servomotor lo que da una altura de 25 mm y un ancho de 7 mm que en el medio del borde se ha introducido un cubo de 5mm de altura y una cara de 6 mm con un hueco de 5 mm con la misma cara de 6mm de lado que encajara entre sí.
La otra parte de esta pieza es el eje externo que se conectara al rodamiento externo para ello se usa una barra de 8 mm de diámetro y una altura de 17mm justo para dejar un cm de separación y meterse al ras dentro del círculo interior del rodamiento, en ese centímetro que dejamos de separación se diseña un brazo perpendicular a la barra pero paralelo al marco en la mitad del mismo que sea el que mueva el servomotor para simplificar los cálculos la longitud de este brazo es la misma que la del servo de 24 mm y para conectarse se ha decidido usar un agujero de 3mm de diámetro que se conecta al eje principal mediante una semicircunferencia del doble de tamaño de radio 3 mm esta unión la podemos calcular de dos manearas por una parte tenemos el valor del seno de 2 mm y el del coseno de 24 mm por otra parte sabemos que está en un lado a 5 mm del centro del tablero y por la otra a 3mm del centro del agujero lo que uniendo estos 2 puntos en línea recta llegaremos al punto final.

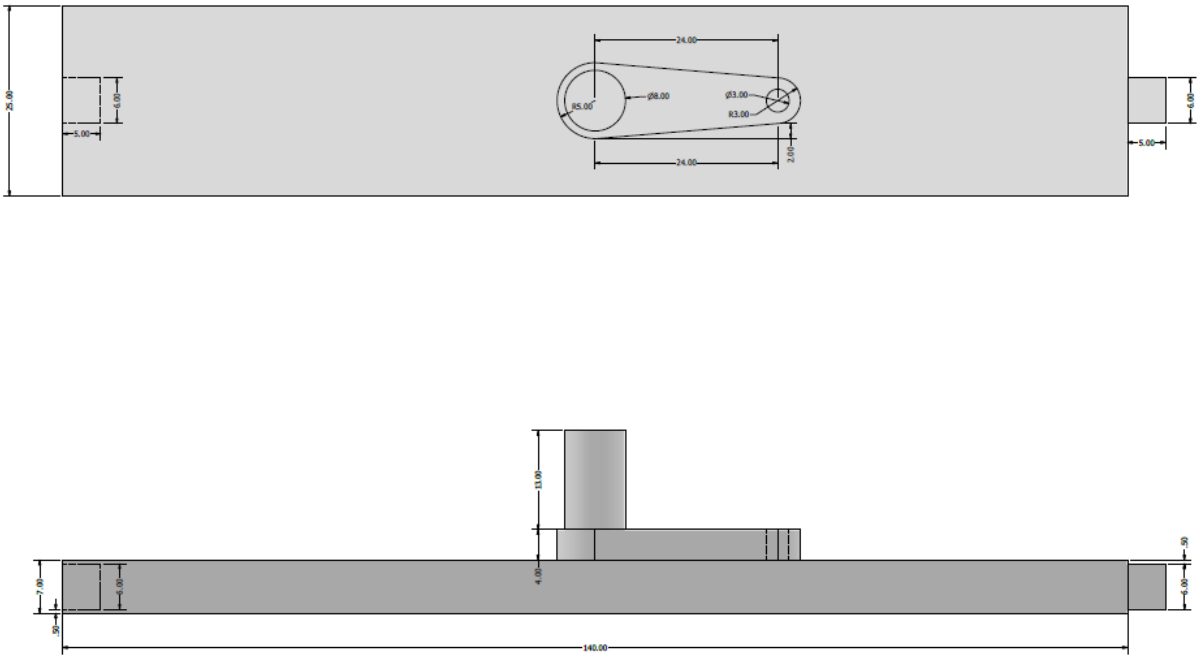
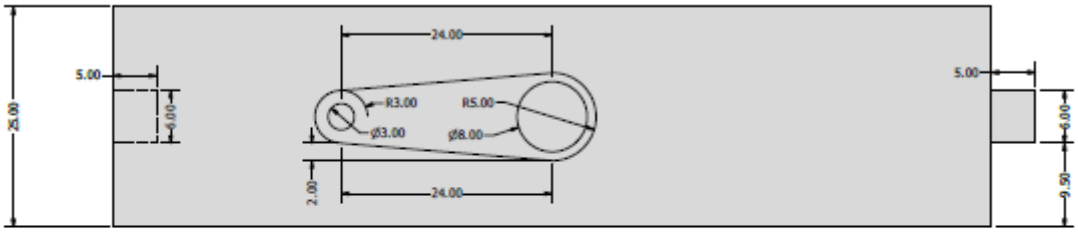


Ilustración 92 Plano eje externo

Esta pieza permite la rotación de todo el marco el hueco de 3 mm de diámetro es tan grande porque tiene que tirar de toda la estructura y la unión entre este va a tener que soportar todo el par del motor por la distancia de la barra con lo cual reducir su grosor nos arriesga a provocar una rotura de está quedándose parte dentro y siendo una reparación algo complicada que se busca evitar simplemente asegurándonos que la unión es suficientemente grande para que no se rompa.

Eje interno

Algo parecido a lo anterior, pero más pequeño se usará en el marco externo en esta parte hay que calcular muy bien cuantos centímetros nos tenemos que quitar para que la separación entre el marco externo y el interno sea de 1 cm además tiene que añadir una separación extra encajar la barra que mueve a la estructura interna con relación al brazo del servo.
La pieza esta dibujada a escala 3:1 para poder añadir las cotas de forma que haya espacio más que de sobra para definir la pieza.
Esta pieza nuevamente se puede dividir en dos partes la del eje que algo diferente a la de la parte anterior teniendo un eje en el centro de 8 mm de diámetro y una altura idéntica de 7 mm la diferencia con el diseño anterior es que esta vez la parte que se conecta a la barra del servomotor está separada unos 3 mm del marco principal manteniendo una anchura menor de 4mm esto se hace para poder hacer la circunferencia pasante y sacar el eje por detrás para asegurarlo si hiciese falta con una tuerca.
Este brazo tiene una separación entre centros de 24 mm para llegar a una circunferencia más pequeña de 3mm de diámetro a la que se le ha añadido una semicircunferencia de 3 mm de radio conectada en el centro a otra semicircunferencia de 5 mm de radio.
Para la parte del marco mantenemos la misma altura de 25 mm y fondo de 7 mm para poder hacer las esquinas de forma universal y no imprimir diferentes esquinas dependiendo del marco y se ha reducido el ancho a 100 mm esto es 2 cm por cada lado ya que el marco exterior hay que sumarle por cada lado la mitad de tamaño del borde de 7 mm con el tamaño de la separación deseada de 17 mm nos queda 20.5 mm por lado que dejando un poco menos de separación se ha truncado a 20 mm por lado menos en total 4 mm menos quedando un lado de 100 mm .



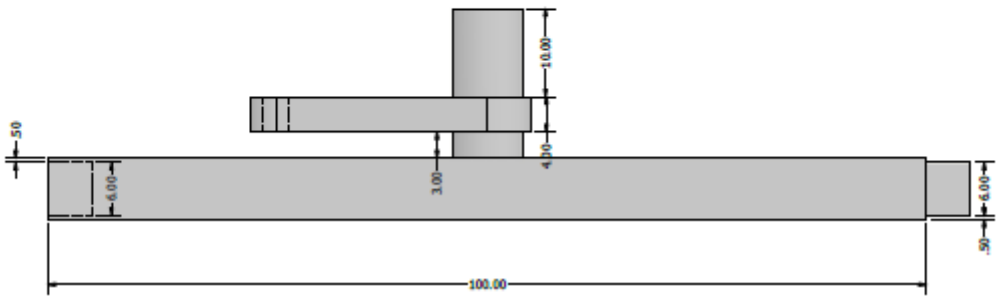


Ilustración 93 Plano eje interno

Corrección

Al imprimir la pieza anterior se hizo con un diámetro de 7 mm que, aunque en la versión final de los planos esta corregido fue necesario diseñar una pieza adicional siendo simplemente un tubo hueco de 1 mm de grosor y 1 cm de alto

El dibujo está a escala 5:1 y se compone de dos círculos uno exterior de 8 mm que se le resta el interior de 7mm con lo que introduciéndose en el eje interno permite una mayor holgura entre el eje interno del rodamiento y el eje del marco interno mejorando la respuesta del sistema como si fuese una junta de goma, pero menos flexible.

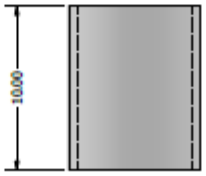


Ilustración 94 Corrección eje interno

Soporte Interno

La siguiente pieza que diseñaremos es la otra dirección del marco interno que dará la separación necesaria para albergar el tablero por ello las dimensiones serán similares quitando la parte del eje, pero añadiendo la opción de poner un marco externo en el centro la escala de este dibujo es de solo 2:1 esto es porque se ha centrado en una vista frontal para poder ver bien el perfil que genera la unión del cubo.

El lado es de 25 mm en total en el cual se contiene un cubo a 9.5 mm del borde de un tamaño de 6 mm en la parte vertical este cubo está a tan solo medio milímetro del borde con un total de 7 mm además a 47 mm del borde hacia arriba nos encontramos con un hueco de 6 mm de lado para poder poner un marco adicional además el tamaño total de esta pieza sin contar la unión es de 100mm.

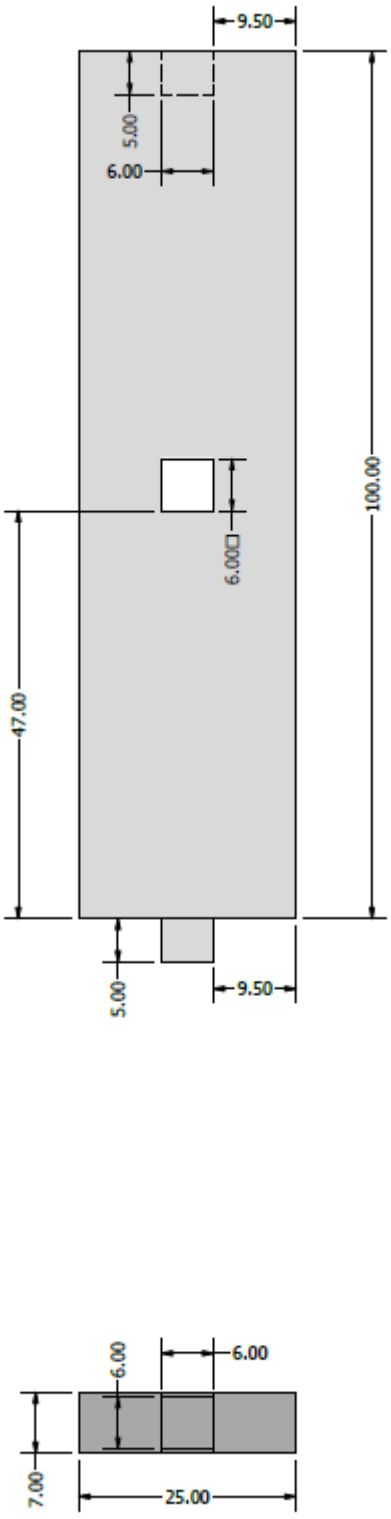


Ilustración 95 Plano soporte cámara

Esta pieza permite la conexión a 90º de un cubo de 7 mm de altura y caras de 6 mm por lado que se usara para colocar la camara es muy importante que este cubo este lo más centrado posible para que la camara este tambien centrada y no haya que cambiar el programa para alinear el centro del tablero con la medida de la camara.

Esquina

Esta pieza es la que más veces se fabricó de toda la estructura su forma en L permite que no tenga que necesitar imprimir estructuras de soporte para el saliente lo que llevaría a un mecanizando menor simplemente ligando esa parte puede entrar muy ajustado en las diferentes barras el siguiente dibujo está a escala 4:1 y se recogen las medidas para llevar a cabo el diseño de esta pieza.
Lo que hay que pensar a la hora de hacer esta pieza es su tamaño ya que esto va a afectar directamente en la longitud del resto de barras para tener en el marco interno un 50% de esquina y un 50% de marco se ha decidido hacer una longitud por lado de 50 mm además mantendremos una cara de unión de 25 mm por 7 mm en total la esquina tendrá un tamaño de 57 por 57 mm en la parte que se imprima de forma horizontal deberá incluir el hueco y en la parte vertical la perturbación para encajar en el hueco.

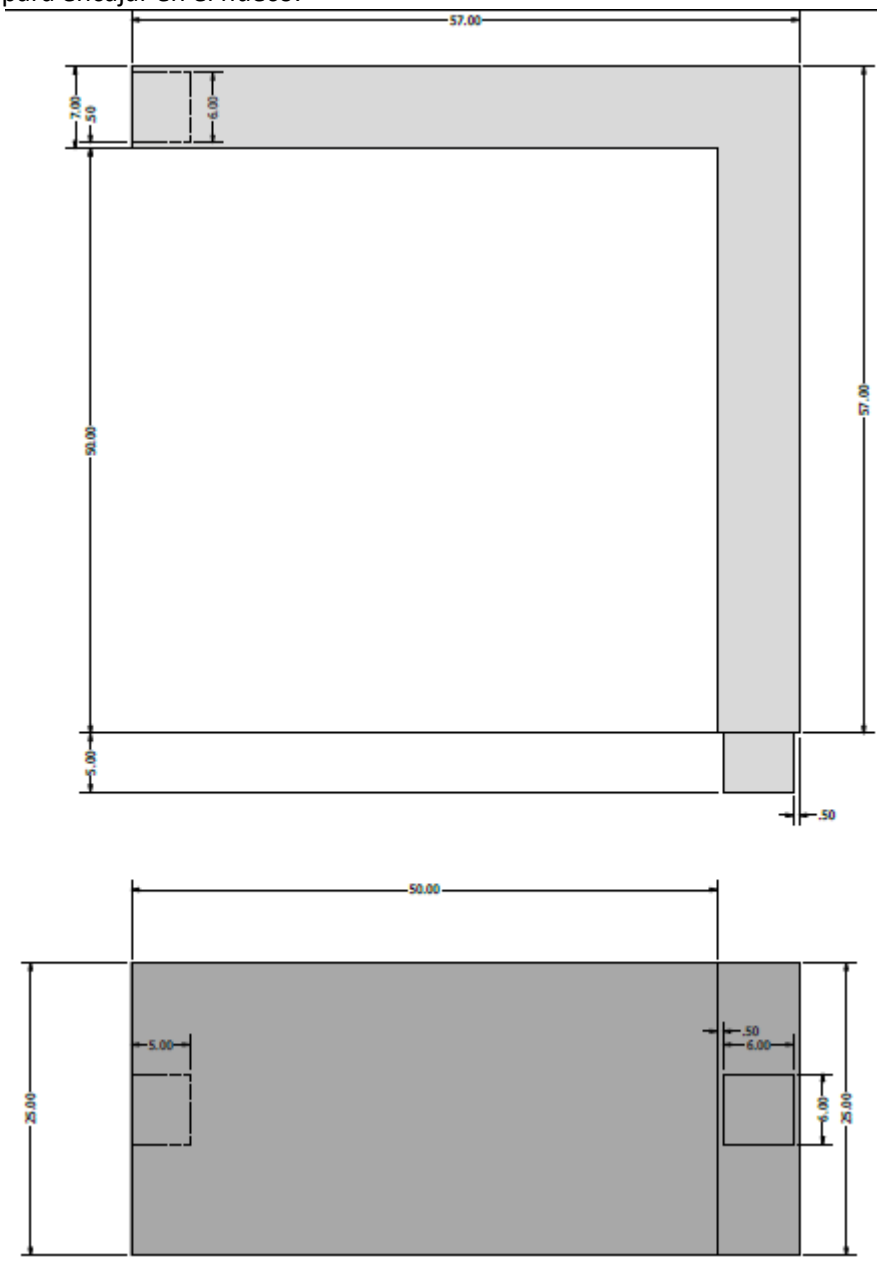


Ilustración 96 Plano esquina

Esta pieza es la base de la estructura y la que la da riguidez en un primer momento se penso cambiar el cubo por prismas como hicieron los romanos hace más de 2000 años el problema era que me costaba más el diseñar un hueco en el prisma para imprimirlo en 3D que lo que me puede llevar

mecanizar la pieza además al final no se ganaba nada en tiempo de impresión ni era más facil colocar las piezas con lo que se opto por este diseño más basico.

Barra servomotor

Para conectar el brazo del servo con el brazo de la estructura se ha decido diseñar un brazo extra que funcione como una barra que conecte el servo con la estructura para ello por una parte va encajado en el agujero de 3 mm de la estructura y por el otro con un tornillo al brazo del servomotor al dejar un rail podemos ajustar la distancia de separación entre el servomotor y la estructura y con ello el par que ejercemos a la estructura el dibujo es muy similar a la barra que usamos en los ejes pero mucho más pequeño representado a una escala 10:1.
Se puede separar en un tubo de 2.5 mm de diámetro y 6 mm de altura con un hueco de 1 mm de diámetro que ha sido estirado 10 mm formando una especie de ranura donde deslizaremos un tornillo, estos dos centros están separados 40 mm entre sí y tienen cada uno asociado un diámetro exterior que para la parte superior es 2.5 mm y para la inferior 1.25 mm uniéndose en línea recta formando un seno de 1.25 mm además la barra tiene un grosor de 4mm para que en conjunto con el pilar que sale se quede en 1cm.

Ilustración 97 barra servomotor-estructura

Teniendo una longitud de más de 40 mm es muy importante que su anchura sea de por lo menos 4 mm porque de otra forma se doblaría o partiría con mucha facilidad al ser una pieza tan pequeña no se suele hacer en 3D, sino que se usan barras de metal fabricadas por extrusión que al conectarse en la punta a una unión cilíndrica y por la cola al brazo del servomotor son más fuertes pero no es fácil encontrar stock de las mismas y es necesario añadir una configuración de tornillos más complicada para conectarse con la estructura además con lo poco que pesa esta parte no es ningún inconveniente diseñarla en 3D.

Soportes motores

Para aguantar los servomotores al sistema se han diseñado una especie de clips de ranura que encajan perfectamente en la base esto lo hace manteniendo el mismo ángulo que la base y simplemente cortando en la parte que queremos quedarnos si lo que queremos es conectarnos al marco solo hay que cambiar la ranura a una de dos líneas paralelas de 7 mm manteniendo la separación al marco de 2 mm como base y acercando la otra parte del clip, esta pieza esta dibujada a una escala 10:1 que hace que se vea muy grande pero la idea es que sea lo gusto para poder alojar al motor atado por bridas conteniendo el centro de gravedad dentro del clip.
En este proyecto solo se va a analizar su versión más compleja que es la del motor exterior ambas parten de un tablero de 12x33 mm y una altura total de 12 mm sobre la que haremos una ranura de 5 mm de alto a 2 mm del borde quedando por el otro lado una distancia de 24 mm, la versión de estos planos es únicamente para la torre separando en la parte inferior 2mm y 19mm y en la parte superior medio milímetro más por lado es decir 2.5mm y 19.5 mm esto hay que hacerlo teniendo en cuenta que la anchura es de 12 cm y que esta pieza a diferencia de la otra si tiene orientación quedándose en un lado mirando hacia fuera y en el otro hacia dentro.

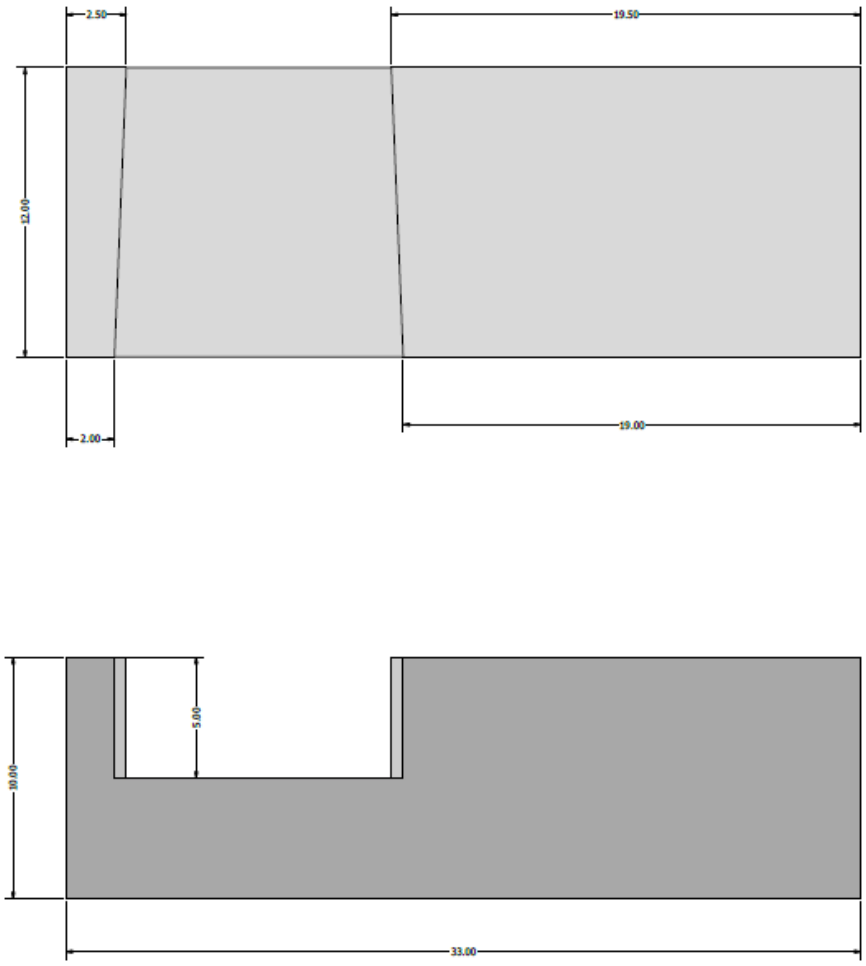


Ilustración 98 Plano soporte motores

Este es lo suficiente grande para albergar el motor con la brida siendo más corto y más de la mitad de ancho, para ello se han usado como medidas para el motor que se albergará las que aparecen al principio de este capítulo.

Soporte cámara

Esta pieza tiene que ser exactamente 100mm menos que el marco contando los 14mm de margen que incluye es decir que la pieza tiene que ser de 114 mm de longitud para que dé justo a todo el marco interior suponemos que está bien ya que es 14 mm más que el interno y 26 mm menos que el externo lo que da un total de 40 mm de diferencia que es la que teníamos al principio.
El dibujo está a escala 3:1 y en él hay que recoger la montura de la cámara y de las conexiones con las esquinas.
Para la montura de la cámara se han colocado los cilindros del medio a una distancia de 57 mm luego los otros cilindros están separados una distancia de 12.7 mm en horizontal están separados 2.7 mm del borde y entre ellos una distancia de 19.6 mm además tienen un diámetro de 2.2 mm y una altura de 2mm siendo insuficiente para poner la cámara por lo que se tuvo que implementar una solución rápida donde se incrementaba esta altura a unos 4 mm dejando un espacio libre para el conector con la estructura.

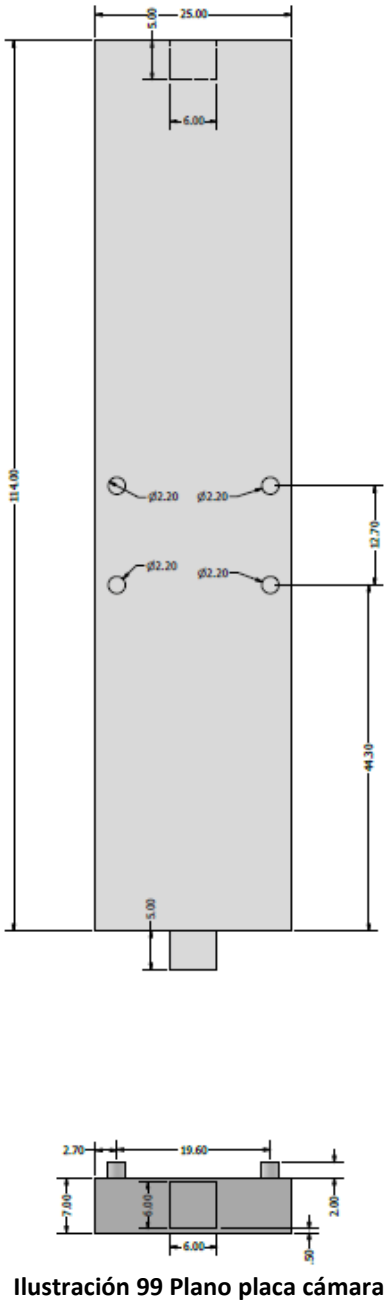


Ilustración 99 Plano placa cámara

Columnas cámara

Para poner la cámara en una altura suficiente para ver el rectángulo hemos calculado anteriormente que esta distancia tiene que ser como mínimo algo mayor que la del marco para un marco de 20 cm la altura eran 25 cm de cámara aun así nos aseguraremos bien de que la altura de la cámara es suficiente para ello con un marco de 19 cm usaremos una altura de 210 mm que cubrirá completamente la estructura con la cámara así aprovecharemos al máximo las propiedades de la cámara y no es necesario recortar la imagen además de esta pieza haremos dos versiones distintas una con el enganche hacia fuera y la otra con el enganche hacia dentro ya que al tener el otro conector en perpendicular no podremos darla la vuelta con lo cual necesitaremos el otro diseño para conectarnos al otro lado.
El dibujo está a una escala de 3:1 y no es más que una barra vertical de 185 mm de alto por 25 cm de ancho y 70 mm de fondo en la parte superior incorpora en esta versión el hueco de 5 mm de alto por 6 mm de cuadrado además para conectarse al marco interior lleva un rectángulo de 7 mm de altura con un cuadrado de 6 mm encajando en la parte perpendicular al eje interno.

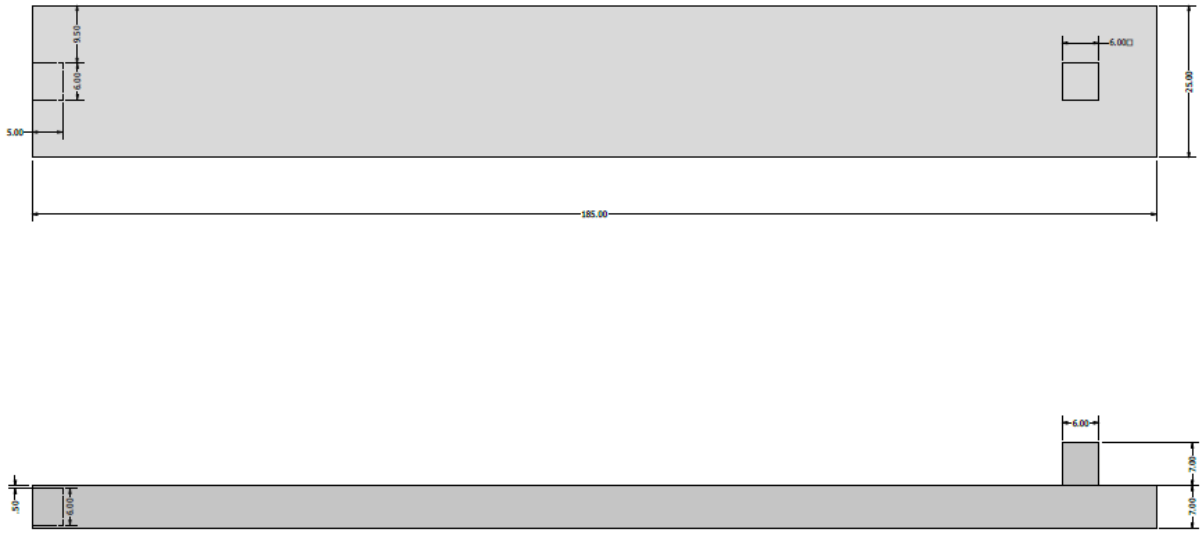


Ilustración 100 Plano columnas cámara

Sorprendentemente este marco cumple una función secundaria no solo sujeta a la cámara de forma adecuada haciendo el sistema más compacto si no que le otorga al sistema muchísima rigidez estructural evitando que se doble la parte de la plataforma pese a un mecanizado no muy bueno.

9.PRESUPUESTO

En este apartado se va a llevar a cabo un presupuesto estimado de la construcción de este prototipo, pero debido al mercado constantemente cambiante y las gestiones de stock también se añadirán otros presupuestos con diferentes configuraciones que se pudieron llevar a cabo donde para una versión comercial hay mucho margen de ahorro para ello se analizara el precio de diferentes componentes entre los elegidos y de diferentes formas de construir la estructura con diferentes materiales.

No se tendrá en cuenta el precio de los cable o resistencias necesarios para mostrar el circuito ya que el precio de estos componentes no supera el céntimo y normalmente vienen en kits con lo cual es difícil de medir al igual que no se tiene en cuenta la cantidad de cinta aislante de la estructura ya que nuevamente el precio es muy reducido y no tiene una función principal en el sistema.

Precio componentes

Este modelo se puede separar en cámara, servomotor, estructura, controlador y microcontrolador.

Cámara

La cámara que se decidió usar es la raspberry pi camera versión 1.3 que tiene un precio de 9.99€ pero también se puede usar la v2 .1 que tiene un precio de 24.83€ o la v3 que ha salido recientemente incorpora un autoenfoque controlado por software y cuesta 29.95€

Aunque también hay cámaras de este tipo para otros dispositivos y de mayor calidad ya que la raspberry pi camera solo permite unos rangos de operación hasta 60º que a nivel industrial no es suficiente o no tiene calidad suficiente para la aplicación necesitando imágenes más grandes o una mayor frecuencia sin perder tanta calidad una para ello se puede mejorar la placa con diferentes sensores o aumentar la disipación añadiendo pequeños disparadores metálicos.

NVIDIA tiene las cámaras JETSON nano diseñadas específicamente para su placa con un precio de 16.88€, otro competidor similar a la NVIDIA es la marca Arducam que ofrecen cámaras compatibles con la raspberry Pi por un precio de 26.95€, también tiene una cámara de 64 megapíxeles con un precio de 71.21€.

Ilustración 101 Arducam

Las cámaras que se venden no varían mucho ya que todas incluyen el mismo sensor fabricado por SONY el IMX219 donde si hay variedad es en la selección del objetivo los cuales pueden proveer a la placa de visión nocturna estos objetivos no son necesarios para la aplicación que se va a desarrollar ya que hemos construido la estructura para que funcione con la cámara con la lente original ya que estos objetivos tienen un coste de entorno a los 60€.

Servomotor

En este apartado se concretará el precio de los servomotores que se pueden usar a mayor coste mayor consumo, pero más controlable será el sistema y podremos realizar una solución más precisa que funcione mejor.
EL primero y más barato es el SG-90 que se puede encontrar por 2.89€ aunque si nos vamos a tiendas como aliexpress lo puedes encontrar por la mitad, el siguiente es el Hs-422 que se puede encontrar por 18.43€ y en aliexpress por un poco más 21.52€ también si la estructura es muy pesada una variación mucho más potente de este servo es el HS-5685MH que tiene un precio de 73.94€ siendo una alternativa el AX-12A con un precio de 59.02€ finalmente otro servomotor que podemos usar con alto par es el AD002 con un precio de 16.99€.
Entre los más típicos para esta operación y que en cuanto a prestaciones precio podemos usar están el MG996R que es una muy buena alternativa si cambiamos los materiales a algo más pesado que se puede encontrar por 8.20€ aunque por aliexpress está por alrededor de 3€ y la solución finalmente escogida es el MG90S que se puede encontrar por 4€ y en aliexpress por 1.6€.
En el presupuesto hay que tener en cuenta que se van a necesitar dos servomotores.

Microcontrolador

Como microcontrolador tenemos varias opciones entre las más populares están la Arduino uno la cual tiene un precio de 29€ aunque tendríamos suficientes líneas implementadas para tener que usar solo uno su precio es muy mejorable, de la misma marca, pero más barato esta la placa Arduino nano la cual puede encontrarse por 8€ y podemos usar 2 para así controlar cada eje con una lo que reduciría el precio final en 10€ además el stock de estas placas es bastante amplio.

Ilustración 102 Arduino nano

La solución que se va a optar es una ESP32 placa que últimamente ha desarrollado gran interés en la industria con la llegada de la industria 4.0 y el IoT lo que permite mucho desarrollo a futuro se puede comprar el chip por menos de un euro y luego diseñar toda la electrónica de manera que hagamos placas personalizadas capaces de adaptarse a la estructura final o podemos comprar el kit final por 7€ la cual al tener varios canales y varios núcleos es capaz de realizar el control de ambos motores pero lo más sencillo es usar una placa por motor además para la cámara y el controlador podemos usar otra placa que por 8 euros incluye la cámara y se puede comunicar por i2c también simplemente usar esta placa sería más barato que la cámara más básica y nos quitaríamos el bloque más caro del sistema pero la idea de este proyecto es poder ver los datos que extraemos.

Controlador

Como controlador se ha elegido una raspberry pi 4 pero en realidad actualmente su precio esta desorbitado hasta un punto donde la opción más económica es comprar la versión más barata y soldar encima las memorias para conseguir los 8gb hoy en día la versión de 4gb cuesta 185€ y la de 8gb cuesta 220€ lo que es un precio muy poco realista así que para el presupuesto se asumirá que el precio es de salida de 59.9€ para la versión de 4gb y la de 8gb por 83€.
Otra opción es usar la placa banana pi cuyo precio está en 110€ pero añade una memoria flash con lo que no habría que introducir ninguna tarjeta de memoria extra para guardar datos y el sistema operativo por un precio similar esta la raspberry pi 4000 que no es más que una raspberry pi 4 en un teclado, aunque no incluye el puerto CSI para usar la cámara con lo cual no es válido para la cámara.
Otra placa que se puede usar es la BeagleBoard el problema de esta placa es que solo tiene 510 kB de RAM lo que puede ser poco si queremos tratar imágenes a mucha velocidad el precio de esta placa es de 62.10€.
La solución que más veces se ha tomado como alternativa a la raspberry pi es la NVIDIA JETSON nano que tiene un precio de 141.93€ pero mucha más potencia que la raspberry pi por un precio actual menor, aunque algo mayor a la banana pi.
Finalmente, la Asus Tinker Edge R tiene un precio de 240 € pero tenemos un núcleo de 1.8 GHz siendo la placa más potente que podemos poner, aunque también la más cara.

Placa	Núcleos	Velocidad	Memoria	Precio
raspberry pi 3	4	1.4 GHz	1 Gb	109.90 €
raspberry pi 4	4	2 GHz	4 Gb	122.95 €
Banana pi M5	4	1.8 GHz	4 Gb	168,60 €
NVIDIA JETSON	4	1.5 GHz	4 Gb	234,90€
Asus Tinker	4	1,8 GHz	6 Gb	281,66€
BeagleBoard	2	1 GHz	4 Gb	48,57€

Tabla 2 Comparativa de controladores

Estructura

En este apartado se van a ver alternativas de material para construir la estructura que fue construida en PLA también puede ser de otros materiales con una producción más rápida como madera o metálica.

Estructura PLA

El PLA es poliácido láctico se trata de un polímero construido de forma similar al ácido láctico su principal ventaja es su biodegradabilidad a temperaturas de 60º en agua y oxido de carbono se fabrica a partir de vegetales como el almidón de maíz o la caña de azúcar lo que hace un material barato, al ser termoplástico es perfecto para rellenar moldes se utiliza en el campo de la impresión 3D por modelado por deposición profunda ya que se funde a solo 200º y se solidifica muy rápido lo que permite crear capas que formen objetos complejos.
Para calcular su coste se va a tomar un precio de 25€ el Kg, aunque varía mucho dependiendo de la calidad. Gracias a el programa Cura cuando creamos el binario el mismo programa nos dice cuántos gramos de material vamos a usar multiplicando la cantidad de material por el número de piezas sacaremos el consumo total de PLA que dividiendo entre el precio del kilo sacaremos el precio completo de la estructura, esto dependerá de la eficiencia de la impresora 3D.
Para la esquina el programa nos da un consumo de 11 gramos que tendremos que multiplicar por 10 siendo el número de esquinas de la estructura dando un consumo de 110 g.
El marco interior tiene un peso de 12 gramos por banda lo que da un peso total de 48 g.
Para el marco externo el peso es mayor teniendo una masa media de 15 gramos lo que le otorga un peso total de 60g.
Finalmente, el marco para la cámara tiene una parte de 12 gramos y otras dos de 19 gramos lo que da un peso de 50 gramos.
La torre del soporte pesa 18 gramos lo que nos añade un peso de impresión de 36 gramos.
En total tenemos un peso de PLA de 304 gramos que traducido a euros son 7.3€ de material a lo que hay que sumar el precio del rodamiento siendo este de 6.5€ lo que la estructura costo 13.8€

Estructura madera

Para construir la estructura de madera no compraremos el material por kilos si no por listones ya que suele ser más económico que mandar a fabricar las piezas necesitaremos una madera que sea ligera por ello se usara madera de balsa de la cual necesitaremos listones de 1cm de ancho y 2.5 de altura su densidad es doce veces menor que la del PLA con lo que necesitaremos aproximadamente 50 gramos de madera de balsa podemos comprar 10 tableros de 20 centímetros de largo por 4 de ancho y 1 cm de alto por 14€ el problema de esta madera es que es muy fina y no es tan resistente como puede ser el PLA además al estar relleno la estructura pesara unos 130 gr al precio hay que sumarle el de los rodamientos de 7.3€ saliendo por un total de 21.3€ aunque la estructura es más ligera y podemos construirla mucho más rápido el precio del kilo de madera de balsa es mucho mayor que el precio del kilo de PLA con lo que no compensa este sistema.

Estructura metal

Para hacer la estructura de metal este tiene que ser un metal ligero y moldeable al ser posible que no se oxide la mejor solución sería usar perfiles de aluminio en total necesitaríamos una longitud de 250 cm de barras lo que haría un precio de 8.65€ el metro siendo un total de 21.65 peor como se venden de 2 metros en 2 metros el precio a pagar seria de 35.8€ más los 7.3€ de los rodamientos la estructura se queda en 43.1€ sin contar que hay que hacer un trabajo de soldadura adicional además no es una solución sostenible ya que no es biodegradable pero por otra parte tenemos una solución mucho más profesional y duradera.

Diferentes presupuestos

Al ser un prototipo y no una versión comercial se plantean diferentes presupuestos que se podrían usar de base para un producto comercial dedicado a la enseñanza de sistemas de control o incluso el diseño de un kit DIY con los que enseñar a hacer un sistema compleja.

Presupuesto normal

El proyecto ha costado en total 107.1€ pasándonos 7.1€ de uno de los retos de este proyecto como es crear un sistema de bajo coste pese a ello las piezas son reutilizables y se pueden utilizar en otros prototipos que requieran de una lógica compleja ya que no hay nada puesto de forma fija.

Componente	Cantidad	Precio Unidad	Total
Raspberry pi camera	1	9.99	9.99
ESP32-Wovere	2	7.66	15.32
Raspberry pi 4	1	59.9	59.9
Motores	2	4	8
Estructura	1	13.8	13.8
Total	7	15.3	107.1

Tabla 3 Presupuesto del prototipo

Presupuesto normal sin esp32

Para llegar al presupuesto propuesto de menos de 100€ podemos hacer una aplicación algo más lenta, pero con un resultado bueno para ello quitaremos la placa ESP32 y podríamos usar únicamente la raspberry pi lo que dejaría un coste de 91.7€ pero el precio medio por unidad sube lo que significa que no es el componente que más está afectando al precio.

Componente	Cantidad	Precio Unidad	Total
Raspberry pi camera	1	9.99	9.99
Raspberry pi 4	1	59.9	59.9
Motores	2	4	8
Estructura	1	13.8	13.8
Total	5	18.34	91.7

Tabla 4 Presupuesto nº1

Presupuesto normal cambiando los motores

Si el sistema no tiene suficiente potencia deberemos cambiar los motores a los de alto par MG996R con un precio unitario de 8.2€ incrementando el precio a 115.41€ subiendo el precio por parte del proyecto a más de 15€, aunque con una solución mucho más rápida.

Componente	Cantidad	Precio Unidad	Total
Raspberry pi camera	1	9.99	9.99
ESP32-Wovere	2	7.66	15.32
Raspberry pi 4	1	59.9	59.9
Motores	2	8.2	16.4
Estructura	1	13.8	13.8
Total	7	16.48	115.41

Tabla 5 Presupuesto nº2

Presupuesto normal sin raspberry pi

La solución más barata de todas es cambiar la Raspberry pi 4 por un kit con cámara que se puede encontrar por 8€ el precio completo del sistema no supera los 50€ y el precio por componente es menor a los 10€ la parte mala es que no tenemos un controlador para ver los resultados en tiempo real por pantalla.

Componente	Cantidad	Precio Unidad	Total
Raspberry pi camera	1	8	8
ESP32-Wovere	2	7.66	15.32
Motores	2	4	8
Estructura	1	13.8	13.8
Total	6	7.52	45.12

Tabla 6 Presupuesto nº3

Presupuesto normal Banana Pi

Cambiando la raspberry pi debido a la falta de stock por una Banana pi el precio completo subirá, pero para el diseño normal estamos considerando un precio desactualizado por el que se adquirió la placa lo que hará este presupuesto mucho más realista, pero lo subirá a 172.35€ debido a lo inflado que esta el mercado de los ordenadores monoplaca.

Componente	Cantidad	Precio Unidad	Total
Raspberry pi camera	1	9.99	9.99
ESP32-Wovere	2	7.66	15.32
Banana pi	1	125.24	125.24
Motores	2	4	8
Estructura	1	13.8	13.8
Total	7	24.62	172.35

Tabla 7 Presupuesto nº4

Presupuesto normal cambiando estructura madera

Usando una estructura de madera que permite una aplicación más rápida aumentaremos un poco más el precio, pero tendremos un sistema un poco más ligero por poco más dinero quedando el precio en 114.21€.

Componente	Cantidad	Precio Unidad	Total
Raspberry pi camera	1	9.99	9.99
ESP32-Wovere	2	7.66	15.32
Raspberry pi 4	1	59.9	59.9
Motores	2	4	8
Estructura	1	13.8	21.3
Total	7	16.31	114.21

Tabla 8 Presupuesto nº5

Presupuesto normal cambiando estructura metal

Si ponemos una estructura metálica el precio se dispara por encima de los 150€ para el prototipo y se complica el montaje ya que es muy posible que tengamos que usar el motor de alto par lo que aumente el precio aún más.

Componente	Cantidad	Precio Unidad	Total
Raspberry pi camera	1	9.99	9.99
ESP32-Wovere	2	7.66	15.32
Raspberry pi 4	1	59.9	59.9
Motores	2	4	8

Estructura	1	41	41
Total	7	15.3	107.1

Tabla 9 Presupuesto nº6

Presupuesto normal banana Pi sin ESP-32

Para reducir un poco el precio del sistema con la banana Pi y que se ajuste a unos 150€ podemos quitar los microcontroladores y que toda la operación de cálculo lo haga la placa principal ahorrándonos unos 15€ que deja un precio final de 157.03 pero con un precio por componente de 31.4€ el más alto de todos.

Componente	Cantidad	Precio Unidad	Total
Raspberry pi camera	1	9.99	9.99
Banana pi	1	125.24	125.24
Motores	2	4	8
Estructura	1	13.8	13.8
Total	5	31.4	157.03

Tabla 10 Presupuesto nº7

Presupuesto posible plataforma comercial

Si lo que se quiere es sacar de este proyecto una solución comercial lo suyo seria crear la opción más resistente que permita un buen uso al paso del tiempo para ello se utilizaran las opciones más potentes y actualizadas cambiando la estructura a la solución metálica y utilizando motores de características industriales además mejorando la placa a la Asus que tiene una mayor potencia y mejorando la cámara a la versión 3.1 pero manteniendo una placa similar para el PID, el precio total se queda en 444.35 pero con una aplicación de mucha mayor calidad que se podría vender como placa de pruebas de diferentes sistemas de control.

Componente	Cantidad	Precio Unidad	Total
Raspberry pi camera	1	29.99	29.99
ESP32-Wovere	2	7.66	15.32
Asus Tinker Edge R	1	240	240
Motores	2	59.02	118.04
Estructura	1	41	41
Total	7	63.48	444.35

Tabla 11 Presupuesto nº8

Los dos presupuestos que veo más interesantes son el primero por ser el que se aplicó en el prototipo y el ultimo por ser un presupuesto realista de un sistema que se podría comercializar para diferentes laboratorios de control, aunque el presupuesto con la estructura de madera es perfecto para sacar un kit DOY y aprender a montar y configurar un lazo de control.

10.CONCLUSIÓN

Personalmente espero que este documento sirva como guía para crear un sistema de control con todos los componentes necesarios para hacer un control completo de un sistema complejo donde tendremos que controlar varias entradas de forma paralela y partes del sistema puede que no serán lineales habiendo que realizar linealizaciones entorno a un punto de equilibrio.

Además se explican varias formas de construcción dependiendo si el proyecto es un prototipo un proyecto comercial o no se disponen de técnicas complejas de fabricación como son la impresión 3D o la soldadura en arco pudiendo ser construido con otros materiales que no añadan un peso excesivo al sistema para no tener que comprar servomotores muy potentes, una de las ventajas de tener un sistema pesado es que el peso de la bola es bastante menor al de la estructura lo que hace que la bola no genere variaciones en el centro de gravedad pudiendo tener un control más laxo.
En este proyecto he aprendido a comunicar en la realidad los distintos bloques que componen un sistema de control con mecanismos que, aunque me pudieran parecer nuevos a simple vista durante el transcurso del grado he visto herramientas similares que me han hecho entender con facilidad cómo funcionan todos los subsistemas y eso me ha permitido hacer una solución mejor y más rápida.

Al principio del proyecto remarcaba unos puntos que espera cumplir en el proyecto de los cuales se han cumplido todos de forma completa aunque sí que es verdad que pueden ser ampliables con información adicional como catálogo de dispositivos o con un uso menor de las librerías teniendo soluciones más configurables la ventaja de utilizar tantas librerías es un ahorro en tiempo a la hora del diseño sí que es cierto que durante la fase de redacción de códigos se decidió escribir el PID a mano este PID es mucho peor que el contenido en la librería y daría un peor resultado ya que tendría que alojar varias variables en la memoria lo que haría el sistema más lento.

Debido a no ser el motivo principal del trabajo y que añadiría a la redacción del proyecto un tiempo del cual no dispongo no se ha incluido en el proyecto la demostración del sistema con su correcto funcionamiento lo que si se ha demostrado es la correcta compilación de los códigos y se garantiza que el sistema es capaz de realizar cualquier tipo de trayectoria de forma continua gracias a un alto grado de modelado del sistema y un buen diseño esquemático del mismo conteniendo todos los bloques para lograr la solución.

Tampoco se ha puesto un gran interés en los materiales de la bola y el plato el cual sí que llegaría a tener una gran repercusión en el rendimiento del sistema cambiando el coeficiente de rozamiento y haciendo que la bola o bien necesite una mayor inclinación para empezar a moverse o se reduzca demasiado y la bola empiece a deslizar por ello se decidió optar por soluciones de materiales muy típicos como es el PVC y la bola de pimpón de plástico que rueda bien en PVC y no desliza además es muy fácil de conseguir aunque la idea del sistemas es poder probar los mecanismos de control con diferentes bolas y superficies para así poder ver como un PID pasa de un sistema a otro que no está correctamente calibrado para el mismo.

Finalmente este proyecto se ha centrado en la programación de los dispositivos más usados en la electrónica sobre todo a nivel de aficionados aunque ahora con el auge de la industria 4.0 cada vez está más presente en la industria y sistemas como estos serán muy reclamados en el futuro para realizar pequeñas aplicaciones donde no se necesiten grandes sensores ni armarios enormes con PLCs que gracias a unos módulos sean capaces de llevar la operación completa desde la generación de la entrada hasta el montaje de la estructura que tenga un movimiento con unas pérdidas mecánicas mínimas.

Este proyecto tiene muchos puntos de mejora el primero es diseñar una versión comercial ya que me parece un campo muy útil y muy poco explotado con únicamente dos empresas con soportes oficiales de los cuales ninguna maneja una solución de código abierto donde se pueda aprender a usar programas más reclamados por la industria por su mayor control y menor coste como son Python en combinación de C.

Aparte de eso este proyecto tiene una pequeña parte de mecanizado que por no tener relevancia en el sistema final y poder ser solventado con un mejor diseño no se ha recogido en el presente documento al igual que la parte de configuración del sistema operativo de la raspberry pi.

Una mejora posible es hacer el sistema a distancia para ello las raspberry pi incluye el protocolo SSH con el que podríamos controlar los parámetros del sistema desde cualquier parte del mundo con una conexión a internet además gracias a las conexiones inalámbricas de la ESP32 se podría llevar un control de las variables del sistema mediante comunicación a la nube para el muestreo de cómo cambian estas varias dependiendo de la situación del sistema siendo una gran ventaja frente a la competencia que pueda surgir y ajustándonos a un sistema de industria 4.0.

Otro punto de mejora del presente proyecto que no se llevó a cabo es la redacción de un apartado de pliego de condiciones donde se recojan todos los sistemas que se pueden implementar para que el sistema cumpla con los estándares industriales al ser un prototipo para la enseñanza y cada sistema hijo de este llevaría unas condiciones diferentes no se vio interesante rellenar un pliego de condiciones.

Otra mejora posible al sistema que se estuvo planteando es implementar un control por inteligencia artificial que recogiendo las señales de un joystick dirigido por un humano pueda reproducirlas a la perfección haciendo un sistema de control en el cual no necesitemos saber el modelo del sistema simplemente un buen entrenamiento que en el momento que sea suficientemente bueno se le podría entrenar por su cuenta para llegar a una solución mejor que la del PID por ello se intentó usar placas que no bajasen de los 4Gbs de RAM y que estén orientadas a redes neuronales.

Finalmente, el sistema podría haber incorporado unos potenciómetros como en el laboratorio de automática de esta universidad que permitan cambiar los valores del PID y no haya que reprogramar todo el sistema simplemente para cambiar la ganancia una versión comercial debería si o si incorporar este sistema.

Como no otro punto que en una versión más avanza sería de gran utilidad y que se ha mencionado con anterioridad es el diseño CAD no solo de la estructura si no de una placa con toda la electrónica necesaria que haga de este sistema un sistema compacto y permitan un mayor orden.

Se puede ver que este proyecto además de ser de gran utilidad para aprender electrónica tiene mucho ámbito de mejora lo que anima a aprender más y crear sistemas más complejos que puedan competir o incluso superar a versiones que llevan muchos años en el mercado y no han necesitado actualizarse lo que genera una oportunidad que seguro que alguien es capaz de aprovechar y utilizar el presente documento como pequeña guía para cumplir su objetivo.

11.BILBLIOGRAFIA

https://www.quanser.com/products/ball-and-beam/
https://www.quanser.com/products/2-dof-ball-balancer/
https://www.feedbackinstruments.com/pdf/brochures/33052_Datasheet_Ball_and_Plate_LABVIEW_10_2013.pdf
https://labview.hackster.io/balancers/ball-beam-balance-d7ef28
https://link.springer.com/chapter/10.1007/978-3-642-19542-6_118
https://idus.us.es/bitstream/handle/11441/102088/TFG2876QUESADA%20MARA%C3%91ON.pdf?sequence=1&isAllowed=y
https://github.com/Michael-Graves/Ball-and-Plate
http://origamata.com/projects/ballbalancemachine/
https://www.gunt.de/images/download/RT12x_spanish.pdf
http://kyb.fei.tuke.sk/laben/modely/gnk.php
https://shane.engineer/blog/iphone-game-playing-robot
https://www.a-m-c.com/es/experiencia/technologies/peripheral-interface/rs-232/#:~:text=El%20"estándar%20recomendado%20232"%2C,ratones%20y%20otros%20dispositivos%20periféricos.
http://www.sc.ehu.es/sbweb/fisica_/solido/mov_general/plano_inclinado/plano_inclinado.html#Balance%20energético/
https://ctms.engin.umich.edu/CTMS/index.php?example=BallBeam&section=SystemModeling