



GRADO EN INGENIERÍA DE ROBÓTICA SOFTWARE

Escuela de Ingeniería de Fuenlabrada

Curso académico 2022-2023

Trabajo Fin de Grado

Diseño y fabricación de un brazo robótico industrial impreso en 3D, de bajo costo y código abierto con soporte para ROS 2.

Tutor: Julio Vega Pérez

Autor: Vidal Pérez Bohoyo



Este trabajo se distribuye bajo los términos de la licencia internacional CC BY-NC-SA International License (Creative Commons AttributionNonCommercial-ShareAlike 4.0). Usted es libre de *(a) compartir*: copiar y redistribuir el material en cualquier medio o formato; y *(b) adaptar*: remezclar, transformar y crear a partir del material. El licenciador no puede revocar estas libertades mientras cumpla con los términos de la licencia:

- *Atribución.* Usted debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciatante.
- *No comercial.* Usted no puede hacer uso del material con propósitos comerciales.
- *Compartir igual.* Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original.

Agradecimientos

Unas bonitas palabras...

Quizás un segundo párrafo esté bien. No te olvides de nadie.

Un tercero tampoco viene mal para contar alguna anécdota...

¿Alguien más? Aunque sean *actores* secundarios.

Un quinto párrafo como colofón.

*A alguien especial;
si no, tampoco pasa nada*

Madrid, xx de xxxxxx de 20xx

Tu nombre

Resumen

Escribe aquí el resumen del trabajo. Un primer párrafo para dar contexto sobre la temática que rodea al trabajo.

Un segundo párrafo concretando el contexto del problema abordado.

En el tercer párrafo, comenta cómo has resuelto la problemática descrita en el anterior párrafo.

Por último, en este cuarto párrafo, describe cómo han ido los experimentos.

Acrónimos

ADN *Ácido Desoxirribonucleico*

STEM *Science, Technology, Engineering and Mathematics*

DIY *Do It by Yourself*

Índice general

1. Introducción	1
1.1. Robótica industrial	1
1.2. Robótica en educación	3
1.2.1. Robótica en colegios e institutos	3
1.2.2. Robótica en universidades	4
1.3. Robótica de bajo coste	4
2. Estado del arte	7
3. Objetivos	8
3.1. Descripción del problema	8
3.2. Requisitos	9
3.3. Metodología	9
3.4. Plan de trabajo	10
4. Plataforma de desarrollo	11
4.1. Software	11
4.1.1. Python	11
4.1.2. Grbl	12
4.2. Hardware	12
4.2.1. Código G	12
4.3. MKS DLC32	12
4.4. Motores paso a paso	13
4.5. Controladores de motores	13
5. Diseño	15
5.1. Snippets	15
5.2. Verbatim	15
5.3. Ecuaciones	16
5.4. Tablas o cuadros	16

6. Conclusiones	18
6.1. Conclusiones	18
6.2. Corrector ortográfico	19
Bibliografía	20

Índice de figuras

1.1.	Robots en líneas de ensamblaje	2
1.2.	Robot Dobot M1 realizando una soldadura con estaño	2
1.3.	Robots en el ámbito científico	3
1.4.	Robots educativos Lego Mindstorms en escuelas.	4
1.5.	Robot IceBot que utiliza una FPGA libre Alhambra II.	5

Listado de códigos

5.1. Función para buscar elementos 3D en la imagen	15
5.2. Cómo usar un Slider	16

Listado de ecuaciones

5.1. Ejemplo de ecuación con fracciones	16
5.2. Ejemplo de ecuación con array y letras y símbolos especiales	16

Índice de cuadros

5.1. Parámetros intrínsecos de la cámara	17
--	----

Capítulo 1

Introducción

El éxito es la capacidad de ir de fracaso en fracaso sin perder el entusiasmo.

Winston Churchill

La tendencia de la industria hacia la automatización total ha ido en aumento en los últimos años, haciendo que la demanda de robots industriales se dispare en todo el mundo. Un ejemplo de ello, son las megafactorías en el sector automovilístico. Se tratan de inmensas fábricas con un componente humano mínimo y con una automatización cada día mayor, mediante el uso de la robótica industrial.

No solo se hace uso de grandes robots pesados, sino también de flotas de pequeños pero ágiles robots que desempeñan tareas en cintas transportadoras, ensamblado de placas base, entre otras. Debido al aumento en su uso y del progreso en estas tecnologías, es necesario formar cada vez más ingenieros que ejercerán en este campo. Es por esto que se necesitan herramientas adaptadas a las nuevas tecnologías y fácilmente accesibles para estudiantes y centros.

Por otro lado, la robótica educativa ha nacido como una herramienta innovadora y poderosa en el ámbito de la enseñanza. Mediante la integración de robots en las aulas, los estudiantes pueden experimentar el aprendizaje de una manera más interactiva y participativa. Desde los centros de educación primaria hasta las universidades, la incorporación de la robótica educativa ha experimentado un crecimiento significativo. Esta tecnología se adapta de manera excepcional a las necesidades y capacidades de cada etapa educativa.

1.1. Robótica industrial

La robótica industrial es la rama de la ingeniería dedicada al diseño, construcción, operación y mantenimiento de robots utilizados en la automatización de procesos

industriales. Estos robots pueden realizar una amplia variedad de tareas, desde la manipulación y transporte de materiales, hasta el ensamblaje y soldadura de piezas. Entre las aplicaciones más importantes encontramos las siguientes:

- *Automatización de líneas de ensamblaje.* Esta tecnología se encuentra ampliamente asentada en las líneas de producción del sector automotriz. Un ejemplo son las factorías de Tesla.



Figura 1.1: Robots en líneas de ensamblaje

- *Soldadura.* Es utilizada para realizar la unión de estructuras mecánicas debido a su alta precisión y capacidad de realizar la misma soldadura perfecta una y otra vez. Además, también son usados para soldar placas de circuitos.

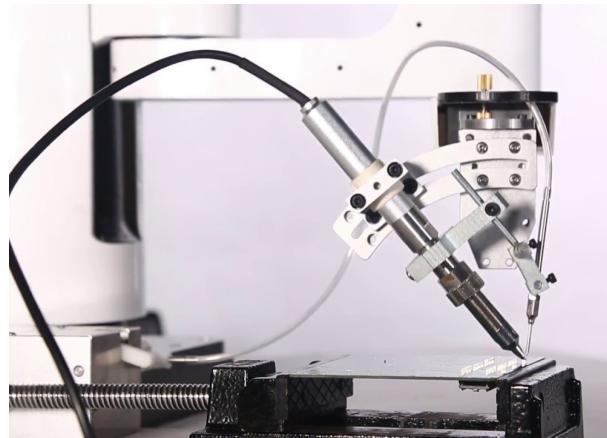


Figura 1.2: Robot Dobot M1 realizando una soldadura con estaño

- *Investigación y desarrollo en laboratorios.* Los brazos robóticos realizan tareas de laboratorio repetitivas y precisas, lo que puede acelerar el proceso de investigación y desarrollo de nuevos productos médicos. Uno de los usos más frecuentes es preparar y procesar muestras en investigaciones científicas. En concreto, pueden desempeñar tareas como la extracción de *Ácido Desoxirribonucleico* (ADN), la separación de componentes, la adición de reactivos y el análisis de muestras químicas, entre otros. Esto ayuda a reducir los errores humanos y garantizar la integridad y reproducibilidad de los experimentos llevados a cabo.



Figura 1.3: Robots en el ámbito científico

1.2. Robótica en educación

La robótica en la educación es una disciplina que ha adquirido una gran relevancia en los últimos años debido a la creciente necesidad de formar a las nuevas generaciones en competencias tecnológicas.

1.2.1. Robótica en colegios e institutos

En las escuelas, se ha convertido en una herramienta pedagógica eficaz para desarrollar habilidades y conocimientos en áreas como la programación, la matemática, la electrónica y la resolución de problemas. Esto es conocido como *Science, Technology, Engineering and Mathematics* (STEM). Los estudiantes aprenden a diseñar, construir y programar robots simples para llevar a cabo una tarea específica, lo que les ayuda a comprender los conceptos de ciencia y tecnología de una manera más práctica e interactiva.



Figura 1.4: Robots educativos Lego Mindstorms en escuelas.

1.2.2. Robótica en universidades

En el nivel universitario, la robótica se ha convertido en una disciplina esencial para formar a los futuros ingenieros. Los estudiantes aprenden a diseñar y construir robots más avanzados, y refuerzan sus habilidades de programación y control de sistemas complejos. Además se hace uso de más tipos de robots, como pueden ser, industriales o plataformas robóticas móviles reales. Al ser sistemas usados en el mundo profesional, los estudiantes pueden aprender con el robot que usarán en un futuro. Aunque es verdad que las universidades disponen de algunas unidades, no siempre son accesibles para el estudiante por diversas razones.

En resumen, la robótica en la educación es una herramienta poderosa para fomentar el aprendizaje y la innovación en las nuevas generaciones. Desde la escuela hasta la universidad, la robótica se ha convertido en una disciplina clave para formar a los futuros líderes tecnológicos del mundo.

1.3. Robótica de bajo coste

La robótica de bajo coste es un área de la robótica enfocada en el diseño y desarrollo de robots accesibles y asequibles. El objetivo principal de esta, es conseguir desarrollar tecnología robótica barata, reduciendo la complejidad de los sistemas empleados y haciendo uso de materiales más económicos.

La disponibilidad de herramientas de fabricación de bajo costo, como la impresión 3D y el corte láser, han hecho posible que los usuarios puedan crear piezas y componentes robóticos personalizados a un precio más bajo que el de la fabricación tradicional.

La Ley de Moore es una ley teórica que establece que la capacidad de procesamiento de los microchips se duplica cada dos años. Esta observación, pese haberse dado hace casi 60 años, sigue siendo válida a día de hoy. Entre otras cosas, es lo que ha permitido el desarrollo de componentes electrónicos cada vez más pequeños, eficientes y potentes. La robótica de bajo coste es un claro ejemplo de cómo el crecimiento de la capacidad de computación y la disminución de los costes de producción ha llevado al abaratamiento de la robótica y de la electrónica en general.

Gracias a esto, la robótica de bajo coste, se basa en la filosofía ‘Hazlo tú mismo’ o *Do It by Yourself* (DIY), que se enfoca en la creación de proyectos personalizados y asequibles utilizando tecnología de bajo costo y materiales comunes.

En resumen, la robótica de bajo costo es una consecuencia directa del crecimiento de la capacidad de computación, el abaratamiento de los precios de los componentes electrónicos y del enfoque *DIY*. Esto ha permitido la creación de robots avanzados y accesibles para el público general. El desarrollo de esta tecnología, permite que más personas puedan experimentar en este área de la ingeniería y desarrollar sus propias soluciones robóticas.

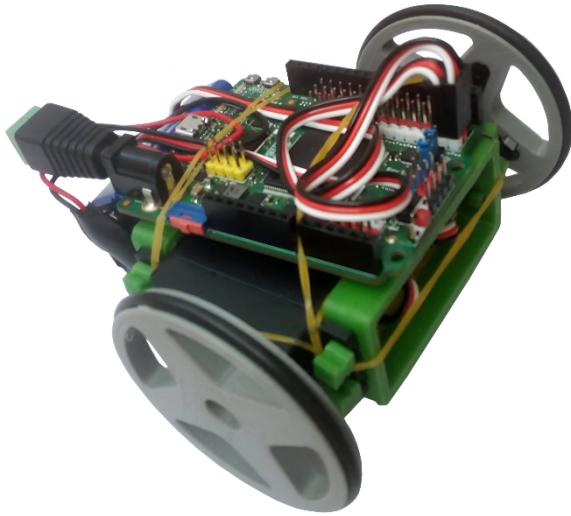


Figura 1.5: Robot IceBot que utiliza una FPGA libre Alhambra II.

Referencias bibliográficas Cita, sobre todo en este capítulo, referencias bibliográficas que respalden tu argumento. Para citarlas basta con poner la instrucción `\cite` con el identificador de la cita. Por ejemplo: libros como [Vega et al., 2012], artículos como [Vega and Cañas, 2019], URLs como [Vega, 2019], tesis como [Vega, 2018a], congresos como [Vega, 2018b], u otros trabajos fin de grado como [Vega, 2008].

Las referencias, con todo su contenido, están recogidas en el fichero `bibliografia.bib`. El contenido de estas referencias está en formato BibTeX. Este formato se puede obtener en muchas ocasiones directamente, desde plataformas como [Google Scholar](#) u otros repositorios de recursos científicos.

Existen numerosos estilos para reflejar una referencia bibliográfica. El estilo establecido por defecto en este documento es APA, que es uno de los estilos más comunes, pero lo puedes modificar en el archivo `memoria.tex`; concretamente, cambiando el campo `apalike` a otro en la instrucción `\bibliographystyle{apalike}`.

Y, para terminar este capítulo, resume brevemente qué vas a contar en los siguientes.

Capítulo 2

Estado del arte

En esta sección, se expondrá el estado del arte de los robots industriales en educación. Esta fase fundamental de la investigación se ha completado gracias a la búsqueda en diversas fuentes de renombre, con el fin de recopilar información que pueda ser de utilidad para desarrollar el presente proyecto de fin de grado. Como resultado, se han seleccionado una serie de trabajos relevantes y significativos en la materia, que se procederán analizar a continuación.

- *Entorno de simulación.* Hemos usado dos entornos de simulación: uno en 3D y otro en 2D.
- *Entornos reales.* Dentro del campus, hemos realizado experimentos en Biblioteca y en el edificio de Gestión.

Capítulo 3

Objetivos

Un objetivo sin un plan es solo un deseo

Antoine de Saint-Exupéry

Tras haber enmarcado el contexto en el cual se encuentra este trabajo de fin de grado, se procede a realizar una descripción del problema, requisitos, metodología y plan de trabajo usado.

3.1. Descripción del problema

El objetivo principal de este trabajo de fin de grado es desarrollar un brazo robótico desde cero. Se pretende que el robot pueda ser empleado en la asignatura de Robótica Industrial en la Universidad Rey Juan Carlos para uso. Es por esto, que se busca que el sistema creado sea asequible y fácilmente replicable mediante el uso de la impresión 3D.

Con el fin de alcanzar esta meta, se ha dividido el proyecto en los siguientes subobjetivos:

1. Realizar una investigación acerca de los robots que actualmente están disponibles y que cumplan con las características y objetivos deseados. Estos robots deberán tener un tamaño y costo similar, y preferiblemente haber sido impresos en 3D. Se dará mayor relevancia a aquellos que utilicen un software y hardware libres.
2. Explorar diversas opciones de diseño para determinar la forma ideal del robot. Se analizará cuidadosamente qué tipo de robot se ajusta mejor al uso que se le pretende dar, así como los grados de libertad necesarios.
3. Realizar una investigación exhaustiva sobre los componentes de hardware disponibles en el mercado, con el fin de seleccionar aquellos que mejor se adapten

a las necesidades y objetivos de construir un robot eficiente y funcional. Se llevará a cabo un análisis detallado de los precios y características de cada componente, para seleccionar aquellos que ofrezcan la mejor relación calidad-precio. Una vez evaluados todos los aspectos, se elegirá un conjunto final de componentes para construir el robot de manera efectiva.

4. Diseño CAD (diseño asistido por ordenador) del brazo. Se pretende hacer uso de FreeCad, entre otras herramientas, para diseñar en 3D cada pieza que constituirá el robot. Puesto a que debe ser imprimible en una impresora FDM convencional, debe estar pensado para no necesitar soportes y necesitar de poco material.
5. Emplear una impresora 3D convencional para materializar los diseños realizados anteriormente.
6. Programar el software necesario para poder controlar el robot desde el ordenador.
7. ROS de

3.2. Requisitos

De este trabajo se espera un brazo robot de tipo industrial de bajo coste cuya fabricación completa esté por debajo de 200€. Otro requisito es que la mayoría de las partes que componen el robot, sean imprimibles en cualquier impresora 3D de gama baja.

Además, se busca un consumo reducido para poder ser usado en robots móviles sin necesidad de estar conectado a una gran fuente de energía.

En cuanto a sus dimensiones, se busca un tamaño idóneo para su uso en escritorios y fácilmente portable. Esto implica que no necesariamente tiene que estar fijo al suelo, permitiendo su fácil traslado.

Se busca continuidad en el proyecto a largo plazo, por lo que una integración en ecosistema ROS 2 es una pieza clave.

3.3. Metodología

Durante el desarrollo del trabajo se ha establecido un protocolo de reuniones semanales con el tutor a través de la plataforma Teams, con el objetivo de compartir los

avances realizados y recibir retroalimentación sobre el trabajo. Además, cada semana se han propuesto las actividades a realizar, asegurando así una adecuada planificación y coordinación del proyecto.

Para el desarrollo del sistema se ha utilizado un repositorio en la plataforma GitHub¹, en el cual se ha ido subiendo el código y diseños generados a lo largo del proyecto. Adicionalmente, en este mismo repositorio se ha incluido una Wiki² con las explicaciones detalladas de todas las actividades llevadas a cabo durante estos meses de trabajo. De esta manera, se ha creado un registro completo y accesible de todo el proceso de desarrollo del sistema.

3.4. Plan de trabajo

El desarrollo del TFG ha estado dividido en dos etapas. La primera, comenzó en octubre y fue abandonada en enero.

¹<https://github.com/RoboticsURJC/tfg-vperez>

²<https://github.com/RoboticsURJC/tfg-vperez/wiki>

Capítulo 4

Plataforma de desarrollo

*Las herramientas adecuadas en las manos adecuadas
pueden cambiar el mundo*

Steve Jobs

Tras haber establecido los objetivos que se pretenden alcanzar en este trabajo de fin de grado, se procede a describir las herramientas *software* y *hardware* utilizadas para lograrlos.

4.1. Software

Llamamos *software* al conjunto de programas, instrucciones y datos que dotan de lógica al sistema.

4.1.1. Python

Python es un lenguaje de programación, es decir, una serie de reglas gramaticales bien definidas que le proporciona a una persona, en este caso el programador, la capacidad de programar una serie de instrucciones o secuencias de órdenes con el fin de crear un comportamiento lógico determinado.

Se trata de un lenguaje de alto nivel interpretado. Cuando se dice que un lenguaje es de alto nivel, se hace referencia a su nivel de abstracción y facilidad de uso en comparación con los lenguajes de bajo nivel, es decir, aquellos que te permiten mayor control sobre los componentes físicos del sistema. Decimos que es un lenguaje interpretado debido a que no es necesario convertir el texto escrito por el humano en instrucciones entendibles por un procesador previamente a su ejecución. Esto agiliza el proceso de desarrollo ya que la conversión de código humano a código máquina (compilación) suele demorar un tiempo. Por el contrario, la ejecución de un programa en lenguaje compilado es más rápida.

Una de las características más destacadas de Python es su amplia librería (conjunto de código destinado a un objetivo concreto) estándar, que proporciona variedad de módulos y funciones para realizar multitud de tareas. Como cualquier otro lenguaje de programación, cuenta con una gran cantidad de librerías de terceros que amplían sus capacidades, como SciPy en la ciencia de datos, Django en el desarrollo web, Numpy para operar con matrices, TensorFlow el aprendizaje automático, entre otros.

4.1.2. Grbl

Grbl¹ es un firmware de código abierto usado para controlar máquinas llamadas CNC (Control Numérico por Computadora). Este firmware se ejecuta en un microcontrolador, que se encuentra dentro de la controladora de la máquina CNC, en nuestro caso, la placa base del robot.

Básicamente, convierte las instrucciones de código G, que posteriormente hablaremos de él, en señales eléctricas que se envían a los motores de la máquina. Además, comprueba los diferentes sensores de la máquina, como pueden ser los finales de carrera, para establecer los límites físicos de cada movimiento.

Grbl es flexible por lo que podemos cambiar la configuración para adaptarla a un caso de uso concreto. De hecho, aunque solo soporta movimientos lineales, en este trabajo se abordará la configuración necesaria para poder usar las articulaciones rotativas del nuestro robot.

4.2. Hardware

Cuando hablamos de *hardware*, nos referimos a todos los componentes físicos de utilizados en el dispositivo electrónico, en este caso, un robot. Estos componentes son tangibles, es decir, se pueden tocar y ver.

4.2.1. Código G

Se trata

4.3. MKS DLC32

Se trata de una placa destinada al mundo de las máquinas de grabado láser. Ha sido creada por *MakerBase* y es considerada *Open Hardware* por lo que toda la información de la placa puede encontrarse en su repositorio de Github². Es fácilmente adquirible

¹<https://github.com/gnea/grbl>

²<https://github.com/makerbase-mks/MKS-DLC32>

por Aliexpress por un precio que ronda los 16€. Está basada en el microcontrolador de 32 bits: ESP32. Se trata de un dispositivo muy asentado en la comunidad *maker* debido a su bajo coste e integración en el ecosistema Arduino. De hecho, gracias a su conectividad wifi y bluetooth ha ganado terreno a los microcontroladores Atmega que incorporan los propios Arduinos.

Esta placa es ideal para este proyecto debido a que cuenta con la posibilidad de controlar hasta 3 motores y es completamente compatible con Grbl. Además dispone de una salida de potencia regulable controlable mediante Grbl que nos permite alimentar dispositivos. Estos podrían ser: electroimán (tipo de imán que es activado mediante electricidad), motor CC (motor de corriente continua) entre otros. Además se puede aprovechar las salidas PWM (*Pulse Width Modulation*) para conectar un grabador láser o un servo. El rango de funcionamiento es de 12 a 24 voltios por lo que es adecuado para ser alimentado mediante baterías y con cargadores de ordenador.

4.4. Motores paso a paso

Un motor paso a paso es un tipo de motor que se mueve en pequeños pasos o incrementos discretos en lugar de girar continuamente. Estos pasos son controlados por señales eléctricas que hacen girar al motor una cantidad específica de grados cada vez que se envía una señal. Debido a que se tiene control sobre su avance son una excelente opción si se quiere tener un motor que sea capaz de posicionarse en un ángulo concreto con exactitud. A pesar de su gran precisión, los motores paso a paso convencionales no tienen el conocimiento absoluto de su posición, por lo que todos los movimientos son relativos. Esto hace que se requiera de un *homming* (proceso en el cual la máquina CNC lleva las partes móviles a una posición conocida) en el arranque de la máquina para conocer su estado antes de operar.

4.5. Controladores de motores

Un controlador paso a paso es el módulo hardware capaz de trasformar las señales lógicas que le envía el controlador en una serie de pulsos de potencia que excitarán las bobinas del motor en un cierto orden para lograr el movimiento. Existen motores bipolares, unipolares e híbridos. La diferencia entre ellos radica en la disposición de las bobinas de su interior. Los más usados en impresoras 3D y CNC son los bipolares. Son reconocibles debido a que tienen 4 cables. En este tipo de placas base se pueden instalar distintos modelos de controladores. Cada uno de ellos tiene unas prestaciones diferentes y por tanto un precio distinto. Unos ofrecen mayor capacidad de corriente

(para controlar motores más grandes), pulsos más suaves que reducen el ruido sonoro y las vibraciones, medición en tiempo real de la corriente consumida para conocer el final de una articulación, entre otras tecnologías.

Capítulo 5

Diseño

Quizás algún fragmento de libro inspirador...

Autor, Título

Escribe aquí un párrafo explicando brevemente lo que vas a contar en este capítulo. En este capítulo (y quizás alguno más) es donde, por fin, describes detalladamente qué has hecho y qué experimentos has llevado a cabo para validar tus desarrollos.

5.1. Snippets

Puede resultar interesante, para clarificar la descripción, mostrar fragmentos de código (o *snippets*) ilustrativos. En el Código 5.1 vemos un ejemplo escrito en C++.

```
void Memory::hypothesizeParallelograms () {
    for(it1 = this->controller->segmentMemory.begin(); it1++) {
        squareFound = false; it2 = it1; it2++;
        while ((it2 != this->controller->segmentMemory.end()) && (!squareFound))
        {
            if (geometry::haveACommonVertex((*it1), (*it2), &square)) {
                dist1 = geometry::distanceBetweenPoints3D ((*it1).start, (*it1).end);
                dist2 = geometry::distanceBetweenPoints3D ((*it2).start, (*it2).end);
            }
        // [...]
    }
}
```

Código 5.1: Función para buscar elementos 3D en la imagen

En el Código 5.2 vemos un ejemplo escrito en Python.

5.2. Verbatim

Para mencionar identificadores usados en el código —como nombres de funciones o variables— en el texto, usa el entorno literal o verbatim

```

def mostrarValores():
    print (w1.get(), w2.get())

master = Tk()
w1 = Scale(master, from_=0, to=42)
w1.pack()
w2 = Scale(master, from_=0, to=200, orient=HORIZONTAL)
w2.pack()
Button(master, text='Show', command=mostrarValores).pack()

mainloop()

```

Código 5.2: Cómo usar un Slider

`hypothesizeParallelograms()`. También se puede usar este entorno para varias líneas, como se ve a continuación:

```

void Memory::hypothesizeParallelograms () {
    // add your code here
}

```

5.3. Ecuaciones

Si necesitas insertar alguna ecuación, puedes hacerlo. Al igual que las figuras, no te olvides de referenciarlas. A continuación se exponen algunas ecuaciones de ejemplo: Ecuación 5.1 y Ecuación 5.2.

$$H = 1 - \frac{\sum_{i=0}^N \left(\frac{d_{js} + d_{je}}{2} \right)}{M} \quad (5.1)$$

Ecuación 5.1: Ejemplo de ecuación con fracciones

$$v(\text{entrada}) = \begin{cases} 0 & \text{if } \epsilon_t < 0,1 \\ K_p \cdot (T_t - T) & \text{if } 0,1 \leq \epsilon_t < M_t \\ K_p \cdot M_t & \text{if } M_t < \epsilon_t \end{cases} \quad (5.2)$$

Ecuación 5.2: Ejemplo de ecuación con array y letras y símbolos especiales

5.4. Tablas o cuadros

Si necesitas insertar una tabla, hazlo dignamente usando las propias tablas de L^AT_EX, no usando pantallazos e insertándolas como figuras... En el Cuadro 5.1 vemos

un ejemplo.

Parámetros	Valores
Tipo de sensor	Sony IMX219PQ[7] CMOS 8-Mpx
Tamaño del sensor	3.674 x 2.760 mm (1/4" format)
Número de pixels	3280 x 2464 (active pixels)
Tamaño de pixel	1.12 x 1.12 um
Lente	f=3.04 mm, f/2.0
Ángulo de visión	62.2 x 48.8 degrees
Lente SLR equivalente	29 mm

Cuadro 5.1: Parámetros intrínsecos de la cámara

Capítulo 6

Conclusiones

Quizás algún fragmento de libro inspirador...

Autor, Título

Escribe aquí un párrafo explicando brevemente lo que vas a contar en este capítulo, que básicamente será una recapitulación de los problemas que has abordado, las soluciones que has prouesto, así como los experimentos llevados a cabo para validarlos. Y con esto, cierras la memoria.

6.1. Conclusiones

Enumera los objetivos y cómo los has cumplido.

Enumera también los requisitos implícitos en la consecución de esos objetivos, y cómo se han satisfecho.

No olvides dedicar un par de párrafos para hacer un balance global de qué has conseguido, y por qué es un avance respecto a lo que tenías inicialmente. Haz mención expresa de alguna limitación o peculiaridad de tu sistema y por qué es así. Y también, qué has aprendido desarrollando este trabajo.

Por último, añade otro par de párrafos de líneas futuras; esto es, cómo se puede continuar tu trabajo para abarcar una solución más amplia, o qué otras ramas de la investigación podrían seguirse partiendo de este trabajo, o cómo se podría mejorar para conseguir una aplicación real de este desarrollo (si es que no se ha llegado a conseguir).

6.2. Corrector ortográfico

Una vez tengas todo, no olvides pasar el corrector ortográfico de L^AT_EXa todos tus ficheros *.tex*. En Windows, el propio editor TeXworks incluye el corrector. En Linux, usa aspell ejecutando el siguiente comando en tu terminal:

```
aspell --lang=es --mode=tex check capitulo1.tex
```

Bibliografía

- [Vega, 2008] Vega, J. (2008). Navegación y autolocalización de un robot guía de visitantes. Master thesis on computer science, Rey Juan Carlos University.
- [Vega, 2015] Vega, J. (2015). De la tiza al robot. Technical report.
- [Vega, 2018a] Vega, J. (2018a). *Educational framework using robots with vision for constructivist teaching Robotics to pre-university students*. Doctoral thesis on computer science and artificial intelligence, University of Alicante.
- [Vega, 2018b] Vega, J. (2018b). JdeRobot-Kids framework for teaching robotics and vision algorithms. In *II jornada de investigación doctoral*. University of Alicante.
- [Vega, 2019] Vega, J. (2019). El profesor Julio Vega, finalista del concurso 'Ciencia en Acción 2019'. URJC, on-line newspaper interview.
- [Vega and Cañas, 2019] Vega, J. and Cañas, J. (2019). PyBoKids: An innovative python-based educational framework using real and simulated Arduino robots. *Electronics*, 8:899–915.
- [Vega et al., 2012] Vega, J., Perdices, E., and Cañas, J. (2012). *Attentive visual memory for robot localization*, pages 408–438. IGI Global, USA. Text not available. This book is protected by copyright.