



## **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 licenciante.
- *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 la misma licencia del original.

*Documento de Julio Vega.*

# 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

---

**AERO** *Autonomous Exploration Rover*

**AI** *Artificial Intelligence*

**ANN** *Artificial Neural Network*

**API** *Application Programming Interface*

**EKF** *Extended Kalman Filter*

**FOA** *Focus of Attention*

**GA** *Genetic Algorithm*

**GPIO** *General Purpose Input/Output*

**GPS** *Global Positioning System*

**HCI** *Human-Computer Interaction*

**HRI** *Human-Robot Interaction*

# Índice general

---

<b>1. Introducción</b>	<b>1</b>
1.1. Robótica industrial . . . . .	1
1.2. Robótica en educación . . . . .	2
1.2.1. Números . . . . .	3
1.2.2. Listas . . . . .	3
<b>2. Objetivos</b>	<b>5</b>
2.1. Descripción del problema . . . . .	5
2.2. Requisitos . . . . .	5
2.3. Metodología . . . . .	6
2.4. Plan de trabajo . . . . .	6
<b>3. Plataforma de desarrollo</b>	<b>7</b>
3.1. Python . . . . .	7
3.2. Grbl . . . . .	7
3.3. Código G . . . . .	8
3.4. MKS DLC32 . . . . .	8
3.5. Motores paso a paso . . . . .	8
3.6. Controladores de motores . . . . .	9
<b>4. Diseño</b>	<b>10</b>
4.1. Snippets . . . . .	10
4.2. Verbatim . . . . .	10
4.3. Ecuaciones . . . . .	11
4.4. Tablas o cuadros . . . . .	11
<b>5. Conclusiones</b>	<b>13</b>
5.1. Conclusiones . . . . .	13
5.2. Corrector ortográfico . . . . .	14



# Índice de figuras

---

1.1. Robots educativos en la escuela secundaria. . . . .	2
1.2. Robots educativos en universidades. . . . .	3



# Listado de códigos

---

4.1. Función para buscar elementos 3D en la imagen . . . . .	10
4.2. Cómo usar un Slider . . . . .	11

# Listado de ecuaciones

---

4.1. Ejemplo de ecuación con fracciones . . . . .	11
4.2. Ejemplo de ecuación con array y letras y símbolos especiales . . . . .	11

# Índice de cuadros

---

4.1. Parámetros intrínsecos de la cámara . . . . .	12
--	----

---

# 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 del automóvil. 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... 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.

### 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 del automotriz. Un ejemplo es la

- *Soldadura.* Es utilizada en soldadura de estructuras mecánicas debido a su alta precisión y capacidad de realizar la misma soldadura perfecta una y otra vez.
- *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.

## 1.2. Robótica en educación

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

En las escuelas de secundaria, 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 *STEM* (Science, Technology, Engineering and Mathematics). 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.

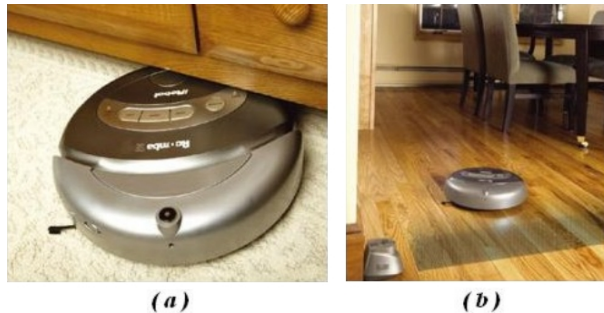


Figura 1.1: Robots educativos en la escuela secundaria.

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 hacen uso de más tipos de robot, 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 usará en un futuro. Aunque es verdad que las universidades disponen de algunas unidades, no siempre son accesibles para el estudiante por diversas razones.

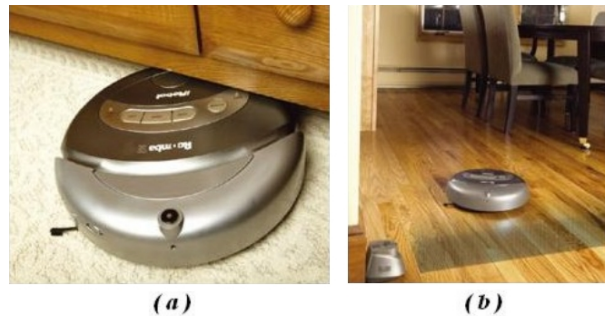


Figura 1.2: Robots educativos en universidades.

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.

No olvides incluir imágenes y referenciarlas, como la Figura 1.2.

Ni tampoco olvides de poner las URLs como notas al pie. Por ejemplo, si hablo de la Robocup<sup>1</sup>.

### 1.2.1. Números

En lugar de tener secciones interminables, como la Sección 1.1, divídelas en subsecciones.

Para hablar de números, mételes en el entorno *math* de L<sup>A</sup>T<sub>E</sub>X, por ejemplo,  $1,5Kg$ . También puedes usar el símbolo del Euro como aquí: 1.500€.

### 1.2.2. Listas

Cuando describas una colección, usa `itemize` para ítems o `enumerate` para enumerados. Por ejemplo:

- *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.

1. Primer elemento de la colección.
2. Segundo elemento de la colección.

---

<sup>1</sup><http://www.robocup.org>

**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

# Objetivos

---

*Quizás algún fragmento de libro inspirador...*

Autor, *Título*

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.

### 2.1. Descripción del problema

El objetivo principal de este trabajo de fin de grado es crear y documentar el proceso de construir tu propio brazo robot desde cero. Además, se busca que el sistema creado sea asequible y fácilmente replicable mediante el uso de la impresión 3D. El campo de uso de este robot está en la robótica educativa en universidades. Para lograr dicha meta, se ha dividido el problema en estos subobjetivos:

### 2.2. Requisitos

De este trabajo se espera un sistema robótico de bajo coste cuya fabricación completa, incluyendo hasta la última tuerca, esté por debajo de 200€.

Otro requisito es que la mayoría de las partes que componen el robot, sean imprimibles incluso en una impresora 3D de gama baja. Esto hace que cualquier persona pueda construirse el suyo sin necesitar de muchas herramientas.

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.

### **2.3. Metodología**

Qué paradigma de desarrollo software has seguido para alcanzar tus objetivos.

### **2.4. Plan de trabajo**

Qué agenda has seguido. Si has ido manteniendo reuniones semanales, cumplimentando objetivos parciales, si has ido afinando poco a poco un producto final completo, etc.

---

## Capítulo 3

# Plataforma de desarrollo

---

*Quizás algún fragmento de libro inspirador...*

Autor, *Título*

### 3.1. Python

Escribe aquí un párrafo explicando brevemente lo que vas a contar en este capítulo. En este capítulo, explica qué has usado a nivel hardware y software para poder desarrollar tu trabajo: librerías, sistemas operativos, plataformas, entornos de desarrollo, etc.

### 3.2. Grbl

GRBL 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.

### 3.3. Código G

Se trata

### 3.4. 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 Github<sup>1</sup>. Es fácilmente adquirible 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 la 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.

### 3.5. 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 *homing* (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.

---

<sup>1</sup><https://github.com/makerbase-mks/MKS-DLC32>

### 3.6. Controladores de motores

Un controlador paso a paso es el módulo hardware capaz de transformar 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 4

# 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.

### 4.1. Snippets

Puede resultar interesante, para clarificar la descripción, mostrar fragmentos de código (o *snippets*) ilustrativos. En el Código 4.1 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 4.1: Función para buscar elementos 3D en la imagen

En el Código 4.2 vemos un ejemplo escrito en Python.

### 4.2. 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 4.2: 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
}
```

### 4.3. 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 4.1 y Ecuación 4.2.

$$H = 1 - \frac{\sum_{i=0}^N \frac{(\frac{d_{js} + d_{je}}{2})}{N}}{M} \quad (4.1)$$

Ecuación 4.1: Ejemplo de ecuación con fracciones

$$v(entrada) = \begin{cases} 0 & \text{if } \epsilon_t < 0,1 \\ K_p \cdot (T_t - T) & \text{if } 0,1 \leq \epsilon_t < M_t \\ K_p \cdot M_t & \text{if } M_t < \epsilon_t \end{cases} \quad (4.2)$$

Ecuación 4.2: Ejemplo de ecuación con array y letras y símbolos especiales

### 4.4. Tablas o cuadros

Si necesitas insertar una tabla, hazlo dignamente usando las propias tablas de L<sup>A</sup>T<sub>E</sub>X, no usando pantallazos e insertándolas como figuras... En el Cuadro 4.1 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 $\mu\text{m}$
Lente	f=3.04 mm, f/2.0
Ángulo de visión	62.2 x 48.8 degrees
Lente SLR equivalente	29 mm

Cuadro 4.1: Parámetros intrínsecos de la cámara

---

## Capítulo 5

# 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 propuesto, así como los experimentos llevados a cabo para validarlos. Y con esto, cierras la memoria.

### 5.1. 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).



## 5.2. Corrector ortográfico

Una vez tengas todo, no olvides pasar el corrector ortográfico de L<sup>A</sup>T<sub>E</sub>Xa 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.