

# COLECCIÓN DE JUPYTER NOTEBOOKS PARA CURSOS DE ROBÓTICA MÓVIL

Samuel-Felipe Baltanas-Molero, Jose-Raul Ruiz-Sarmiento, Javier Gonzalez-Jimenez  
Dpto. Ingeniería de Sistemas y Automática, Instituto de Investigación Biomédica  
de Málaga (IBIMA), Universidad de Málaga, Campus de Teatinos, 29071, Málaga,  
{sambalmol,jotaraul,javiergonzalez}@uma.es

## Resumen

*Las expectativas de implantación de vehículos autónomos en nuestra sociedad (coches, UAVs, robots, etc.) está impulsando la demanda de profesionales en el campo de la robótica móvil. Dada la diversidad y complejidad de las materias que abarca la robótica, e.g. probabilidad, álgebra lineal, cálculo diferencial, mecánica, programación, inteligencia artificial, etc., se hace necesario contar con material didáctico innovador que facilite el aprendizaje y posibilite el desarrollo de problemas prácticos soportados por sus correspondientes fundamentos teóricos. Este trabajo presenta una colección de documentos interactivos para el aprendizaje en robótica, basados en la novedosa tecnología Jupyter Notebook, la cual permite al docente incluir en el mismo documento texto, ecuaciones matemáticas, recursos visuales como imágenes o videos, enlaces externos, e implementación de código. De este modo, los documentos interactivos introducen temas fundamentales de la robótica móvil situando al estudiante en un mejor contexto, incorporando los conceptos necesarios de distintas materias, y permitiéndole interactuar y analizar los resultados obtenidos. Esta herramienta está siendo integrada en diversos cursos de la Universidad de Málaga, entre ellos la asignatura obligatoria de Robótica del grado de Ingeniería Informática (Computación).*

**Palabras clave:** cuadernos interactivos, Jupyter Notebooks, robótica, docencia, aprendizaje

## 1. Introducción

El campo de la robótica móvil tiene cada vez más relevancia en nuestra sociedad, encontrando aplicaciones en ámbitos como la educación, sanidad, vigilancia, etc. [4, 16], donde drones, coches autónomos, robots asistenciales, de ayuda en las tareas domésticas, son algunos ejemplos de cómo se materializa esta tecnología [14, 11, 1]. Quizá una de sus aplicaciones más populares en nuestra vida cotidiana es la de limpieza de suelos, la cuál arrancó en el 2002 con la llegada de los aspiradores autónomos *Roomba*. Este primer éxito

originó un nuevo mercado para los robots móviles, demostrando las posibilidades que ofrecen en cuanto a mejorar la calidad de vida de las personas [19]. Hoy en día no hay duda del potencial que tiene este sector, tal y como vienen demostrando las políticas de I+D+i en la Unión Europea (programa H2020<sup>1</sup>), los Estados Unidos, China, Corea o Japón<sup>2</sup>, con un apoyo claro que se plasma en programas específicos de financiación. Además, cada vez más empresas internacionales como *Google* o *Facebook* están interesadas en sus posibilidades, prestando especial atención a las nuevas compañías y aplicaciones que emergen en este campo.

Este interés ha propiciado un aumento substancial de la demanda de ingenieros con un perfil orientado hacia las tecnologías relacionadas, lo que requiere una formación de calidad en las materias involucradas [13]. Esta necesidad es palpable en la cantidad de nuevos programas educativos relacionados con la robótica, que cada vez están más presentes en las universidades y en plataformas *online* de aprendizaje como *Coursera* o *Udacity*.

En general, estos cursos cubren los aspectos fundamentales dentro de este campo, como puedan ser: movimiento del robot, percepción del entorno, construcción de mapas y localización, planificación, o utilización arquitecturas de control robóticas, entre otros. El aprendizaje de estas temáticas requiere la incorporación tanto de los fundamentos teóricos en las que se basan como de ejercicios y demostraciones para que los alumnos los pongan en práctica. Típicamente, los ejercicios prácticos están formados por una serie de problemas que el estudiante ha de resolver, los cuales pueden incluir código a completar. Aunque este enfoque tradicional es una metodología válida en diversos campos, su aplicación a la robótica presenta una serie de inconvenientes. Esto se debe principalmente a la diversidad de materias que abarca y los conceptos abstractos involucrados, lo cual dificulta la presentación del problema en su correcto contexto, produciendo lecciones prácticas donde los conceptos teóricos pueden resultar inconexos e incompletos.

<sup>1</sup><https://eshorizonte2020.es/>

<sup>2</sup><https://www.hansonrobotics.com/research/>

En este trabajo presentamos una colección de documentos interactivos para cursos de robótica móvil basados en la tecnología Jupyter Notebook [6, 15]. Esta tecnología se está integrando recientemente en multitud de cursos *online*, más dinámicos a la hora de adaptarse a nuevas tendencias, dada su capacidad para entrelazar el contexto del problema, visualizaciones de datos y código interactivo. Su aplicación en el contexto académico permite crear tareas accesibles y de comprensión guiada, gracias al uso de componentes de texto, ecuaciones matemáticas, imágenes, vídeos, recursos externos, código, etc. (ver Fig. 1) [8]. Los documentos interactivos (también llamados *cuadernos* o *historias computacionales*) que acompañan a este trabajo están pensados como material introductorio a la robótica móvil, y agrupan temas a tratar compuestos de problemas relacionados. De este modo, suponen un material docente completo y contextualizado, que añade la interactividad y las posibilidades de una aplicación a la tradicional presentación de contenidos de un libro. Para la realización de los componentes de programación se ha optado por el lenguaje Python [9], dada su reciente popularidad y capacidad de integración con Jupyter Notebook. También cabe mencionar que Python puede ser usado en la arquitectura de control robótica más popular, Robot Operating System (ROS) [10], añadiendo un extra a su práctica por parte del estudiante. Actualmente los cuadernos presentados se están integrando en asignaturas de la Universidad de Málaga con resultados iniciales muy prometedores.

Cabe señalar que se mantienen dos versiones de los documentos interactivos: una versión *para estudiantes*, incluyendo una serie de ejercicios a completar, y una versión *para docentes* que, además, incluye las soluciones a dichos ejercicios. La versión para estudiantes está disponible públicamente en: <https://github.com/jotaraul/jupyter-notebooks-for-robotics-courses>, mientras que la versión para docentes puede ser solicitada por cualquier docente interesado. Las próximas secciones proveen más detalles acerca de las tecnologías utilizadas (ver Sec. 2), los distintos cuadernos elaborados (Sec. 3), su posible uso como material docente (ver Sec. 4), y una breve discusión sobre este trabajo (Sec. 5).

## 2. Tecnologías utilizadas

### 2.1. Jupyter Notebook

Jupyter Notebook [6] es un *framework* para la creación de documentos que integran código interactivo con elementos textuales y gráficos. Esta plataforma fue creada por *Project Jupyter*, un

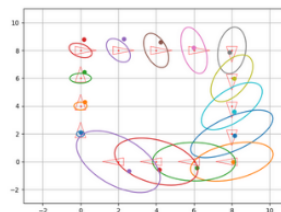
### 3.3.1 Analytical form

#### Context

Just as we did in lesson 3.1, the analytic form of the odometry motion model uses the composition of poses to model the robot's movement, providing only a notion of how much the pose has changed, not how did it get there.

$$p_1 \oplus \Delta p = \begin{bmatrix} x_1 + \Delta x \cos \theta_1 - \Delta y \sin \theta_1 \\ y_1 + \Delta x \sin \theta_1 + \Delta y \cos \theta_1 \\ \theta_1 + \Delta \theta \end{bmatrix}$$

#### Example



```
class Robot():
    def __init__(self, x, P, Q):
        self.pose = x
        self.P = P
        self.true_pose = self.pose
        self.Q = Q
```

Figura 1: Contenido típico de un cuaderno, incluyendo: cabecera, texto, imágenes, fórmulas matemáticas y celdas de código.

grupo sin ánimo de lucro que tiene como objetivo crear herramientas libres (*open-source*) para la computación científica y la ciencia de los datos. Jupyter Notebook está compuesto por varias herramientas:

**Notebook.** Formato que organiza el código en una serie de celdas que pueden ser editadas y ejecutadas fácilmente, mostrando los resultados debajo de la celda correspondiente. Las celdas pueden contener texto, gráficas, visualizaciones, fórmulas matemáticas y código.

**Aplicación web.** Editor de código y texto y plataforma de cómputo. El hecho de ser una aplicación web permite, si fuera necesario, alojar los cuadernos en un servidor. Sin embargo, puede ser instalada de forma local por los alumnos fácilmente, *e.g.* usando paquetes distribuidos mediante *pip* o *Anaconda*<sup>3</sup>.

**Kernel.** *Backend* encargado de ejecutar el código de una celda. Originalmente, solo había un *kernel* para el lenguaje de programación Python, sin embargo la gran popularidad y la naturaleza *open-source* de la plataforma ha fomentado la creación de *kernels* para más de 50 lenguajes de programación adicionales<sup>4</sup>.

<sup>3</sup><https://www.anaconda.com/distribution/>

<sup>4</sup><https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>

Con respecto a las celdas de texto, estas pueden formatearse empleando Markdown<sup>5</sup>, que permite incluir fácilmente elementos típicos en otros procesadores de texto, como pueden ser: fórmulas matemáticas, tablas, listas, imágenes, enlaces, código estático, etc. en un formato de texto plano. Markdown destaca por su sintaxis limpia y sencilla, no obstante, también puede usarse HTML en celdas de texto, permitiendo por ejemplo la reproducción de vídeos desde las mismas. La siguiente sección describe con más detalle el lenguaje de programación usado para las celdas de código.

## 2.2. Python

El lenguaje escogido para la implementación de código es Python, uno de los más populares actualmente gracias a su sintaxis clara y concisa, y a la disponibilidad de una amplia variedad de librerías para la mayoría de casos de uso [18]. Estas características lo hacen un lenguaje idóneo para la creación de prototipos y la experimentación, eliminando el uso de convenciones de código complejas y los tiempos de compilación. Concretamente, el material desarrollado hace uso de las siguientes librerías de Python:

**Numpy.** Una librería para facilitar la computación con vectores y matrices multidimensionales o de gran tamaño, incluyendo además gran cantidad de funciones matemáticas de alto nivel, tales como el producto de matrices, funciones trigonométricas, etc.

**Scipy.** Librería para el cómputo matemático que dio origen a *Numpy* y proporciona ciertas herramientas complementarias a esta (métodos estadísticos, de álgebra lineal, o del procesamiento de señales) que son de gran utilidad en los cuadernos desarrollados.

**Matplotlib.** Visualizar el problema es una parte fundamental del proceso de aprendizaje, especialmente en el campo de los robots móviles. *Matplotlib* es una librería para la creación de figuras con soporte nativo en Jupyter Notebook, pudiendo mostrar figuras dentro de cuadernos o incluso refrescarlas para crear animaciones.

Con respecto al ámbito de la robótica móvil, cabe destacar que Python es uno de los dos lenguajes de programación soportados por la plataforma *Robot Operating System (ROS)* [10], la arquitectura robótica más popular. Esta provee librerías y otras utilidades necesarias para crear aplicaciones

<sup>5</sup><https://www.markdownguide.org/getting-started>

modulares para robots. De esta forma, el estudiante obtiene el extra de practicar un lenguaje de uso general que, además, puede ser usado en un *framework* específico para robots.

## 3. Jupyter Notebooks desarrollados

Esta sección describe los cuadernos para cursos de robótica desarrollados que acompañan este trabajo. Estos se encuentran divididos según la temática que abarcan, aunque la mayoría comparten una estructura similar:

1. **Introducción.** Descripción general del problema/tema tratado, incluyendo conceptos teóricos y fórmulas relacionadas, así como los resultados esperados.
2. **Librerías.** Celda de código que importa al entorno las librerías necesarias para completar el cuaderno.
3. **Código auxiliar (opcional).** En el caso de cuadernos complejos, celdas de código auxiliar que ofrecen ciertas utilidades adicionales listas para ser usadas.
4. **Ejercicios.** Cada cuaderno estará formado por varias tareas a completar por el alumno, entrelazando celdas descriptivas con celdas de implementación de código.
5. **Demostraciones.** Permiten comprobar el correcto funcionamiento del código realizado por el alumno e ilustrar en mayor medida los conceptos relacionados mediante resultados gráficos.

Los cuadernos elaborados abordan temas fundamentales como el movimiento de un robot móvil (Sec. 3.2), cómo este percibe el entorno (Sec. 3.3), como construir mapas (Sec. 3.5) y localizarse en el mismo (Sec. 3.4), etc. Las siguientes secciones los describen en más detalle.

### 3.1. Fundamentos probabilísticos

**Problema.** A diferencia de la mayoría de robots industriales, que realizan tareas repetitivas (*e.g.* en líneas de montaje), los robots móviles están expuestos a una mayor variedad de situaciones y condiciones del entorno de trabajo, lo que provoca incertidumbre en el estado de ambos, el robot y el entorno en sí. Esta incertidumbre esta originada por muchos factores: entornos dinámicos, errores en las medidas de los sensores, inexactitud de los modelos empleados (por ejemplo, el modelo de movimiento o el mapa), entre otros.

Para afrontar esta incertidumbre, la robótica recurre a métodos probabilísticos [17]. El principio fundamental de la *robótica probabilística* se basa en representar explícitamente la incertidumbre mediante el uso de distribuciones probabilísticas. Este concepto es parte básica de la robótica moderna y, por tanto, es necesario conocer los fundamentos estadísticos en los que se basa.

**Material propuesto.** El primer grupo de cuadernos se centra en la *distribución Gaussiana* y sus propiedades, dado su uso recurrente en la robótica probabilística y, por lo tanto, en siguientes cuadernos. Tres son los cuadernos que exploran distintos aspectos:

- **Distribución Gaussiana.** Este cuaderno sirve como introducción al tema, incluyendo como se define una distribución Gaussiana y como generar muestras aleatorias a partir de esta (Fig. 2).
- **Propiedades.** Este cuaderno persigue ilustrar las propiedades y operaciones más relevantes de las *Gaussianas*, tales como el teorema central del límite, la suma, el producto, la transformación lineal, etc. permitiendo al estudiante experimentar con distintos parámetros de las distribuciones.
- **Distribuciones multidimensionales.** Reitera conceptos y propiedades de cuadernos anteriores, demostrando su aplicación a distribuciones multidimensionales.

En ciertos casos, es útil observar cómo los parámetros del problema pueden afectar a los resultados obtenidos. Para ello se provee al alumno con la posibilidad de interactuar con el código mediante elementos como la barra deslizante en la Fig. 2. Estos *widgets* permiten pasar nuevos valores a una función sin tener que volver a ejecutar una celda manualmente. Existe una amplia variedad de *widgets*, incluyendo muchos de los elementos propios de HTML como botones, selectores, o entrada de texto, pudiendo ser usados para crear demostraciones más dinámicas e interactivas.

### 3.2. Movimiento del Robot

**Problema.** Otro aspecto fundamental de la robótica es la locomoción de robots, en especial aquellos que se desplazan mediante ruedas. Este no es un tema trivial, ya que muchos de estos robots presentan restricciones no holonómicas, *i.e.* no se pueden mover libremente por el espacio, teniendo sus desplazamientos restringidos a trayectorias circulares. La mayoría de los robots con

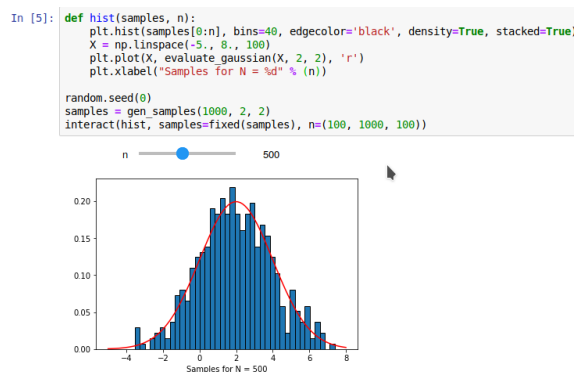


Figura 2: Histograma de un conjunto de muestras aleatorias de una distribución Gaussiana.

ruedas presentan un sistema de movimiento diferencial que emplea dos actuadores independientes (uno por rueda). Los sistemas de movimiento son de interés para entender como varía la posición de un robot en base a un comando de velocidad recibido por su controlador. La posición global del robot tras una serie de comandos de movimiento puede obtenerse mediante la composición de posiciones o poses.

No obstante, la posición real del robot puede diferir de la esperada debido a errores causados por deslizamiento de las ruedas, calibración incorrecta de los sensores/actuadores usados, resolución de las medidas, suelo irregular, etc. Para afrontar estas fuentes de incertidumbre, se necesita expresar el movimiento en términos probabilísticos mediante el uso de modelos de movimiento, entre los cuales los más populares son los basados en velocidad, y los basados en odometría. Independientemente de que modelo usemos, el resultado del movimiento será una distribución de probabilidad (Gaussiana en este caso), representando una nueva pose del robot.

**Material propuesto.** El objetivo de los siguientes cuadernos es familiarizar a los estudiantes con los distintos aspectos relativos al movimiento y los comandos de control. Concretamente:

- **Composición de poses con ruido.** Simulación del movimiento de un robot mediante el uso de la *composición de poses y ruido Gaussiano*. El objetivo es usar comandos para que el robot se mueva siguiendo una trayectoria rectangular, un patrón de movimiento útil para ilustrar el funcionamiento de distintos algoritmos.
- **Conducción diferencial con comandos de velocidad.** Introducción a las ecuaciones del sistema de movimiento diferencial. Posteriormente, el estudiante lo pondrá en prácti-

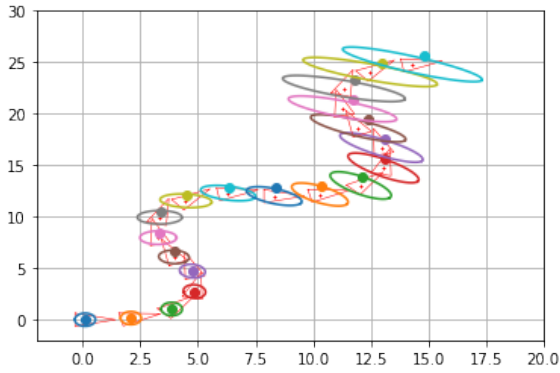


Figura 3: Movimiento de un robot mediante comandos de velocidad, sin tener en cuenta el ruido (en rojo), considerando un ruido Gaussiano (como puntos) y la incertidumbre sobre su pose (como elipse).

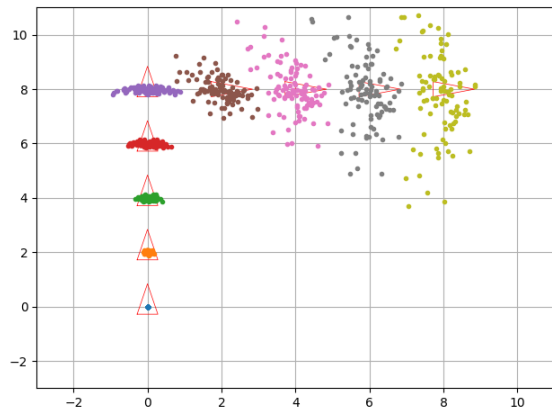


Figura 4: Movimiento de un robot mediante comandos de odometría (forma muestreada), sin tener en cuenta el ruido (en rojo), como muestras considerando ruido (nube de puntos).

ca mediante el uso de *comandos de velocidad* para dirigir al robot siguiendo una trayectoria serpentina (Fig. 3)

- Conducción diferencial con comandos de odometría.** Finalmente para mostrar el uso de *comandos de odometría*, el estudiante los usará tanto en *forma analítica*, como en *forma muestreada*. Esta última es de especial interés por su uso en filtros de partículas [3], y para representar de forma más verosímil el movimiento de una base robótica (Fig. 4)

### 3.3. Sensores

**Problema.** La obtención de información del entorno es un campo bastante amplio dada la variedad de sensores que existen: sonares, cámaras, láseres infrarrojos, etc. No obstante, se suele recu-

rrir a sensores de distancia y rumbo para ilustrar los conceptos principales dado que son simples pero, a su vez, altamente ilustrativos.

**Material propuesto.** Se ha diseñado un cuaderno en el que los estudiantes han de implementar el modelo de un sensor de distancia y rumbo, y manipular la información que proporciona de distintas maneras. Este cuaderno introduce algunos conceptos fundamentales empleados en algoritmos más sofisticados (localización, construcción de mapas, etc) tales como: transformar una observación desde la perspectiva del robot al sistema de coordenadas global (del mundo) o vice versa, convertir la observación a la perspectiva de otro robot, o combinar la información de distintas observaciones independientes. También refuerza conceptos que, hasta ahora, solo habían sido mencionados, *e.g.* la composición inversa de poses y los jacobianos del modelo de distancia y dirección, los cuales han de ser implementados por el alumno.

### 3.4. Localización

**Problema.** La localización de un robot consiste en estimar la pose de este dentro de un mapa del entorno en el que se desenvuelve [17]. Esta necesidad surge del hecho de que los robots no son capaces de observar su pose directamente, teniendo que inferirla en base a información externa. La mayoría de las tareas que conciernen a los robots móviles implican navegar por un entorno de forma efectiva, *e.g.* los aspiradores han de navegar por todas las áreas de la casa, un vehículo autónomo ha de llegar a su destino, etc. Para esto es imprescindible ser capaces de orientarse dentro de un entorno específico. No obstante, existen una variedad de problemas que dificultan esta tarea en el mundo real, como entornos dinámicos, lugares indistinguibles, sensores ruidosos, etc., y que motivan de nuevo el uso de modelos probabilísticos.

**Material propuesto.** El cuaderno desarrollado propone al estudiante implementar ciertos algoritmos para estimar la pose de un robot, dados una serie puntos de referencia que actúan como balizas, cuyo homólogo real podrían ser satélites de GPS o bases WiFi. En este caso, los algoritmos empleados para tratar el problema son el método de los mínimos cuadrados y el filtro extendido de Kalman (EKF). Una vez completada su implementación, el alumno podrá probar su funcionamiento usando el código auxiliar proporcionado (ver Fig. 5).

### 3.5. Construcción de mapas

**Problema.** La creación de modelos del entorno, comúnmente llamados *mapas*, a partir de la infor-



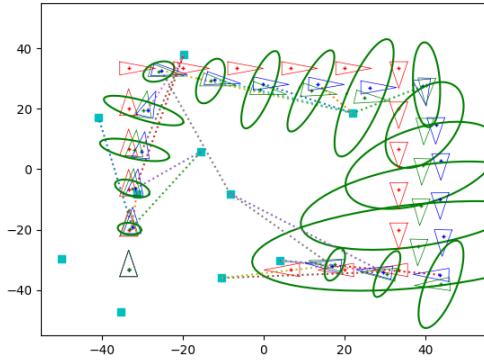


Figura 5: Algoritmo EKF aplicado localización de robots. La pose esperada del robot aparece en rojo, la pose real en azul y la pose estimada usando EKF y su elipse de confianza en verde. Los cuadrados representan son los landmarks del entorno.

mación percibida del mismo es otro aspecto necesario en muchas aplicaciones robóticas, ya que se requieren para localizar al robot, planificar sus movimientos, etc [12]. Cabe señalar que en los cuadernos anteriores se ha supuesto que el mapa del entorno era conocido para el robot. No obstante, algunas aplicaciones no pueden depender de mapas existentes, ya sea porque pueda estar obsoleto o no esté disponible. La creación de mapas es un problema complejo, de una alta dimensionalidad proporcional al tamaño del entorno de trabajo, y que tampoco está exenta de distintas fuentes de incertidumbre.

**Material propuesto.** El cuaderno que aborda este problema propone la creación de un mapa de un entorno compuesto de una serie de puntos característicos, siendo el robot capaz de distinguirlos inequívocamente (ver Fig. 6). Al contrario que los anteriores ejercicios, la pose del robot es conocida perfectamente durante toda la ejecución. Para enfrentarse a este reto, los alumnos harán uso nuevamente del algoritmo EKF manteniendo sus aspectos básicos, aunque su implementación tiene algunas diferencias significativas cuando se aplica a la construcción de mapas.

### 3.6. SLAM

**Problema.** El problema de Localización y Mapeado Simultáneos (*Simultaneous Localization And Mapping, SLAM*) es el pilar central de las aplicaciones robóticas modernas, ya que habilita a un robot a operar en un entorno desconocido sin información previa sobre él, comportándose así de forma autónoma [2, 20]. Concretamente, SLAM consiste en construir un mapa a la vez que se es-

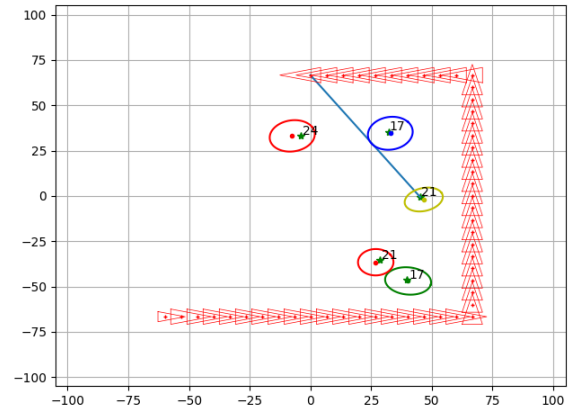


Figura 6: Ejecución del algoritmo EKF aplicado a la construcción de mapas. Las estrellas verdes representan la pose real de las landmarks. Las diferentes elipses representan la confianza de cada pose del mapa estimado.

tima el estado del robot, *i.e.* su posición dentro de él. El aspecto que más dificulta este problema es que dependemos únicamente de la información de los sensores propioceptivos y exteroceptivos, los cuales introducen incertidumbre en el sistema, teniendo que recurrir nuevamente a técnicas probabilísticas [3, 17]. La existencia de esta incertidumbre imposibilita crear un modelo exacto del entorno, sin embargo estas técnicas lo aproximan adecuadamente.

**Material propuesto.** La tarea elegida para poner en práctica los conceptos de SLAM es similar a la de los dos cuadernos anteriores, dado que SLAM es una combinación de los problemas de localización y construcción de mapas, resultando en una demostración similar a la Fig. 3.5. Los alumnos volverán a hacer uso del algoritmo EKF, que ahora incorporará todos los conceptos aprendidos en las Sec. 3.4 y Sec. 3.5. Una vez completado, los estudiantes deberán evaluar el comportamiento del algoritmo, para lo que han de recoger y analizar ciertos indicadores de su efectividad, *i.e.* el error y el grado de confianza para la pose y el mapa.

### 3.7. Planificación del movimiento

**Problema.** La planificación del movimiento consiste en computar los comandos de velocidad y giro necesarios para comandar al robot de tal manera que navegue de un punto de origen  $A$  a otro destino  $B$  [7]. La complejidad de este problema yace en varios factores: la propia geometría del robot (no considerada en cuadernos interiores), la variedad de obstáculos en el entorno, o las limitaciones que impone el movimiento no holonómico. A su vez, este problema considera dos aspectos:

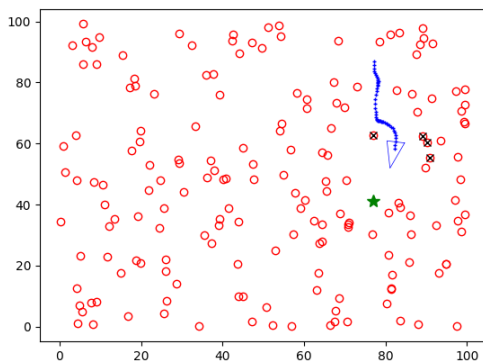


Figura 7: Movimiento de un robot usando campos potenciales. La figura muestra unos obstáculos (en rojo), una meta (en verde) y un robot (en azul). Los obstáculos marcados con una cruz son aquellos en el radio de influencia del algoritmo.

- **Navegación reactiva.** Enfocada a evitar obstáculos cercanos a partir de la información proporcionada por los sensores [5].
- **Navegación global.** Que tiene como función encontrar la mejor ruta posible para alcanzar una localización objetivo.

**Material propuesto.** Los estudiantes se sitúan en el contexto de un robot rodeado de obstáculos que ha de alcanzar una meta. Para conseguirlo han de completar un algoritmo de navegación reactiva basado en campos potenciales, un método simple pero eficaz para la evitación de obstáculos. La idea principal de este algoritmo es el uso de fuerzas repulsivas emanando de obstáculos cercanos, y una fuerza atrayente generada por la meta en cuestión. Una vez completado el algoritmo, los estudiantes son alentados a experimentar usando la demostración proporcionada (ver Fig. 7), analizando las limitaciones inherentes del algoritmo y discutiendo posibles formas de mitigarlas.

#### 4. Uso para la enseñanza

El propósito de los documentos interactivos que acompañan a este trabajo es convertirse en un recurso útil para la enseñanza, principalmente en cursos relacionados a la robótica móvil. Estos cuadernos han sido utilizados por estudiantes universitarios de la Universidad de Málaga con resultados prometedores, y creemos que podrían ser de utilidad para otros docentes.

Para su uso académico, las herramientas de este tipo han de tener dos características principales: completitud y facilidad de implantación y uso. Con completitud nos referimos a que se cubran

todos los aspectos esenciales de la robótica móvil: movimiento, localización, etc. Siendo una herramienta completa, puede ser aplicada en cursos de robótica sin la necesidad de depender en materiales adicionales. A lo largo de la Sec. 3 se ha mostrado que los cuadernos en cuestión cumplen este requisito. También es importante para su uso académico la facilidad de instalación y uso. Hoy en día existen ciertas distribuciones, como Anaconda, que facilitan el proceso de instalación, tanto para el *framework* Jupyter Notebook, como para Python y sus librerías. La instalación puede realizarse individualmente en un ordenador personal, o en un servidor para facilitar el acceso a los estudiantes. Además existen servicios *online* útiles, como *Binder*, que convierten un repositorio *Git*, como el que contiene nuestros cuadernos, en un entorno interactivo sin necesitar una instalación local. Estas alternativas para la instalación permiten un uso sencillo de la herramienta.

#### 5. Discusión

En este trabajo se han presentado una serie de documentos interactivos para la formación en robótica móvil basados en la novedosa tecnología *Jupyter Notebook*. Los documentos suponen una herramienta que aúna conceptos teóricos con la práctica de los mismos. Para la descripción de los primeros se emplean campos o *celdas* de texto, ecuaciones matemáticas, imágenes, vídeos y enlaces externos, mientras que la parte práctica se compone de celdas de código a completar. Dicho código se desarrolla empleando *Python* y librerías potentes y populares como *Numpy*, *Scipy* y *Matplotlib*, lo cual nos permite simular de forma visual el comportamiento de robots móviles, sin añadir la dificultad adicional de aprender un *framework* específico para la robótica. La versión para estudiantes de los documentos interactivos desarrollados se ha hecho pública en <https://github.com/jotaraul/jupyter-notebooks-for-robotics-courses>.

#### Agradecimientos

Este trabajo se ha desarrollado en el marco del proyecto WISER (DPI2017-84827-R), financiado por el Ministerio de Ciencia e Innovación contando con fondos del Fondo Europeo de Desarrollo Regional (FEDER), y por un contrato posdoctoral del I-PPIT de la Universidad de Málaga.

#### English summary

### A COLLECTION OF JUPYTER NOTEBOOKS FOR MOBILE ROBOTICS COURSES

#### Abstract.

The recent popularity of mobile robotics has resulted in an increasing demand of practitioners with a solid theoretical and practical background in the field. Nevertheless, the diversity of related topics e.g. probability, algebra, computing, etc. complicates the development of learning materials to reinforce these fundamentals. This paper presents a collection of interactive notebooks targeted at robotics courses, making use of the Jupyter Notebook framework, which allows the use of rich text components within a document, e.g. mathematical equations, multimedia content such as images or videos, external links and interactive code. Therefore, these interactive documents are able to introduce the mobile robotics fundamentals, in a proper context, interweaving the core concepts of the different fields, and allowing the student to analyze and interact with the results. The developed notebooks are meant to become a useful tool in academia, particularly in our courses at the University of Malaga, where they are being integrated.

**Keywords:** interactive notebooks, Jupyter Notebooks, robotics, teaching material, learning

## Referencias

- [1] Bogue, R., (2016) Growth in e-commerce boosts innovation in the warehouse robot market. *Industrial Robot: An International Journal*, 43(6):pp. 583–587.
- [2] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J., (2016) Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):pp. 1309–1332.
- [3] Fernández-Madrigal, J.-A. and Blanco, J.-L., (2012) *Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods*. IGI Global.
- [4] Gonzalez-Jimenez, J., Galindo, C., and Ruiz-Sarmiento, J. R., (2012) Technical improvements of the giraff telepresence robot based on users' evaluation. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*.
- [5] Gonzalez-Jimenez, J., Ruiz-Sarmiento, J., and Galindo, C., (2013) Improving 2d reactive navigators with kinect. In *10th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*.
- [6] Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J. B., Grout, J., Corlay, S., et al., (2016) Jupyter notebooks—a publishing format for reproducible computational workflows. In *EL-PUB*, pp. 87–90.
- [7] LaValle, S. M., (2006) *Planning algorithms*. Cambridge university press.
- [8] Perez, F. and Granger, B. E., (2015) Project jupyter: Computational narratives as the engine of collaborative data science. Retrieved September, 11(207):p. 108.
- [9] Python Software Foundation, (2018). Python language reference, version 3.7.x. Online, last accessed: April 28, 2019.
- [10] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y., (2009) Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, p. 5. Japan.
- [11] Restas, A., (2015) Drone applications for supporting disaster management. *World Journal of Engineering and Technology*, 3(03):p. 316.
- [12] Ruiz-Sarmiento, J.-R., Galindo, C., and Gonzalez-Jimenez, J., (2017) Building multiversal semantic maps for mobile robot operation. *Knowledge-Based Systems*, 119:pp. 257–272.
- [13] Ruiz-Sarmiento, J. R., Galindo, C., and Gonzalez-Jimenez, J., (2017) Experiences on a motivational learning approach for robotics in undergraduate courses. 11th annual International Technology, Education and Development Conference (INTED2017), pp. 3803–3811.
- [14] Ruiz-Sarmiento, J. R., Galindo, C., and Gonzalez-Jimenez, J., (2017) Robot@ home, a robotic dataset for semantic mapping of home environments. *The International Journal of Robotics Research*, 36(2):pp. 131–141.
- [15] Ruiz-Sarmiento, J. R., Monroy, J., Moreno, F.-A., and Gonzalez-Jimenez, J., (2019) A tutorial on object recognition by machine learning techniques using python. 13th International Technology, Education and Development Conference (INTED2019), pp. 3321–3330.
- [16] Song, G., Yin, K., Zhou, Y., and Cheng, X., (2009) A surveillance robot with hopping capabilities for home security. *IEEE Transactions on Consumer Electronics*, 55(4):pp. 2034–2039.
- [17] Thrun, S., (2002) Probabilistic robotics. *Commun. ACM*, 45(3):pp. 52–57.
- [18] TIOBE Index, (2019). Tiobe-the software quality company. Online, last accessed: April 28, 2019.
- [19] Vaussard, F., Fink, J., Bauwens, V., Rétornaz, P., Hamel, D., Dillenbourg, P., and Mondada, F., (2014) Lessons learned from robotic vacuum cleaners entering the home ecosystem. *Robotics and Autonomous Systems*, 62(3):pp. 376–391.
- [20] Younes, G., Asmar, D., Shammas, E., and Zelek, J., (2017) Keyframe-based monocular SLAM: design, survey, and future directions. *Robotics and Autonomous Systems*, 98:pp. 67–88.



© 2019 by the authors.  
Submitted for possible  
open access publication  
under the terms and conditions of the Creative Commons Attribution CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>).