

# Guiado y Navegación de Robots: Práctica 1

---

Fernando Matía

# 1. Objetivos del Trabajo



# Objetivos del Trabajo

- Permitir al alumno familiarizarse con los temas principales que aparecen a la hora de programar la navegación de un robot en presencia de incertidumbre, entre otros:
  - Sistemas de locomoción.
  - Elección de sensores.
  - Algoritmos de estimación del estado (localización).
  - Algoritmos de control de movimientos.
  - Algoritmos de planificación de trayectorias.

# Objetivos del Trabajo

- Para ello, se implementará en simulación una aplicación de navegación de un robot escogida libremente.
- Para la realización del trabajo se seguirá un procedimiento similar al siguiente:
  1. Plantear un problema que requiera la navegación de un robot terrestre, definiendo el mapa del entorno sobre el que se va a trabajar. Puede escogerse una de las aplicaciones vistas en el tema de introducción de la asignatura, o bien cualquier otra aplicación de la robótica, siempre que reúna unas características similares.
  2. Utilizar el sistema de locomoción de tipo diferencial que contiene el simulador facilitado. Determinar la incertidumbre asociada al modelo cinemático del mismo.

# Objetivos del Trabajo

3. Escoger el sistema de percepción que se estime más adecuado para las tareas de localización y control reactivo del robot. El simulador dado dispone para ello de los siguientes sensores: ultrasonidos, telémetro láser en modo nube de puntos y telémetro láser en modo detección de balizas reflectantes. Modelar su incertidumbre mediante la calibración de los mismos.
4. Programar el algoritmo de localización que se decida utilizar y probarlo moviendo el robot por el entorno mediante comandos de tele-operación. Experimentar con el algoritmo y analizar la influencia de los principales parámetros del filtro en la bondad de la estimación conseguida, mostrando gráficamente la evolución de aquellos que considere más significativos.

# Objetivos del Trabajo

5. Programar un **controlador** básico que permita que el robot vaya pasando por una secuencia de puntos de una trayectoria previamente definida. Programar algún **controlador reactivo** que permita **bordear obstáculos** que se encuentren en la trayectoria o navegar por un paso estrecho.
  6. Programar e integrar con el controlador anterior un **planificador** capaz de generar trayectorias en presencia de **obstáculos**, para el mapa del entorno escogido.
  7. Preparar un **demostrador** final que integre todos los componentes anteriores.
  8. Redactar la **documentación** del trabajo llevado a cabo.
- En ningún caso se implementarán algoritmos de SLAM.

## 2. Requisitos de Realización del Trabajo



- Requisitos de realización del trabajo
  - El trabajo se llevará a cabo en equipos de tres alumnos, que deberán estar dados de alta en Moodle.
  - Sólo en caso de que el número de alumnos no sea múltiplo de tres, se permitirá algún equipo formado por 2 personas.
  - El trabajo a desarrollar deberá repartirse entre los alumnos que forman cada equipo de manera equilibrada.
  - El reparto de roles deberá quedar claramente establecido.

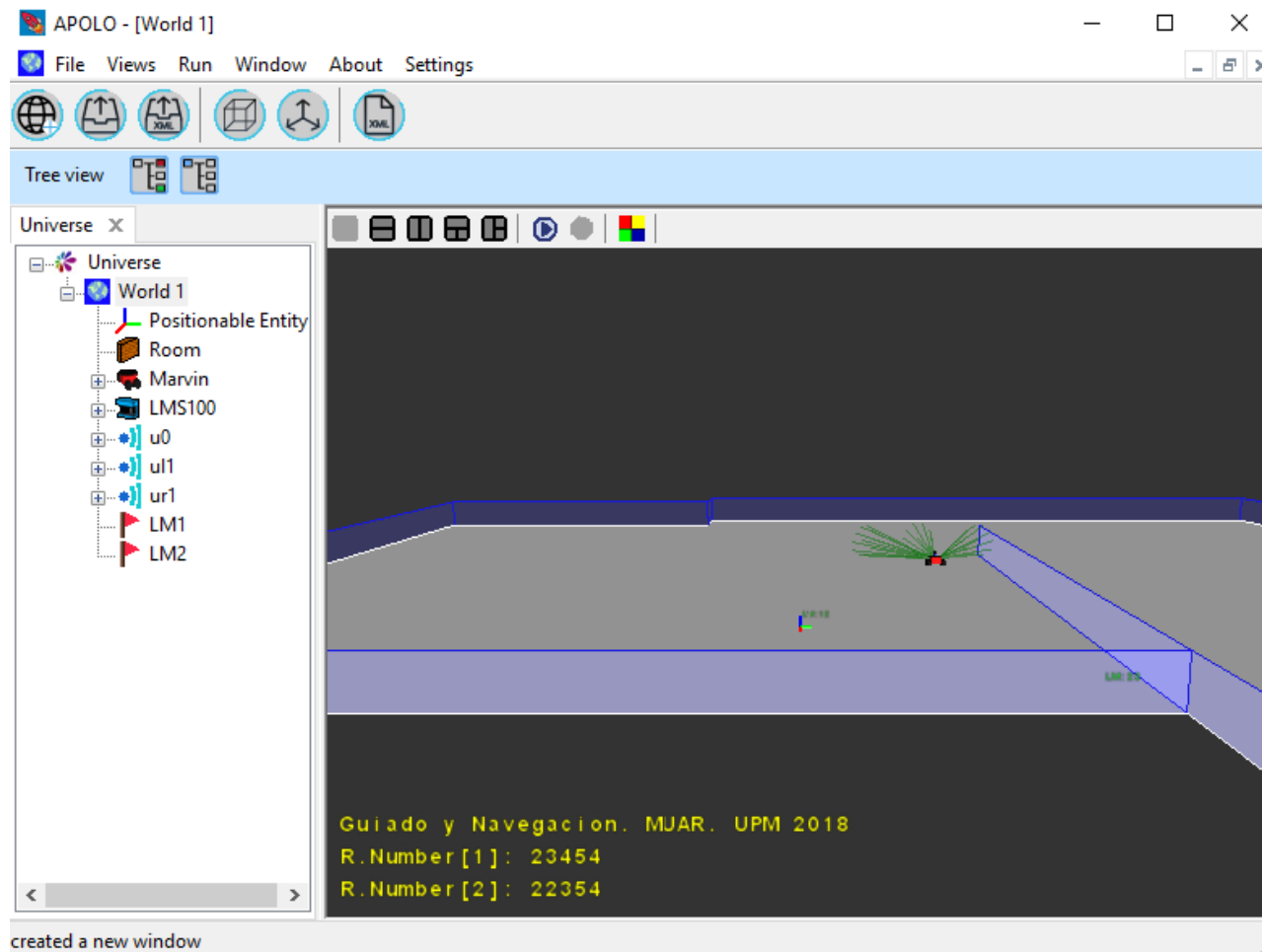


# 3. Herramientas para el Desarrollo del Trabajo

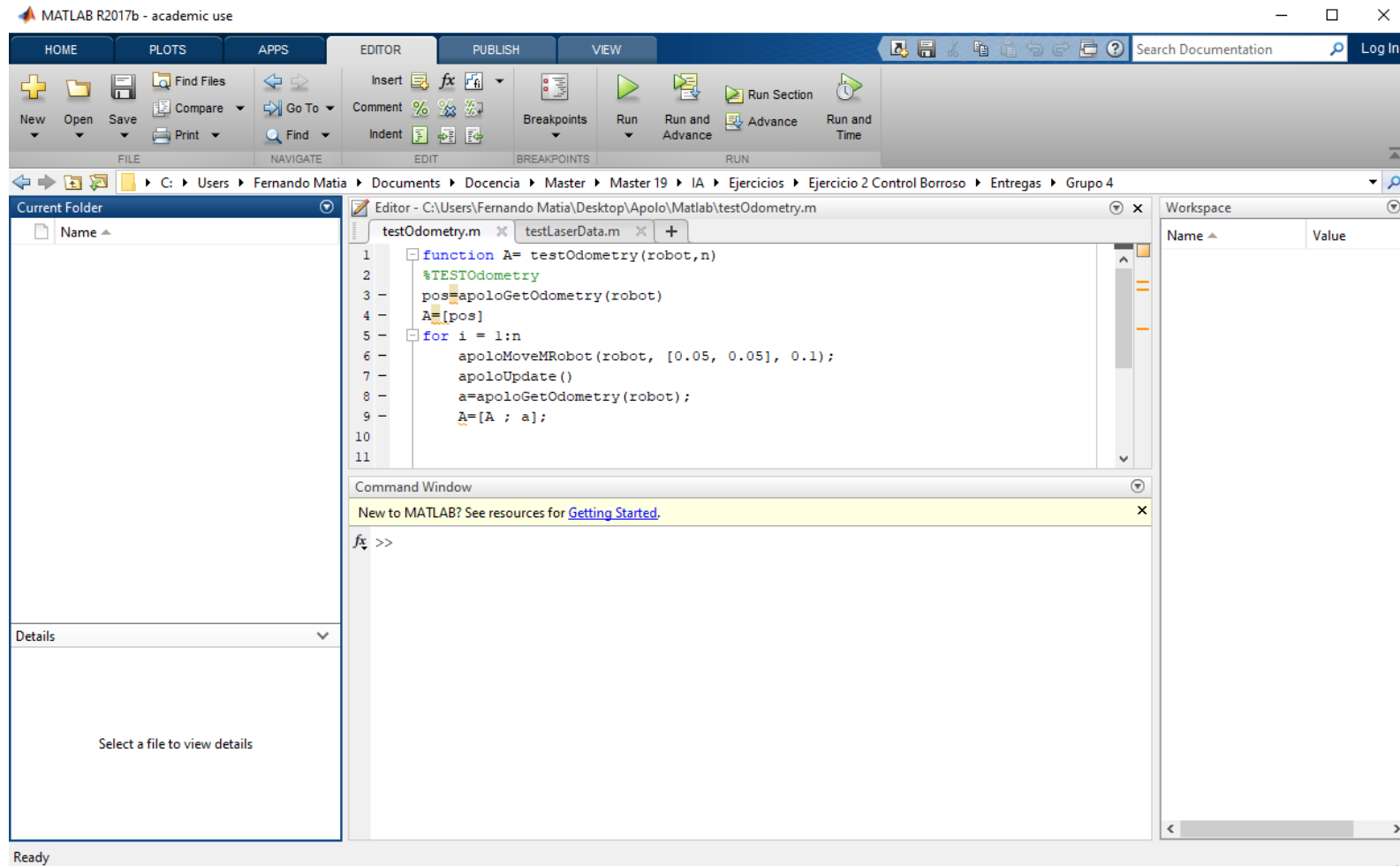


- Para el desarrollo del trabajo se recomienda la utilización de dos herramientas software:
  - **Apolo:** Entorno de simulación para sistema operativo Windows, desarrollado por profesores de la ETSIDI (UPM). El software se puede descargar de Moodle. Permite la definición del mapa del entorno y la visualización gráfica del robot y de sus sensores en dicho entorno. Simula el modelo cinemático del robot y el modelo de medidas de los sensores.
  - **MATLAB:** (licencia para estudiantes de MathWorks). Permite la implementación de los algoritmos de localización, control y planificación.

# Herramientas



- Apolo
  - Instalación en el escritorio
  - Arrancar el simulador
  - Definir el entorno en ficheros xml.
  - Cargar GyNPractica1.xml
  - Poner en marcha la simulación (Run)



- MATLAB
  - Arrancar la versión de MATLAB instalada en los ordenadores de prácticas.
  - Añadir al *path* de búsqueda el directorio apolo/Matlab.
  - Comprobar tecleando *ApoloUpdate* que se comunica con Apolo.
  - Scripts de ejemplo:
    - **testOdometry**('Marvin',100): mueve el robot en círculo y recoge los datos de la odometría.
    - **testLaserData**('LMS100','Marvin',100): permite ir viendo cada medio segundo las capturas del sensor laser.
    - **apoloGetLaserLandMarks**('LMS100'): permite conocer la posición de las balizas.
  - Para *resetear* la posición del robot:
    - `apoloPlaceMRobot('Marvin',[0 0 0],0);`
    - `apoloResetOdometry('Marvin');`
    - `apoloUpdate;`

# Herramientas

- La documentación de Apolo explica cómo comunicar Apolo con MATLAB.

- Ejemplo de filtro extendido de Kalman en MATLAB.
- Inicialización:
  - Parámetros del filtro
  - Trayectoria inicial
- Bucle principal:
  - Fija comando de velocidad para seguir la trayectoria
  - Simula el avance real del robot
  - Estima el avance del robot
  - Simula las medidas de los sensores
  - Estima las medidas de los sensores
  - Corrige la estimación de la posición del robot



- Hay que integrarlo con Apolo
- Inicialización:
  - Parámetros del filtro -> **calibrar odometría y sensores**
  - Trayectoria inicial -> **debería generarla un planificador *off-line***
- Bucle principal:
  - Fija comando de velocidad -> **debería generarlo un controlador**
  - Simula el avance real del robot -> **debe simularlo apolo**
  - Estima el avance del robot
  - Simula las medidas de los sensores -> **debe simularlas apolo**
  - Estima las medidas de los sensores -> **debe cambiarse el modelo**
  - Corrige la estimación de la posición del robot

- Importancia de los siguientes aspectos:
  - Elección de  $P(0|0)$ .
  - Calibración de las matrices  $Q(k)$  y  $R(k+1)$ .
  - Análisis de la influencia de estos parámetros en la estimación del estado  $P(k+1|k+1)$ .
  - Análisis de la influencia del mapa y de los sensores escogidos en la estimación del estado.
- Deberá cambiarse el modelo de observación dado en el ejemplo.

## 4. Requisitos de Entrega del Trabajo

---

# Entrega del Trabajo

- El trabajo se entregará en Moodle, como muy tarde el día anterior a la fecha programada para el examen final de enero.
- Adicionalmente a la entrega en formato electrónico, y de forma ocasional, algún grupo podría ser requerido a realizar una presentación presencial ante los profesores de la asignatura para aclarar aspectos concretos del trabajo.

# Entrega del Trabajo

- La entrega constará de tres partes que se entregarán en un único fichero en formato comprimido ZIP o RAR:
  - **Parte 1:** ficheros con todo el código desarrollado en MATLAB y definición del entorno en formato xml. La ejecución de este código se corresponderá con un demostrador que integre todos los componentes desarrollados.
  - **Parte 2:** un fichero en formato pdf con la documentación del trabajo desarrollado.
  - **Parte 3.** Un vídeo demostrativo en el que se aprecie la integración de los aspectos principales desarrollados en el trabajo (localización, control y planificación).

# 5. Documentación



- Deberá ser breve pero concisa (se recomienda no exceder 20 páginas), debiendo contener al menos la **descripción y justificación** de los siguientes aspectos:
  - La aplicación escogida.
  - Las pruebas de **calibración** realizadas para determinar la incertidumbre del sistema de locomoción.
  - Los sensores escogidos, de la utilización que se ha hecho de ellos en el trabajo, y resultados de la **calibración** de los mismos.
  - Descripción del algoritmo de **localización** programado (modelos cinemático y de observación y matrices Jacobianas  $\Phi(k)$ ,  $H(k)$ ), así como de los experimentos llevados a cabo para el ajuste de sus parámetros principales ( $P(0|0)$ ,  $Q(k)$ ,  $R(k)$ ).

- Descripción del algoritmo de **control** reactivo programado y de los experimentos realizados para su ajuste.
- Descripción del algoritmo de **planificación** de trayectorias utilizado y de las pruebas llevadas a cabo.
- Breve descripción del demostrador final integrando todos los componentes anteriores.
- Es imprescindible que se indique en la documentación el **reparto de roles** entre los miembros del equipo de trabajo.



## 6. Evaluación



- Se valorarán los siguientes aspectos:
  - Originalidad tanto en el problema a resolver como en la solución propuesta.
  - Corrección del planteamiento del problema y de la solución.
  - Calidad de las pruebas y experimentos llevados a cabo.
  - Claridad y completitud de la documentación aportada.
  - Adecuación del reparto de roles.
- La ponderación será la siguiente:
  - Control reactivo (20%)
  - Localización y calibración (50%)
  - Planificación (20%)
  - Reparto de roles (10%)

- Cada una de estas calificaciones podrá aplicarse de manera colectiva o individual a los miembros del equipo.
- Los trabajos deberán ser originales, no admitiéndose compartir código entre los equipos.

# Guiado y Navegación de Robots: Práctica 1

---

Fernando Matía