

Navindoor: Software de simulación para el algoritmos localización

Jesus Oroya
DeustoTech Mobility
Universidad de Deusto, Bilbao
Email: djoroya@deusto.es

Resumen—En este artículo se presentará la herramienta Navindoor. Esta es un paquete para MATLAB que permite el desarrollo de algoritmos de posicionamiento en interiores, así como la simulación de la señales asociadas al movimiento del objetivo, como puede ser las señales de radio frecuencia, señales inerciales, norte magnético, presión atmosférica, entre otras. Navindoor contiene la implementación de filtros de Kalman con capacidad de contener nuevos algoritmos de todo tipo. De esta forma se reduce el tiempo de desarrollo en la investigación de posicionamiento.

Index Terms—localización en interiores, procesamiento de señales

I. INTRODUCCIÓN

Las metodologías utilizadas para la sistemas de localización son tan variadas como la diversidad de sensores existentes. Sin embargo, aunque los algoritmos sean diversos, existe una serie de pasos comunes que se deben realizar antes que desarrollar un algoritmo de posicionamiento. Estas son, la toma de medidas experimentales de las señales, así como medidas de la propia trayectoria. Esta recolección de datos debe ser variada, ya que la colección en pocas casuísticas puede agregar sesgos no deseados a nuestros algoritmos. Es por ello que lo ideal es realizar pruebas en varios entornos con distintas trayectorias. Sin embargo, esto hace que el tiempo de desarrollo se alargue.

Navindoor propone una solución a este problema mediante la simulación del proceso de posicionamiento. El escenario, la trayectorias y las señales son simulados mediante modelos correspondientes. Además de permitir al usuario crear nuevos modelos si así lo quisiese. De esta forma Navindoor se convierte en un simulador muy versátil.

Navindoor está diseñado para poder desarrollar tres tipos de algoritmos.

1. **Algoritmos de simulación de trayectorias:** Los modelos de movimiento de un objetivo deberá reproducir las velocidades y aceleraciones. Este modelo puede cambiar si el objetivo es un robot o una peatón, es más el modelo de movimiento puede cambiar según el escenario.
2. **Algoritmos de simulación de señales:** En Navindoor se contemplan dos tipos de señales, estas son las que dependen de balizas para ser generadas y las que son solo dependientes de la trayectoria seguida (Sección 1). Navindoor contiene modelos señales RSS, ToF, AoA,

Barometer, Magnetometer, etc. que pueden ser modificados o creado otros a partir de ellos.

3. **Algoritmos de posicionamiento:** Algoritmos de posicionamiento ya conocidos como los filtros de Kalman ya están implementados en Navindoor por defecto. Y de la misma forma que en los dos puntos anteriores, la implementación de nuevos algoritmos es muy fácil.

Gracias a que la herramienta contiene modelos por defecto en cada uno de sus frentes, podemos desarrollar de forma independiente a los demás modelos.

En este artículo, describiremos el estado del arte en materia de simuladores, además de la arquitectura de software debajo de navindoor, la funcionalidad del software en algunos ejemplos concretos y por último la dirección de los futuros desarrollos.

II. ESTADO DEL ARTE

En la comunidad del posicionamiento se está buscando una solución para estandarizar los desarrollos. Es por ello que se han realizado trabajos relacionados. En este apartado mencionaremos algunos trabajos y sus principales características.

II-A. SMILe

SMILe [1] es un paquete escrito C++ y Python, dedicado a la simulación de señales ToF (Time of Flight) y a su procesamiento. Este paquete soluciona la parte de definición de escenario. En SMILe se puede definir las posiciones de los puntos de acceso y colocar paredes para que luego un modelo sofisticado pueda simular las señales recibidas por un objeto en movimiento. Por otro lado, el procesamiento de las señales está escrito en python. SMILe se creó debido a la falta de herramientas de simulación de ToF, por lo que por ahora no contiene otros tipos de señales, sin embargo se encuentra entre sus próximos desarrollos.

II-B. PyLayers

PyLayers [2] es un software de simulación centrado en la simulación de RSS (Received Signal Strength), contiene modelos de señales muy realistas

II-C. Sensor Fusion and Tracking Toolbox

1

¹<https://es.mathworks.com/help/fusion/>

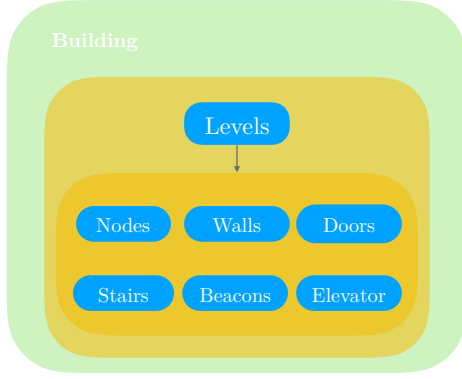


Figura 1. Visualización de la Facultad de ingeniería

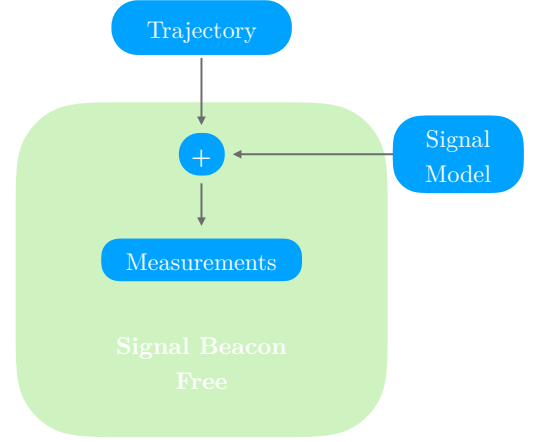


Figura 3. Visualización de la Facultad de ingeniería

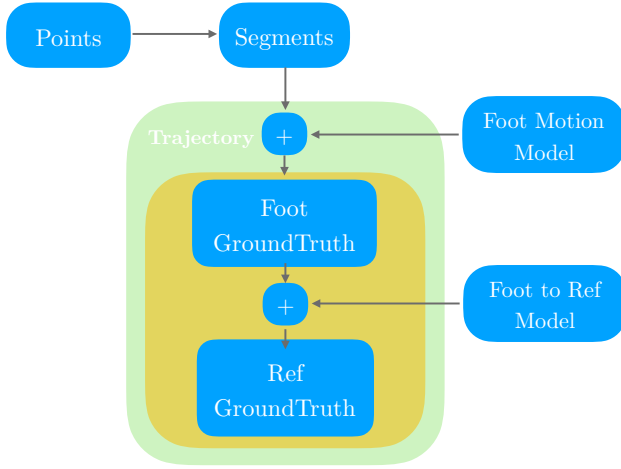


Figura 2. Visualización de la Facultad de ingeniería

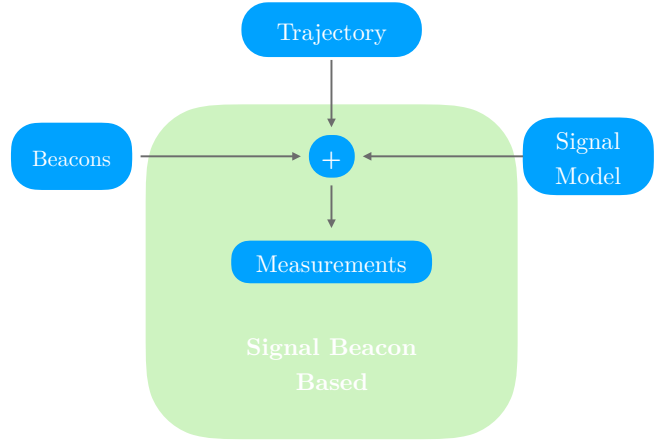


Figura 4. Visualización de la Facultad de ingeniería

III. ARQUITECTURA DE SOFTWARE

El desarrollo de Navindoor se ha dividido en módulos claramente diferenciados, que siguen los pasos para el posicionamiento mencionado en la introducción. A continuación describiremos cada una de ellas.

III-A. Planimetría

III-B. Trayectorias

III-C. Señales

Para la representación de las señales se ha creado un objeto MATLAB ...

III-D. Procesamiento

III-E. Comparación

IV. SIMULACIÓN

Veremos una ejemplo para ver la capacidades de navindoor en un ejemplo real. Para ello previamente, hemos creado

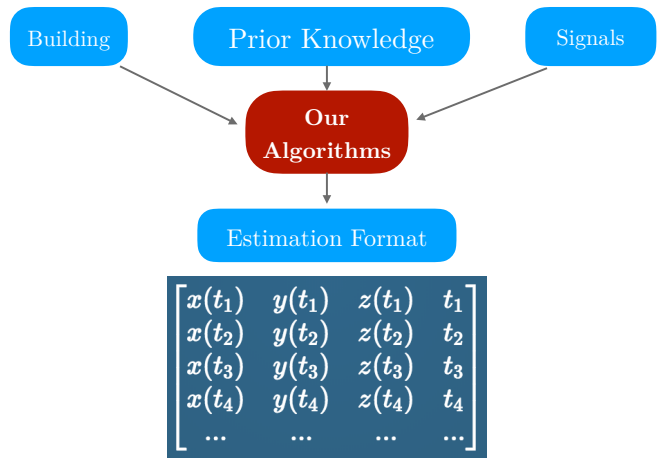


Figura 5. Visualización de la Facultad de ingeniería

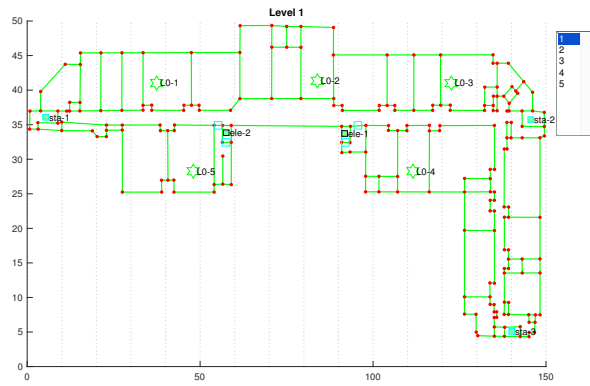


Figura 6. Visualización de la Facultad de ingeniería

la planimetría del edificio de la facultad de ingeniería de Deusto, con ayuda de la interfaz gráfica *navindoor*. Un vez creada podemos guardar esta en un archivo *.mat* y utilizar las distintas funciones implementadas para la clase edificio. Un ejemplo muy útil es utilizar la función *plot* sobre el objeto creado. Obtendremos la figura IV

V. CONCLUSIÓN

- Contiene todos sus módulos en el mismo lenguaje de programación, lo que hace que sea fácilmente instalable y mejora su mantenimiento
- Aunque todos sus módulos están en MATLAB puede contener simuladores programados en otros lenguajes gracias a los MATLAB APIs.
- La navegación en exteriores será desarrollada mediante la importación de mapas desde la plataforma *Open Street Maps*
- NAVINDOOR mezcla la experiencia GUI (interfaz gráfica) y CLI (línea de comandos) en el desarrollo de los algoritmos, lo que hace que la curva de aprendizaje sea menor.

REFERENCIAS

- [1] Jankowski T, Nikodem M. SMILe – Simulator for Methods of Indoor Localization. 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN). 2018;(September):24–27.
- [2] Amiot N, Laaraiedh M, Uguen B. PyLayers: An open source dynamic simulator for indoor propagation and localization. 2013 IEEE International Conference on Communications Workshops, ICC 2013. 2013;(July 2014):84–88.