

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. Es un framework 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

Gracias al desarrollo de IoT que se está viendo acelerado estos últimos años, los algoritmos de localización están experimentando un nuevo interés en el mundo tecnológico. Esto se debe a que nos estamos volviendo una máquina de recolección de señales, lo que hace que los métodos de localización sean tan variados como ecosistemas de sensores.

Aunque los algoritmos sean diversos, existe una serie de pasos comunes que se deben realizar antes que desarrollar un algoritmo de posicionamiento. Como puede ser la toma de medidas experimentales de las señales con la que se estimará la trayectoria, así como medidas de la propia trayectoria. Esta recolección de datos debe ser variada, ya que la colección en pocas casuística puede agregar sesgos no deseados a nuestros algoritmos. Es por ello que lo ideal es realizar pruebas en varios entornos con distintas trayectorias. Esto hace que el tiempo de desarrollo se alargue.

Navindoor soluciona este problema mediante la simulación del proceso de posicionamiento. El escenario, la trayectoria 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, ya que dentro de ella se puede desarrollar tres tipos de algoritmos: [1]

1. **Algoritmos de simulación de trayectorias:** Los modelos de movimiento de un objetivo deberán 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 de señales *RSS*, *ToF*, *AoA*, *Barometer*, *Magnetometer*, etc. que pueden ser modificados o creados 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. TRABAJOS RELACIONADOS

III. ARQUITECTURA DE SOFTWARE

El desarrollo de Navindoor se ha dividido en módulos claramente diferenciados. Estos son:

1. Generación de planimetría
2. Generación de trayectorias
3. Generación de señales
4. Procesamiento de señales
5. Comparación de métodos

Dentro de cada uno de estos módulos se encuentran definidas clases de MATLAB que representan su versión en la realidad. Un ejemplo muy claro está en los objetos de la planimetría. En este módulo existen objetos que representan paredes, puertas, escaleras, etc. La idea de esto es crear una interfaz intuitiva para que el usuario correspondiente aprenda a utilizar Navindoor de forma progresiva.

III-A. *Planimetría*

III-B. *trayectorias*

III-C. *señales*

III-D. *Procesamiento*

III-E. *Comparación*

IV. EJEMPLO

V. CONCLUSIÓN

The conclusion goes here.

REFERENCIAS

- [1] Correa A, Barcelo M, Morell A, Vicario JL. A review of pedestrian indoor positioning systems for mass market applications. *Sensors (Switzerland)*. 2017;17(8).