|  |  |  |
| --- | --- | --- |
| **Módulo 3 – Tema 3** | Daniela Rus |  |

**Video 1:**

IoT tiene el potencial de despertar el 99% de dispositivos.

2020 => 50 billones de dispositivos => cada persona mínimo 7 conectados a Internet.

Los autos serán los nodos más importantes en IoT.

Todos estos dispositivos deben ubicarse unos a otros instantáneamente. Tendrán identificación y sentido del mundo que los rodea.

**Video 2**

Dado una representación del entorno (marco de referencia) el dispositivo debe, a través de detección, computar su posición y dirección.

**Pose** => Hay que:

* Determinar la distancia desde el origen del sistema de referencia y el punto (ej: x, y).
* Vector de dirección del objeto.

Estimación de Pose de lazo abierto (dispositivos basados en ruedas):

* No hay detección.
* Odometría: cnt del nro. de rotaciones de la rueda.
* Llena de errores:
* Las ruedas resbalan.
* Error de cómputo.
* Arrastre de errores.
* Simple pero no es lo suficientemente buena.

**Video 3**

Detección en 2D

* Detectar distancias. Basada en rangos. Sonares o sensores infrarrojos. Rango a punto potencia señal recibida, fuerza wifi, tiempo de vuelo de la acústica o balizas RF. Tambien por la diferencia en tiempo de llegada de 2 señales, por ejempo: acústica y radio.
* Detectar ángulos. Basada en rodamientos. Brújula o visión por computador.
* Rango y rodamiento relativo de un cuerpo a un objeto: radar, escáner láser.

**Video 4**

**Triangulación: enfoque más común (natural) para localización 2D:**

Se mide el rango a un punto de ref. y rodamiento relativo al punto de ref.

Posición:

* P: que no lo sabemos.
* L1: punto fijo. Medir distancia d1 => P está dentro de un círculo con centro en L1 (radio d1).
* L2: punto fijo. Medir distancia d2 => P está dentro de un círculo con centro en L2 (radio d2).
* P está en la intersección de ambos círculos.
* Se cambia L1 como inicio del sistema de coordenadas (0, 0) y L2 por sobre eje x (a, 0).
* X= (a1 + d12 – d22) / 2ª.
* Y= +- raíz (d12 – x2).
* Utilizar la transformación del sistema global – local de coordenadas para calcular las coordenadas de P.

El tema es que calculamos P, pero es lo mismo que P’ (simétrico del otro lado del plano).

* Podemos utilizar un mapa y trackear como se va moviendo.
* También guardando la historia.
* Agregar otro punto en el cómputo

Dirección (si no está en movimiento):

Podemos utilizar cómputo de otro punto y diferencia entre esas 2 posiciones.

* Mediante medida de ángulos de ref. con 2 puntos L1 y L1 (@1 y @2). No alcanza porque la dirección se puede dar en cualquiera de los puinto que forma la circunferencia formada por L1 y L2.
* Sumar otro punto L2 y el ángulo.

**TODO ESTO SI PODEMOS MEDIR LAS 3 DISTANCIA Y 3 ANGULOS.**

En el mundo real los sensores tendrán ruido => **Incertidumbre!!!**

Generalmente los sensores de distancias traen como parte del producto, el error máximo por lo que podemos pulir los cálculos.

Distancia +error:

* Círculo con precisión.
* Circulo máximo error.
* Se puede calcular la región donde probablemente esté

**SE TRABAJA CON REGIONES. IDEM PARA ANGULOS DE DIRECCION.**

**Video 5**

Algoritmo que garantiza una solución robusta de ubicación.

* Probabilísticamente robusto con respecto al ruido y no necesita de los 3 puntos como los anteriores.
* No necesita balizas como L1, L2 y L3.
* Funciona aún si los nodos se mueven.

Asumimos que los nodos se comunican entre sí y pueden medir la distancia relativa entre ellos.

Problemas con el ruido:

* Un error menor puede transformarse en un enorme problema en la topología de nodos. PROBLEMA PRNCIPAL. Llamado **cambio de ambigüedad**.

Base teórica el algoritmo que llamamos “**Cuadrilátero robusto**”.

* Características: rigidez (sin deformaciones continuas), sin ambigüedades de flexión continuas, probabilísticamente constreñir el cuadrilátero para minimizar la probabilidad de cambio de ambigüedad.

**Trilateración para garantizar robustez.**

Video ejemplo de utilización del algoritmo en robots y nodos crickets.

External node con otro método de ubicación para los nodos no ubicados luego de utilizar quads robustos.

**Video 6**

Ubicación en **ambientes** **externos**: GPS nos da una ubicación regular. Puede introducir mucho error. Dentro ni siquiera tenemos GPS.

iMiEv: vehículo autónomo eléctrico :

* LIDAR hacia abajo para detectar la localización.
* LIDAR planar: reconocimiento de objetos.
* Webcam para varios propósitos.
* Sensor IMU odometría.
* Necesita un mapa.
* Localización algoritmo Montecarlo. MCL.
* **El error máximo ocurre en distancias largas donde no hay características particulares del entorno.**

**Video 7**

Limitaciones del algoritmo anterior:

* Asume que las características de intersección existen.
* Incertidumbre segmentos sin características. Rectos.

Que hacer:

* Usar más características verticales. Árboles, edificios, cerco.
* LIDAR sintético: módulo percepción 3D y localización 2D.

**Video 8**

**Localizaciones interiores**. Es más complicado obtener las características dentro que afuera.

No hay una solución ideal.

Todas las que hay necesitan:

* Mucha infraestructura.
* Tiempo de config.
* Hard. específico.
* Más cálculos.

Técnica SLAM.

Nueva manera de calcular:

* Señales wifi intrínsicas.
* Utilización de dispositivo cel. Por ejemplo.
* Si tuviera varias antenas + 2 subestaciones => Integrar la info.
* No existe tal dispositivo con muchas antenas.

Que hacer:

* Emular **array de antena**.
* **Moviendo** el dispositivo podemos emular el arreglo de antenas.
* Necesitamos la ubicación de varias estaciones del entorno.
* Ejemplo de localización de un libro. Geo etiquetar objetos.

**Video 9**

CIERRE.