

# TSRMI: Assignment #15

Luis Alberto Ballado Aradias  
luis.ballado@cinvestav.mx

CINVESTAV UNIDAD TAMAULIPAS — 4 de julio de 2023

- Documentar el pseudocódigo de los algoritmos Bug1 y Bug2.

---

**Algorithm 1:** Algoritmo Bug1

---

```
Definir punto objetivo  $P$ ;  
Inicializar posición actual del robot  $R$ ;  
Inicializar posición variable hit como false;  
while  $R \notin P$  do  
    if  $R$  libre de obstáculos en la dirección hacia  $P$  then  
        Mover  $R$  hacia  $\Rightarrow P$ ;  
    else  
        if No se ha encontrado obstáculo anteriormente ( $hit=false$ ) then  
            Girar  $R$  en sentido horario;  
            Marcar posición actual como punto de contacto;  
            Establecer hit como verdadero  
        else  
            if Si  $R$  ha vuelto al punto de contacto inicial then  
                Mover  $R$  hacia adelante hasta superar el punto de contacto  
            else  
                Girar  $R$  en sentido horario  
            end  
        end  
    end  
end  
end
```

---

---

**Algorithm 2:** Algoritmo Bug2

---

```
Definir punto objetivo P;  
Inicializar posición actual del robot R;  
Inicializar posición variable hit como falso;  
while  $R \notin P$  do  
  if R está libre de obstáculos en la dirección hacia P then  
    Mover R hacia  $\Rightarrow P$ ;  
  else  
    if Si no se ha encontrado obstáculo anteriormente (hit=false) then  
      Girar R en sentido horario;  
      Marcar posición actual como punto de contacto;  
      Establecer hit como verdadero;  
    else  
      if Si R ha vuelto al punto de contacto inicial then  
        Calcular la distancia a P desde la posición actual;  
        if Distancia a P desde el punto de contacto then  
          Girar R en sentido horario;  
        else  
          Mover R hacia adelante hasta superar el punto;  
        end  
      else  
        Girar R en sentido horario;  
      end  
    end  
  end  
end
```

---

- Una tercera variante, denominada TangentBug, utiliza sensores de distancia en lugar de sensores de contacto. Documentar un artículo científico que emplee el algoritmo TangentBug.

El algoritmo TangentBug es una variante del algoritmo Bug utilizado para la navegación de robots móviles en entornos desconocidos o con obstáculos. A diferencia de los algoritmos Bug1 y Bug2, el algoritmo TangentBug utiliza sensores de distancia en lugar de sensores de contacto para detectar los obstáculos.

En el algoritmo TangentBug, el robot utiliza sus sensores de distancia para medir la distancia a los obstáculos en su entorno. El objetivo principal del algoritmo es seguir la tangente de los obstáculos para alcanzar el objetivo deseado. A medida que el robot se acerca a un obstáculo, mide la distancia a ese obstáculo en varios puntos alrededor de su posición. Luego, elige el punto en el que la distancia al obstáculo es mínima y se dirige en la dirección perpendicular a ese punto, siguiendo la tangente del obstáculo. Este proceso se repite iterativamente mientras el robot se mueve hacia el objetivo.

La idea central del algoritmo TangentBug es utilizar la información proporcionada por los sensores de distancia para tomar decisiones de movimiento basadas en la geometría del entorno y la ubicación relativa del objetivo y los obstáculos. Esto permite que el robot navegue de manera eficiente y evite colisiones al seguir la tangente de los obstáculos en lugar de intentar rodearlos completamente.

**A Simple Local Path Planning Algorithm for Autonomous Mobile Robots.** El artículo propone la aplicación del algoritmo TangentBug para la navegación de robots móviles autónomos. Describe cómo el algoritmo utiliza sensores para seguir el perfil de un obstáculo y encontrar una ruta alrededor de él.

- ¿Podría describir, en forma de pseudocódigo, el algoritmo necesario para seguir el perfil de un muro empleando sensores de distancia?

---

**Algorithm 3:** Algoritmo para seguir el perfil de un muro

---

Inicializar variables y configurar los sensores de distancia;

Obtener la lectura de los sensores de distancia y almacenarla en una variable *distancia*;

**while** *No se haya alcanzado el objetivo* **do**

**if** *la distancia es menor que el umbral predefinido* **then**

        | Girar hacia la derecha o la izquierda;

**end**

**if** *Si la distancia es mayor que el umbral predefinido* **then**

        | Mover hacia adelante durante un tiempo predefinido para acercarse al muro;

**end**

    Actualizar la lectura de los sensores de distancia;

**end**

---