

Laboratorio de Robots Móviles Practica No. 1

Instalación de Herramientas de Software Comportamientos Reactivos

Objetivo:

* Familiarizar al alumno con la configuración del sistema operativo Linux versión Xubuntu, así como las herramientas de software utilizadas en la primera parte del curso.

*Familiarizar al alumno con los comportamientos reactivos

Duración: Dos semanas

1.- Instale el sistema operativo Linux versión Xubuntu en una computadora.

2.- Descargue de la siguiente página, el archivo denominado robotics, el cual contiene material de apoyo, código en C++ y Python:

<http://biorobotics.fi-p.unam.mx/robotics-courses/robots-moviles>

Recompile el código de C++/C que controla la operación de un robot simulado de la siguiente forma:

2.1.- Primero colocarse en el directorio en donde se encuentran parte de los archivos fuente:

```
$ cd ~/robotics/motion_planner
```

2.2.- Recompile el archivo GoTo_State_Machine.cpp con:

```
$ make -f Makefile_GoTo_State_Machine
```

2.3.- En una terminal ejecute la interface gráfica para visualizar la operación del robot utilizando Python. Colocarse primero en el directorio en donde se encuentra este software:

```
$ cd ~/robotics/gui
```

Después ejecute python con el código de la interface gráfica:

```
$ python python GUI_robots.py
```

Esta interfaz gráfica muestra el resultado de la simulación del código `GoTo_State_Machine.cpp`. Seleccione con el botón izquierdo del mouse la posición inicial del robot y con el derecho la final, después seleccione «Execute Robot Command». Observe el comportamiento del robot. Seleccionando de nuevo el botón derecho del mouse el robot tomará su última posición como origen y el destino la posición del botón derecho. Las coordenadas (0,0) del mapa se encuentran en el lado inferior izquierdo de la figura.

Familiarícese con el funcionamiento de la interfaz gráfica con diferentes configuraciones y comportamientos ejecutados por `GoTo_State_Machine`.

En el apéndice A se muestra el mapa simbólico en donde el robot navega, los objetos se representan con polígonos. Este tipo de archivos tienen terminación `*.wrl`.

En este apéndice se muestra las siguientes directivas:

1. “dimensions” indica las dimensiones del medio ambiente, las cuales están indicadas en el sistema métrico decimal. En el siguiente ejemplo se presentan las dimensiones del medio ambiente “room” de 1m x 1m:

```
( dimensions room 1.000 1.000 )
```

2. “polygon” indica los vértices de un polígono, se indica si es de tipo obstáculo o pared, después viene el nombre del obstáculo y a continuación los vértices. Los vértices de los polígonos se indican con las coordenadas X_i y Y_i , ordenados hacia el sentido horario de las manecillas del reloj:

```
( polygon obstacle obs1 .40 .55 .60 .55 .60 .35 .40 .35 )
```

```
( polygon wall wall1 0.0 0.0 0.0 1.0 0.01 1.0 0.01 0.0 )
```

3. Se pueden incluir comentarios o comentar una línea si se coloca un “;” al principio del renglón.
Ejemplo:

```
; * File: room.wrl *
```

En el apéndice B se muestra una parte del archivo de datos generado por la simulación, es decir cuando se ejecuta `GoTo_State_Machine`. Este archivo de datos es utilizado por la interfaz gráfica para mostrar su resultado, este tipo de archivos tienen terminación `*.raw`

En este apéndice se muestra las siguientes directivas:

1. “radio_robot” indica el radio del robot, en el ejemplo siguiente se indica que el radio del robot es de 3 cm:

```
( radio_robot 0.030000 )
```

2. “robot” indica la posición del robot, que en este ejemplo se llama Justina, con la posición X, Y y el ángulo del robot con respecto al eje x, el cual se encuentra en la parte inferior de la figura.

```
( robot Justina 0.080000 0.332500 0.000000 )
```

3. “sensor” indica el tipo de sensor utilizado, así como el número de sensores, el rango angular del sensor, la posición de la primera lectura con respecto al centro del robot, después vienen los valores obtenidos por los sensores.

En el siguiente ejemplo se obtuvieron las lecturas de 2 sensores láser, con un rango angular de 0.400 radianes, posición inicial -0.2000 (lado derecho del robot) y las lecturas de 30 y 28.5697 centímetros:

(sensor laser 2 0.400000 -0.200000 0.300000 0.285697)

Los archivos *.wrl y *.raw se encuentran en el directorio ~/robotics/data

El tamaño del robot, números de sensores, tipo de sensores, posición de estos, etc, se incluyen como parámetros cuando se ejecuta el programa GoTo_State_Machine y los cuales son colocados por la interfaz gráfica.

3.- En la figura 1 se muestra el comportamiento de un robot que evade obstáculos.

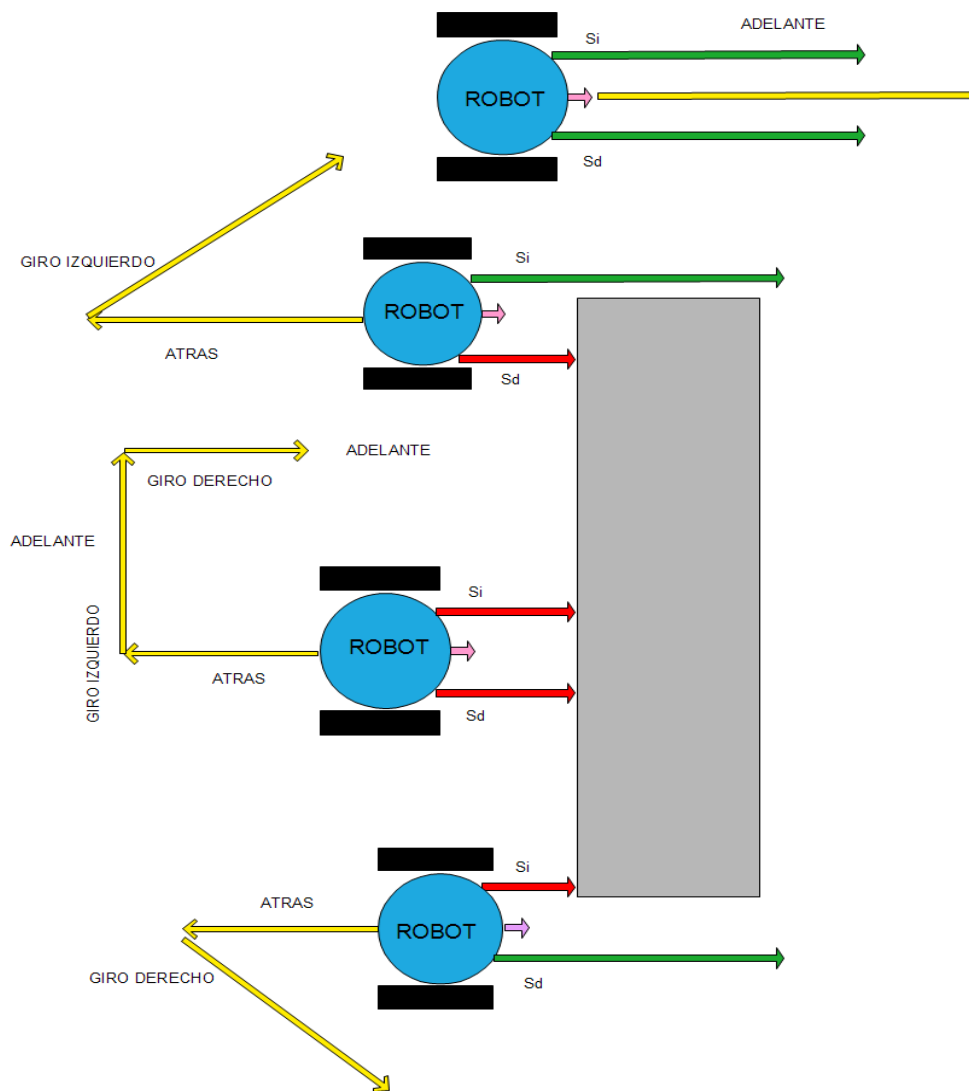


Figura 1. Robot Móvil que evade obstáculos

Este algoritmo, usando el lenguaje de programación C++/C, se encuentra en el archivo `~/robotics/state_machines/state_machine_avoidance.h`, el cual es incluido en el código de `GoTo_State_Machine.cpp`. Entienda el funcionamiento de este código, también el de los algoritmos que mueve el robot al destino, y el que combina estos dos comportamientos, los cuales se encuentran en `state_machine_destination.h` y `state_machine_avoidance_destination.h` respectivamente.

Pruebe estos comportamientos seleccionando en «Behaviour Selection» con un 1 el comportamiento que evade obstáculos; con un 2 el comportamiento que mueve al robot al destino y con un 3 la combinación ambos. Pruebe estos algoritmos con diferentes configuraciones modificando el número de sensores, rango angular, posición inicial, magnitud de avance y giro del robot, valor máximo de lectura sensorial, etc.

Reporte bajo que condiciones no funcionan apropiadamente estos algoritmos.

4. Modifique el código de `GoTo_State_Machine.cpp` para que se tenga el mismo comportamiento que `state_machine_avoidance_destination.h` utilizando los comportamientos separados que se encuentran en `state_machine_avoidance.h` y `state_machine_destination.h` utilizando un arbitro.
5. Haga un comportamiento para que el robot encuentre un obstáculo y lo rodee completamente.
6. Haga un comportamiento para que el robot siga paredes.

APENDICE A
Mapa Simbólico room.wrl

```
; *****  
; * File: room.wrl *  
; * Definition of the forbidden and allowed areas in the Robot's *  
; * world. These areas are derivated from the objects in the *  
; * world. *  
; *****  
  
( dimensions room 1.000 1.000 )  
  
( polygon obstacle obs1 .40 .55 .60 .55 .60 .35 .40 .35 )  
  
( polygon wall wall1 0.0 0.0 0.0 1.0 0.01 1.0 0.01 0.0 )  
  
( polygon wall wall2 0.0 0.99 0.0 1.0 1.0 1.0 1.0 0.99 )  
  
( polygon wall wall2 0.99 1.0 1.0 1.0 1.0 0.0 0.99 0.0 )  
  
( polygon wall wall2 1.0 0.01 1.0 0.0 0.0 0.0 0.0 0.01 )
```

APENDICE B
Parte del archivo room.raw generado por la simulación

```
( radio_robot 0.030000 )  
  
( robot Justina 0.080000 0.332500 0.000000 )  
  
( sensor laser 2 0.400000 -0.200000 0.300000 0.300000 )  
  
( robot Justina 0.120000 0.332500 0.000000 )  
  
( sensor laser 2 0.400000 -0.200000 0.300000 0.285697 )  
  
( robot Justina 0.160000 0.332500 0.000000 )  
  
( sensor laser 2 0.400000 -0.200000 0.300000 0.244883 )  
  
...  
  
...  
  
...
```