



Universidad Nacional Autónoma de México
Posgrado de Ingeniería
Laboratorio de Biorrobótica



Robot Móvil “Azcatl”

Servicio Social

Descripción de algoritmo
“Evasor de Obstáculos y Seguidor de Luz”

A cargo de:
Dr. Savage Carmona Jesús

Realiza:
Silva Guzmán Alejandro

9 de junio de 2021

Descripción general

Este documento presenta la descripción de la unión de los dos algoritmos reactivos realizados previamente: “Evasor de obstáculos” y “Seguidor de luz”. Una vez se realizaron las pruebas necesarias para determinar su buen funcionamiento se realizó este tercer algoritmo.

Este nuevo código no presenta nuevos parámetros ni variables, únicamente la unión de todos los elementos que conformaban a los algoritmos reactivos ya mencionados.

El nuevo código se describe a continuación.

Inserción de datos por el usuario:

```
378 printf( "\033[2J\033[1;1H" );
379 printf("\n----- Evasor de Obstaculos y Seguidor de Luz -----");
380 printf("\n----- START -----", step+1, total_steps);
381 printf("\n\nIntroduzca:\n\nEl numero de pasos deseado [entero]: ");
382 scanf("%d",&total_steps);
383 printf("Angulo a girar por paso [°]: ");
384 scanf("%f",&angulo_usuario);
385 printf("Distancia a recorrer por paso [cm]: ");
386 scanf("%f",&distancia_usuario);
387 printf("Distancia a detectar los obstaculos [cm]: ");
388 scanf("%f",&umbral_usuario);
389 printf("Ancho del obstaculo mas delgado a detectar [cm]: ");
390 scanf("%f",&obstacle_Width);
391 printf("Intensidad a detectar de luz:\n\t\t1. Debil 30%c-50%c\n\t\t2. Media 50%c-70%c\n\t\t3. Alta 70%c-90%c\n");
392 scanf("%d",&intensity);
```

Se observa que hay una unificación en cuanto a todos los datos que requiere el algoritmo en esta sección que va de la línea 378 hasta la 392.

Recaltar que se mantiene la modalidad de “ancho deseado” agregada ya en el algoritmo “Evasor de Obstáculos” como también la modalidad de “intensidad de luz deseada” vista en el algoritmo “Seguidor de Luz”.

Booleanos de conexión:

```
403 //-----Loop para seguir recibiendo datos de Hokuyo y Arduino-----/
404
405 photosensors_flag = false;
406 printf("\n\n--->Esperando a Arduino.");
407 while(!photosensors_flag && ros::ok()){
408     printf(".."); //Imprimira ".." mientras espera respuesta de Arduino(Foto
409     ros::spinOnce();
410     rate.sleep();
411 }
412
413 hokuyoFlag = false; //Bandera para detectar la conexion con el hokuyo
414 printf("\n\n--->Esperando a Hokuyo.");
415 while(!hokuyoFlag && ros::ok()){
416     printf(".."); //Imprimira ".." mientras espera respuesta de Hokuyo
417     ros::spinOnce();
418     rate.sleep();
419 }
```

Se agregan ahora dos métodos iterativos para detectar la conexión con los dos controladores del Ázcatl: Arduino y Hokuyo. En ellos se emplean dos variables booleanas para detectar la conexión efectiva con estos componentes.

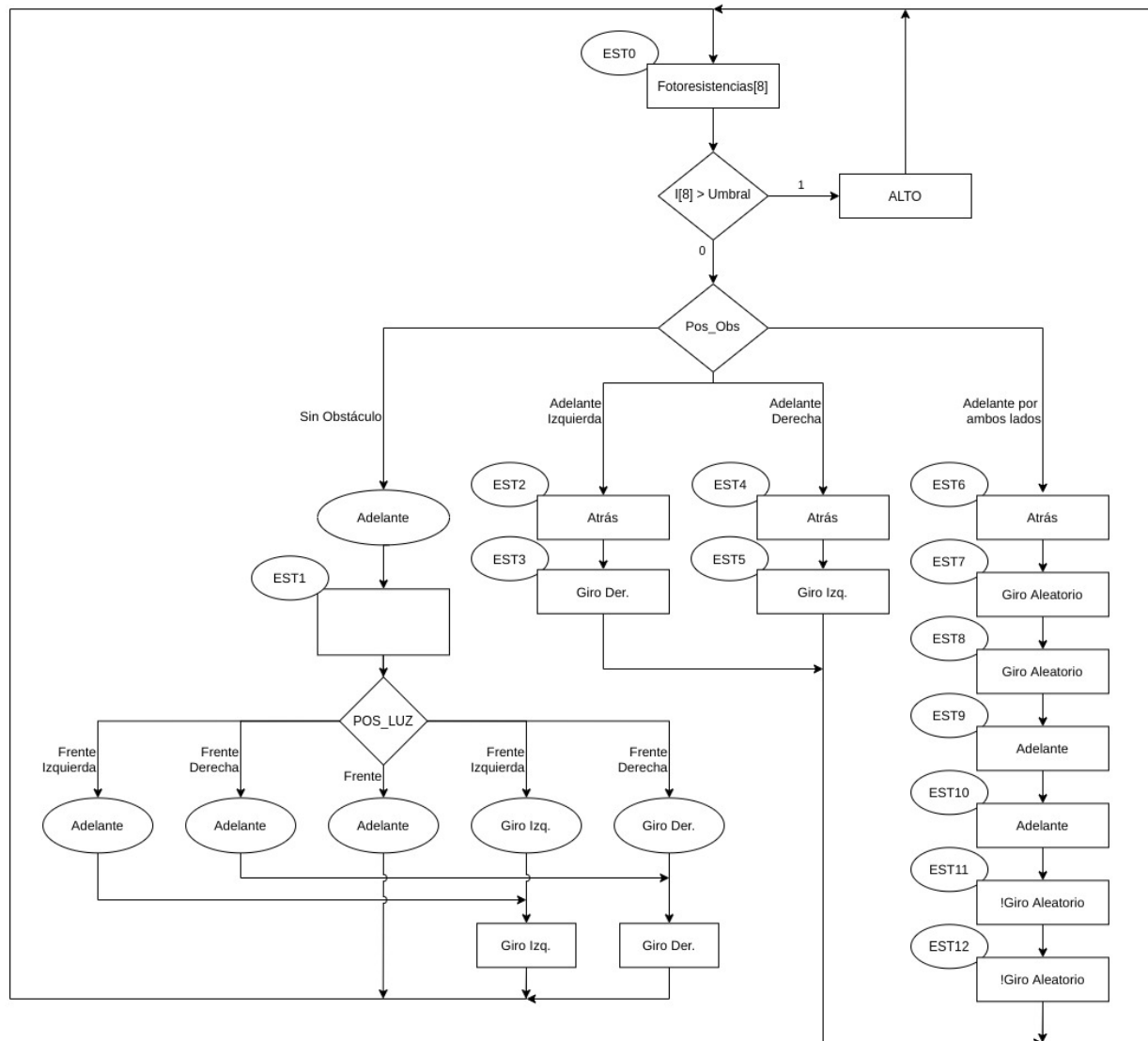
Dichos dos ciclos utilizan las siguientes funciones:

Función	Línea	Controlador
callbackPhotosensors()	58	Arduino
data_photosensors()	141	Arduino
callbackHokuyo()	216	Hokuyo
data_hokuyo()	334	Hokuyo

Mismas que trabajarán a la par con la nueva máquina de estados.

Máquina de estados: Evasor de obstáculos y seguidor de Luz

El nuevo algoritmo de describe con la siguiente carta ASM:



Carta ASM del Algoritmo Evasor de Obstáculos y Seguidor de Luz

Explicación Breve del algoritmo:

Se observa que la estructura básica de los dos algoritmos anteriormente usados se mantendrá, dándole prioridad al algoritmo *Evasor de Obstáculos*. Si y sólo si se encuentra el camino libre de obstáculos se procederá a buscar la fuente de luz y actuar en consecuencia para encontrarla, entrando en acción el algoritmo *Seguidor de Luz*.