

Universidad de la República, Facultad de Ingeniería

PROYECTO PEZ ROBOT

**DIEGO ASTESSIANO
PABLO ROMERO
BERARDI SENSALÉ**

TUTOR: RAFAEL CANETTI

Fecha de entrega: 15 de junio de 2008

Contenido

Introducción.....	5
Descripción del proyecto.....	5
Objetivo general del proyecto.....	5
Especificación general del proyecto.....	6
CAPITULO 1 Elección de los componentes e implementación.....	7
1 Elección del sistema de Sensores.....	8
1.1 Introducción.....	8
1.2 Sensores de Presión.....	8
1.3 Sensores de Infrarrojos.....	10
1.4 Sensores de ultrasonido.....	12
1.5 Conclusión.....	14
2 Ensayo de los sensores.....	15
2.1 Ensayo de los sensores de Presión.....	15
2.1.1 Elección de la electrónica asociada.....	15
2.1.2 Descripción del ensayo.....	16
2.1.3 Análisis de los resultados.....	17
2.1.4 Conclusión.....	18
2.2 Ensayo de los sensores de distancia.....	19
2.2.1 Elección de la electrónica asociada.....	19
2.2.2 Descripción del ensayo.....	19
2.2.3 Análisis de los resultados.....	20
2.2.4 Conclusión.....	23
3 Sistema de Control de Profundidad.....	24
3.1 Conexiones.....	24
3.2 Funcionamiento.....	25
3.3 Modulación por ancho de Pulsos.....	25
3.4 En relación a los motores.....	26
3.5 Ensayos del servomotor.....	27
3.5.1 Elección de la electrónica asociada.....	27
3.5.2 Descripción del ensayo.....	27
3.6 Sistema de Control de Movimiento.....	28
4 Elección de microcontroladores a utilizar.....	29
4.1 Introducción.....	29
4.2 Opciones de arquitectura del sistema.....	29
4.3 Estudio de diversos microcontroladores y "Single Board Computers" para su posible utilización en el proyecto.....	31
4.4 Comparación de las distintas alternativas.....	33
4.5 Elección de arquitectura y microcontrolador.....	34
5 Modelo Matemático del Sistema.....	35
5.1 Resumen.....	35
5.2 Modelo Utilizado.....	35
5.3 Implementación en Matlab.....	37
5.4 Modelado computacional de los sensores.....	37
5.5 Primer intento de control: Rutina para seguir la pared.....	38
6 Comportamientos del Pez.....	45
6.1 Introducción.....	45
6.2 Movimientos Simples.....	45

6.3.	Atributos.....	45
6.4.	Tareas.....	45
6.5.	Sensores activos en distintas tareas.....	46
6.6.	Modos.....	48
6.6.1.	Primer enfoque.....	48
6.6.2.	Segundo enfoque.....	49
6.6.3.	Comparación de los enfoques.....	50
6.7.	Plan de distinción de los estados de ánimo.....	50
6.8.	Análisis Temporal.....	50
6.9.	Asignación de probabilidades de transición.....	51
6.10.	Programación.....	53
6.11.	El Programa principal.....	58
7.	Primera implementación.....	59
7.1.	Implementación.....	59
7.1.1.	Componentes utilizados.....	60
7.1.2.	Arquitectura del sistema.....	60
7.2.	Control.....	60
7.3.	Algunos resultados.....	62
7.4.	Principales inconvenientes y dificultades.....	62
7.5.	Conclusiones.....	62
7.6.	Divulgación.....	63
8.	Baterías.....	64
8.1.	Baterías Ni-Cd (Nickel Cadmium).....	64
8.2.	Baterías NiMH (Níquel Metal Hydride).....	65
8.3.	Baterías LiIon (Litio-Ión).....	66
8.4.	Comparativa de baterías recargables.....	67
8.5.	Precios.....	68
8.6.	Detalles de consumo.....	69
8.7.	Circuito propuesto para el cargador.....	70
9.	Diseño de la arquitectura de software y hardware.....	71
9.1.	La arquitectura del Hardware.....	71
9.2.	La Arquitectura del Software.....	72
9.2.1.	Descripción a nivel del usuario.....	73
9.2.2.	Lenguajes de implementación de los códigos de los programas.....	73
9.2.3.	Diagrama del flujo de la información a nivel de software:.....	73
9.2.4.	Detalle de los programas implementados.....	74
9.3.	Tareas realizadas para la implementación de estos programas y para iniciar la programación del Gumstix.....	74
10.	El Enlace Inalámbrico.....	76
10.1.	Introducción.....	76
10.2.	Propagación electromagnética en un conductor.....	76
10.3.	Propiedades del agua doméstica.....	77
10.4.	Terminales.....	78
10.5.	Estudio de Pérdidas de propagación.....	78
10.6.	Experimento.....	81
10.7.	Conclusiones.....	81
CAPITULO 2 Manual de Usuario.....		82
1.	Requisitos previos.....	83

2.	Software a instalar.....	83
3.	Inicialización de la Comunicación entre el Gumstix y el PC.....	84
4.	Inicialización de la Comunicación entre el Gumstix y el Robostix.....	86
5.	Inicialización del funcionamiento del Pez.....	87
6.	Descripción de los programas disponibles en el Gumstix.....	88
7.	Despliegue de información en tiempo real en la PC.....	89
CAPITULO 3	Implementación final y discusión de pruebas y resultados.....	91
1.	Introducción.....	92
2.	Implementación.....	92
3.	Ensayos.....	93
3.1.	Primeras pruebas.....	93
3.2.	Pruebas en agua.....	93
4.	Conclusiones.....	95
5.	Futuras aplicaciones y trabajo a futuro.....	95
BIBLIOGRAFIA.....		96

Introducción

Descripción del proyecto

Este proyecto consiste en diseñar un pez robótico autónomo que sea capaz de nadar en un medio desconocido o en un ambiente cambiante.

Este pez robótico tendrá las características de autonomía al navegar, aproximándonos al movimiento y las acciones mediante inspiración biológica. Se buscará imprimirle funciones básicas como la navegación rectilínea, evitar colisiones, y otras. Todas estas funciones serán implementadas basándonos en un sistema de sensado del medio y reacción a diferentes estímulos.

Para hacer que el comportamiento del pez sea similar al de un pez real, se implementará un mecanismo de propulsión que haga posible un movimiento similar a aquel de un pez en su estado natural.

Para trabajar en la implementación de estructuras de control que hagan posible que el pez robótico pueda navegar en ambientes desconocidos y cambiantes, como puede ser frente a la presencia de obstáculos este sensorará datos del medio, y reaccionará seleccionando aquel movimiento que sirva como la mejor acción frente al problema.

Respecto a los antecedentes del presente proyecto, los más importantes son el pez biológicamente inspirado desarrollado en el año 2006 a un costo de medio millón de dólares, en la universidad de Essex, Reino Unido, y actualmente en exhibición en el acuario de Londres [1]. También sirven como antecedentes, pero sin la cualidad de autonomía,, los robots desarrollados por el National Maritime Research Institute de Japón [2]. Luego también hay algunos peces robots con capacidades o autonomía limitadas, como por ejemplo los desarrollados en el MIT [3] y EPFL [4].

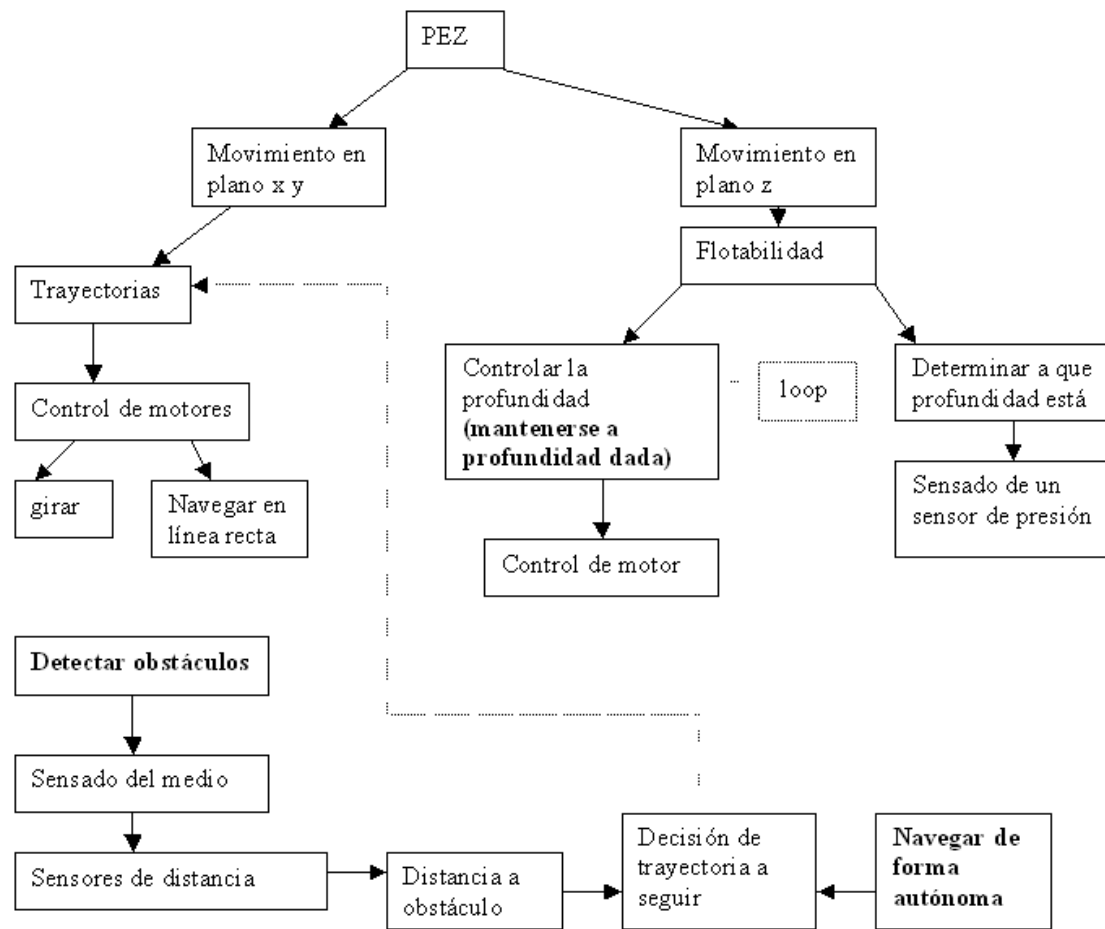
Objetivo general del proyecto

Exactamente lo que se realiza en este proyecto es diseñar el sistema eléctrico y de control de un pez robótico (el diseño de la estructura mecánica es por parte del tutor). Siendo este autónomo, capaz de navegar manteniendo una profundidad variable y de evitar obstáculos.

El qué del proyecto es tratar de simular conductas de los peces reales, imitando el comportamiento al nadar de los mismos (tanto en la forma de propulsión como en algunas conductas). Otra de las razones para los cuales es útil, es para estudiar a otros peces pues sirve para investigarlos sin invadir su hábitat natural, ya que un vehículo acuático con otros medios de propulsión (Ej. con hélice) puede ser muy ruidoso. Por último, también puede considerarse como una mascota electrónica.

Este proyecto será una plataforma inicial para futuras aplicaciones, como por ejemplo el estudio de la aplicación de sensores basados en la naturaleza.

Especificación general del proyecto



Se procurará la posibilidad de expansión de electrónica a nuestro pez, para implementar un sistema con sensores de electrolocation [5-8].

CAPITULO 1

Elección de los componentes e implementación

1. Elección del sistema de Sensores

1.1. Introducción

Según el diccionario de la real academia española, un sensor es un “dispositivo que detecta una determinada acción externa, temperatura, presión, etc., y la transmite adecuadamente”.

Una parte muy importante del diseño de robots es el estudio de su funcionamiento y de su interacción con el entorno. Para dotar a los robots de la capacidad para realizar pequeñas tareas de forma independiente es necesario dotarlas de sensores, que le transmitan información a partir de la cual puedan conocer su entorno y pueden llevar a cabo múltiples funciones.

En nuestro caso, dadas las características del entorno donde nuestro robot se encontrará, el cual es bajo el agua, este deberá moverse y ubicarse en el espacio (coordenadas x , y , z), con la restricción física de estar debajo de la superficie del agua. El movimiento en principio lo consideraremos independiente en el plano xy del eje vertical z , ya que se tratará de mantener constante la posición en esta última dirección.

El movimiento en z , será manejado sensando la profundidad a la que se encuentra el pez, esto será implementado con sensores que permitan determinar la distancia hasta la superficie del agua. Los sensores más adecuados para esta tarea aparentan ser los de presión, pues la presión del agua encima del mismo es directamente proporcional a la profundidad en que se encuentre.

Un requisito inicial para lograr autonomía sería la posibilidad de controlar la distancia a obstáculos en el plano xy , evitando así colisiones. Esto puede ser realizado con sensores de ultrasonido o bien de infrarrojos y a partir de las lecturas de estos decidir cómo continuar el movimiento.

1.2. Sensores de Presión

Los sensores de presión son dispositivos que miden la presión, tímicontroladoramente de gases o fluidos, siendo la presión una magnitud de la fuerza requerida para detener la expansión de un gas o de un fluido, que usualmente se expresa en términos de fuerza por unidad de área.

Los sensores de presión con los que trabajaremos, dadas las características de la aplicación, generan una señal eléctrica relacionada a la presión impuesta. Estos usualmente están compuestos de membranas delgadas que se flexionan debido a la presión [9].

Estos sensores de presión se pueden clasificar en términos de los rangos de presión de operación y del tipo de presión que miden. Los segundos a su vez se pueden clasificar básicamente en tres categorías:

- Sensores de presión absoluta
Los cuales miden presiones relativas a la presión del vacío perfecto (0 kPa)
- Sensores de presión relativa
Los cuales miden presiones relativas a la presión atmosférica, la cual es de aproximadamente 100kPa al nivel del mar.
- Sensores de presión diferencial
Estos sensores miden la diferencia de presión entre dos puntos, en principio a distintas presiones.



Figura 1.1. Ejemplos de sensor de presión relativa a la izquierda y sensor de presión diferencial a la derecha.

Otro criterio de clasificación de estos sensores es basándose en su principio de funcionamiento. Aquí hay muchas opciones, desde sensores de fibra óptica, hasta elementos vibrantes o piezo-resistivos.

Los sensores de presión más apropiados para nuestra aplicación, por su bajo costo y ya que son muy fáciles de encontrar son aquellos basados en semiconductores piezo-resistivos, los cuales cambian su resistencia en función de la presión a la que se encuentren.

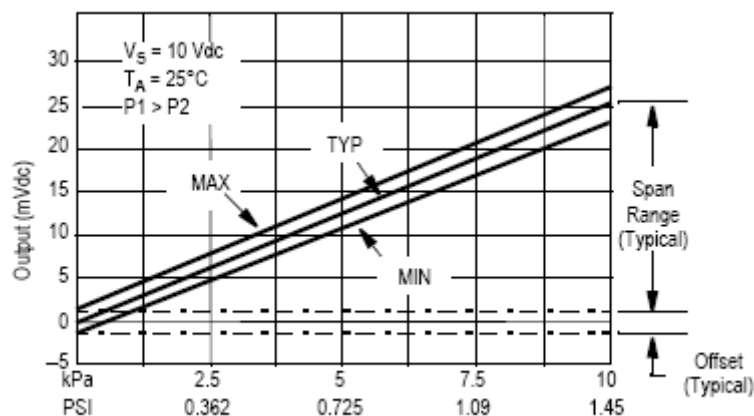


Figura 1.2. Curva tímicrocontroladora de Voltaje presión de un sensor piezo-resistivo.

Con respecto a los rangos y al tipo de los sensores de presión que utilizaremos, estos deben ser de presión relativa para medir profundidad (utilizaremos un sensor primario

para poder medir profundidades de hasta 100 cm). Estos sensores se consiguen gratis pidiendo muestras.

1.3. Sensores de Infrarrojos:

Contienen un pequeño dispositivo que puede alojar en su interior un diodo emisor de infrarrojos que trabaja a una longitud de onda determinada y un receptor, que puede ser un fototransistor (detección de línea o proximidad) que se dispone en paralelo al emisor apuntando ambos en la misma dirección, puede ser un sensor lineal (medidor de distancias), que dependiendo del modelo su salida puede ser analógica, o digital (posiblemente booleana).

Dado el objetivo de la aplicación que es el de poder medir distancias a las que se encuentran determinados objetos, los sensores de infrarrojos más adecuados son los lineales.

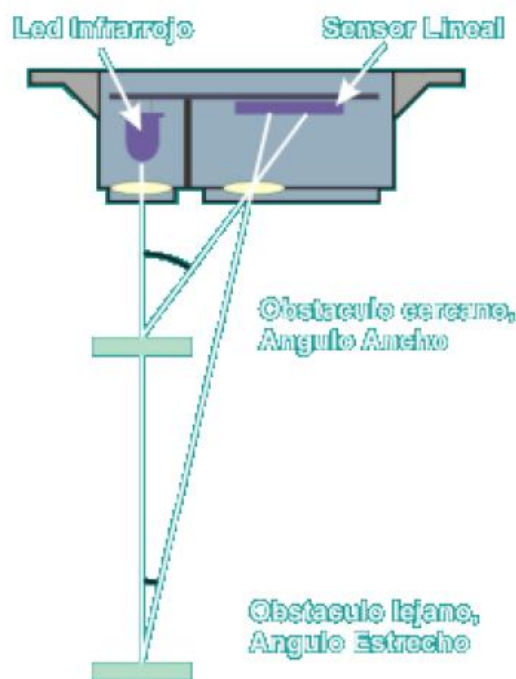


Figura 1.3. Principio de Funcionamiento de un sensor lineal.

En éstos la luz del emisor pasa a través de una lente que concentra los rayos de luz formando un único rayo lo más concentrado posible para así mejorar la directividad del sensor, la luz va recta hacia adelante y cuando encuentra un obstáculo reflectante rebota y retorna con cierto ángulo de inclinación dependiendo de la distancia, la luz que retorna es concentrada por otra lente y así todos los rayos de luz inciden en un único punto del sensor de luz infrarroja que contiene en la parte receptora del dispositivo. Este sensor es un CCD lineal y dependiendo del ángulo de recepción de la luz incidirá esta en un punto u otro del sensor pudiendo de esta manera obtener un valor lineal y proporcional al ángulo de recepción del haz de luz [10].

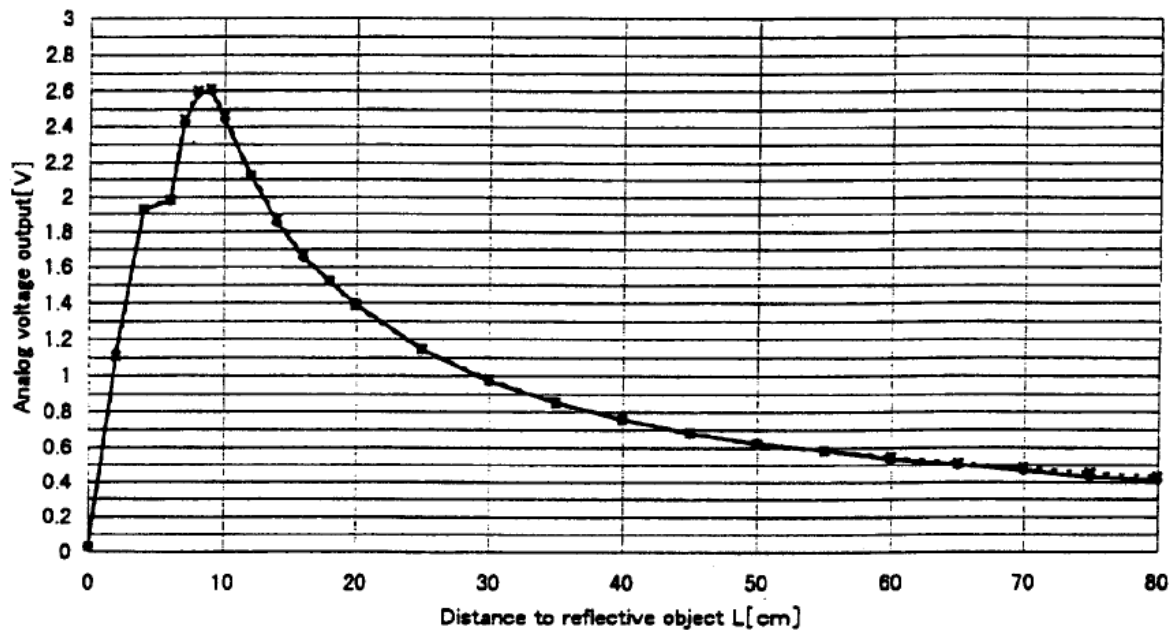


Figura 1.4. Curva tensión distancia característica de este tipo de sensores.

Los sensores de infrarrojos más baratos y fáciles de encontrar son los de la serie GP2DXX de sharp. De los cuales hay modelos tanto analógicos como digitales.



Figura 1.5. Ejemplo de sensor sharp GP2DXX

En los modelos analógicos la salida es un voltaje proporcional a la distancia medida, mientras que en los modelos digitales la lectura será de 8 bits serie con reloj externo. En los modelos Booleanos la salida será de 1 bit y este marcará el paso por la zona de histéresis del sensor con lo cual sólo tendremos una medición de una distancia fija. Dada nuestra aplicación nos concentraremos únicamente en sensores digitales de 8 bits.

Los modelos disponibles son los siguientes:

Modelo	Rango de medida	Tipo de salida
GP2D-02	10 a 80 cm.	Digital 8 bits
GP2D-05	10 a 80 cm. (ajustable)	Lógica 1 bit
GP2D-12	10 a 80 cm	Analógica (0-3V)
GP2D-15	24cm de fabrica	Lógica 1 bit
GP2D-120	4 a 30 cm.	Analógica (0-3V)

Para aislar el sensor del agua, este puede ser colocado en un recipiente, en el cual se debe cumplir que el led y el receptor del sensor no estén a más de 1mm del filtro óptico. (más información al respecto en la hoja de datos de los sensores).

Por último vale la pena resaltar que si bien el funcionamiento usual de estos sensores es en aire, en otros proyectos fueron utilizados en agua obteniéndose buenos resultados, considerando siempre objetos a detectar reflectivos. El largo de estos sensores es de 3.7 centímetros, y tienen un costo de alrededor de 12 dólares.

1.4. Sensores de ultrasonido

Los sensores de ultrasonido basan su funcionamiento en emitir sonido a altas frecuencias (de aquí su nombre), midiendo el tiempo entre la emisión y la recepción del eco.

El funcionamiento básico de los ultrasonidos como medidores de distancia se muestra de una manera muy clara en el siguiente esquema, donde se tiene un receptor que emite un pulso de ultrasonido que rebota sobre un determinado objeto y la reflexión de ese pulso es detectada por un receptor de ultrasonidos:

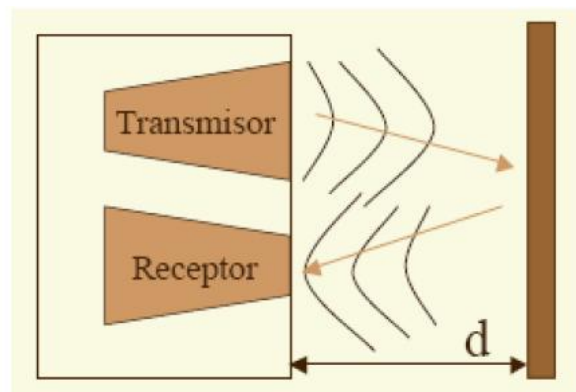


Figura 1.6. Principio de funcionamiento de los sensores de ultrasonido.

La mayoría de los sensores de ultrasonido de bajo coste se basan en la emisión de un pulso de ultrasonido cuyo lóbulo, o campo de acción, es de forma cónica. Midiendo el tiempo que transcurre entre la emisión del sonido y la percepción del eco se puede establecer la distancia a la que se encuentra el obstáculo que ha producido la reflexión de la onda sonora, mediante la fórmula [11]:

$$D = V t / 2$$

donde V es la velocidad del sonido en el medio y t es el tiempo transcurrido entre la emisión y recepción del pulso.

El campo de actuación del pulso que se emite desde un transductor de ultrasonido tiene forma cónica. El eco que se recibe como respuesta a la reflexión del sonido indica la presencia del objeto más cercano que se encuentra dentro del cono acústico y no especifica en ningún momento la localización angular del mismo [12]. Aunque la máxima probabilidad es que el objeto detectado esté sobre el eje central del cono

acústico, la probabilidad de que el eco se haya producido por un objeto presente en la periferia del eje central no es en absoluto despreciable y ha de ser tomada en cuenta y tratada convenientemente.

Un factor de error muy común es el conocido como falsos ecos [11]. Estos falsos ecos se pueden producir por razones diferentes: Puede darse el caso en que la onda emitida por el transductor se refleje varias veces en diversas superficies antes de que vuelva a incidir en el transductor (si es que incide). Este fenómeno, conocido como reflexiones múltiples, implica que la lectura del sensor evidencia la presencia de un obstáculo a una distancia proporcional al tiempo transcurrido en el viaje de la onda; es decir, una distancia mucho mayor que a la que está en realidad el obstáculo más cercano, que pudo producir la primera reflexión de la onda. Otra fuente más común de falsos ecos, conocida como crosstalk, se produce cuando se emplea un cinturón de ultrasonidos donde una serie de sensores están trabajando al mismo tiempo. En este caso puede ocurrir (y ocurre con una frecuencia relativamente alta) que un sensor emita un pulso y sea recibido por otro sensor que estuviese esperando el eco del pulso que él había enviado con anterioridad (o viceversa).

Una dificultad de utilizar este tipo de sensores en nuestra aplicación es que hay que realizar toda la electrónica del circuito detector del eco. Esto se puede solucionar encontrando el chip LM1812 de National, que actualmente se encuentra fuera de producción. Este chip se conecta a un microcontrolador y al transductor de ultrasonido como se muestra en la siguiente figura. Lo que realiza básicamente es enviar y recibir una frecuencia (por ejemplo: 200 kHz) del transductor.

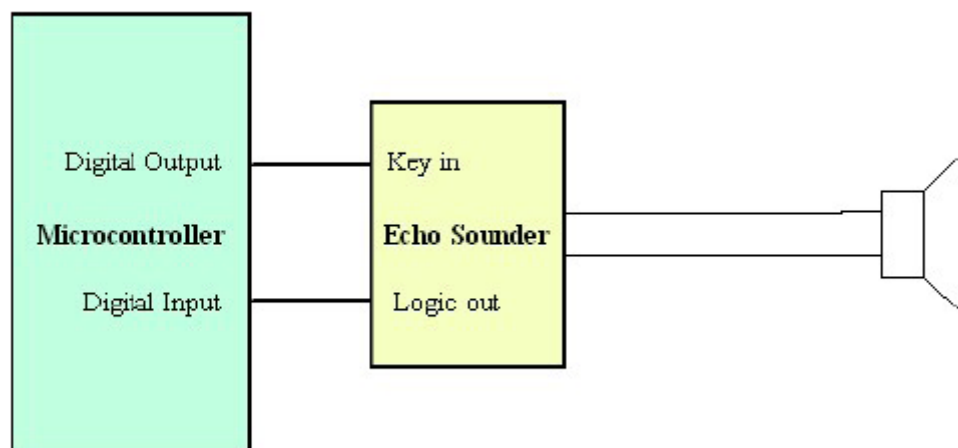


Figura 1.7. Conexión del chip LM1812.

Tanto la modulación del pulso a 200 kHz, como el control del tiempo deben ser realizados por el microcontrolador. Aunque este chip no se encuentra fácilmente en el mercado, tanto el microcontrolador como el transductor pueden ser encontrados con mucha facilidad.

El costo del transductor es de alrededor de 10 dólares para 40 kHz (Prowave 400EP18A), mientras que encontramos sensores de 200 kHz por unos 40 dólares (Murata MA200A1).

Por las razones ya vistas, aparenta ser mejor la opción de utilizar sensores infrarrojos que de ultrasonidos.

1.5. Conclusión

Para dotar a los robots de la capacidad para realizar pequeñas tareas de forma independiente es necesario dotarlas de sensores. En nuestro caso particular, debemos utilizar sensores para medir profundidad bajo el agua, y distancia a objetos también bajo la misma.

Los sensores más adecuados para medir profundidades son los sensores de presión piezo-resistivos, que poseen una respuesta lineal con la profundidad.

Con respecto a los sensores para detectar distancias a objetos, los sensores disponibles son infrarrojos, o bien de ultrasonidos. Mientras que los sensores de infrarrojos son sensores ópticos y basan su funcionamiento en determinar el ángulo con el que se refleja un haz de luz, los sensores de ultrasonido basan su funcionamiento en un principio acústico, que es el de medir el tiempo entre que se emite un pulso y llega el eco.

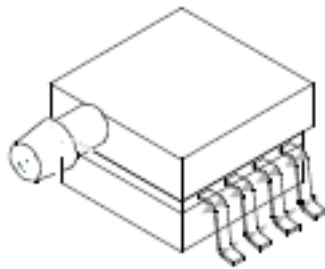
Los sensores de infrarrojos parecen ser la mejor opción para nuestra aplicación por su bajo costo, y no requerir de una electrónica extra para su funcionamiento. Como es en el caso de los sensores de ultrasonido, donde es necesario un circuito para detectar el eco.

2. Ensayo de los sensores

2.1. Ensayo de los sensores de Presión

2.1.1. Elección de la electrónica asociada

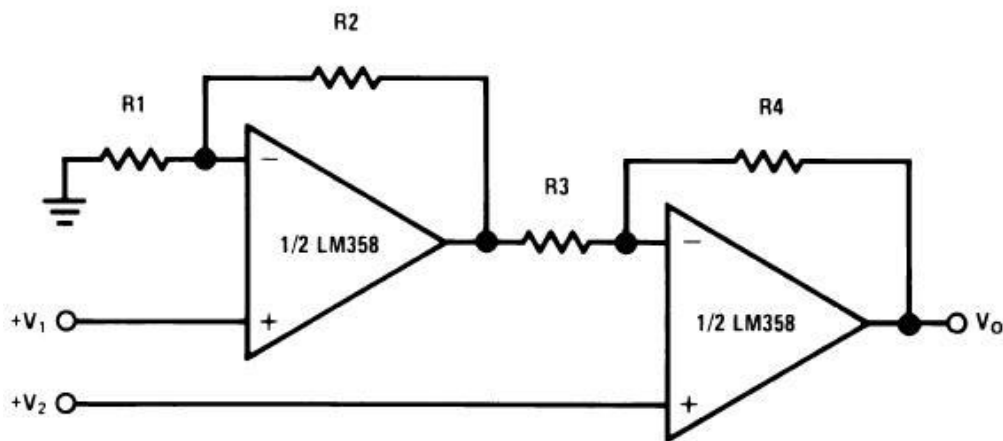
El sensor ensayado fue el MPXV2010GP, de Freescale Semiconductor, que fue descrito a rasgos generales en la sección de elección del sistema de sensores.



MPXV2010GP

Figura 2.1. Esquema del sensor utilizado

Para tener una salida legible desde el microcontrolador, tuvimos que amplificar la señal, para lo cual utilizamos un amplificador de instrumentación con la siguiente implementación:



$$\frac{R1}{R2} = \frac{R4}{R3}$$
$$V_O = \left(1 + \frac{R4}{R3}\right)(V_2 - V_1)$$

Figura 2.2. Implementación del amplificador de instrumentación

donde el amplificador utilizado corresponde a un LM358 (Low Power Dual Operational Amplifier) de National Semiconductor.

Para obtener una ganancia adecuada, de modo de tener a la entrada del microcontrolador valores razonables de tensiones, tomamos los siguientes valores para las resistencias:

$$\begin{aligned} R1 &= R4 = 180 \text{ k}\Omega \\ R2 &= R3 = 470 \text{ }\Omega \end{aligned}$$

En este caso queda la ganancia aproximadamente igual a 380 V/V. Como la tensión máxima obtenida en el caso de colocar solamente el sensor sin una posterior etapa de amplificación estaba en el entorno de los 10mV para una tensión de alimentación de 5 V, es de esperar que la salida luego de amplificar se obtengan valores máximos de alrededor de 3.8 V, los cuales ahora son razonables para la lectura desde el micro.

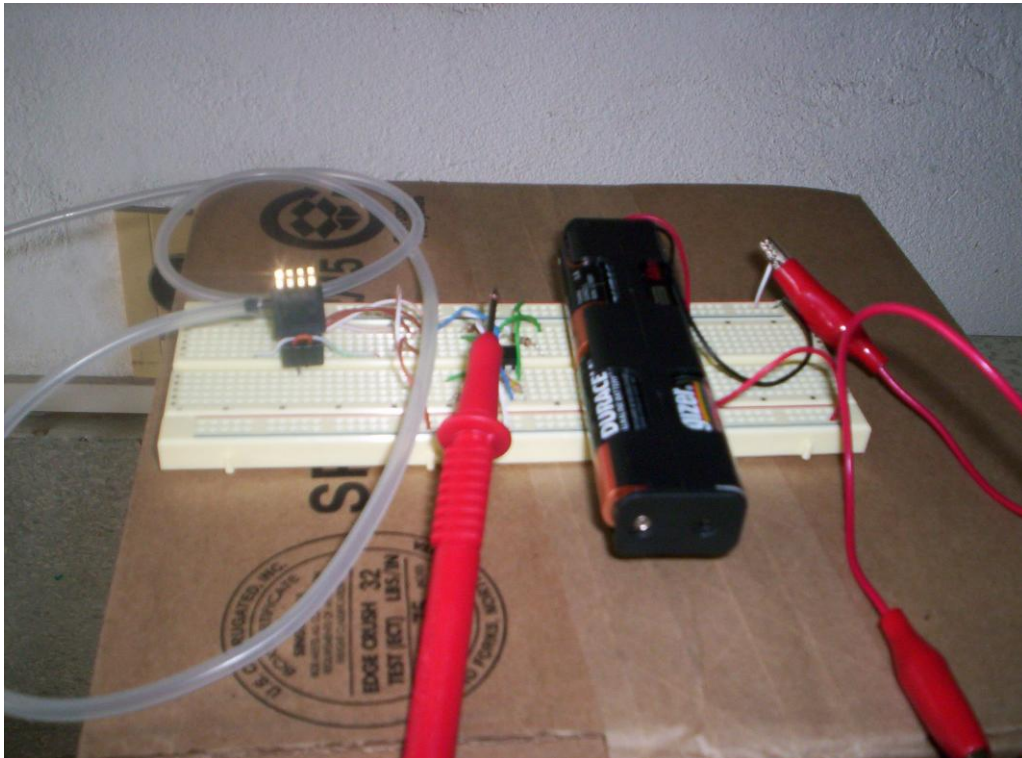


Figura 2.3. Circuito implementado para el ensayo del sensor, se puede ver a la izquierda el sensor conectado al tubo.

2.1.2. Descripción del ensayo

Para ensayar el sensor, procedimos a conectarle un tubo de silicona (de longitud 60 cm.) e ir sumergiéndolo en el agua, midiendo para valores fijos de profundidad las respectivas salidas de tensión. Este ensayo lo repetimos unas 10 veces para tener datos de mayor confiabilidad.

Estos ensayos fueron realizados llegándose a profundidades de alrededor de 50 cm., por falta de recipiente con profundidad adecuada para llegar a mayores valores. Igualmente, esto no es algo crítico, puesto que la profundidad a la que se encontrara el pez será menor a los 50cm.

La fuente de alimentación fueron pilas, produciendo una tensión de alimentación de 4.93 V. El ensayo fue realizado a temperatura de unos 17 °C.

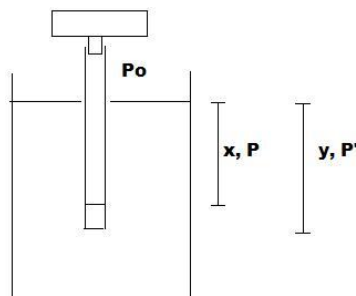


Figura 2.4. Vista general del ensayo, se ve a la izquierda el tubo sumergido en el recipiente y a la izquierda el multímetro arrojando los correspondientes valores de tensión.

2.1.3. Análisis de los resultados

A partir de las medidas, verificamos que la característica presión (en nuestro caso profundidad) – tensión en bornes del sensor es lineal¹, esto es de esperar pues era la información que se daba en la hoja de datos. Como disponíamos de varias lecturas de

1. En realidad no es lineal, pero para valores grandes de x , se puede aproximar casi como lineal, esto es debido a lo siguiente:



Nosotros medimos P e y , se cumple que : $P = P_o + \rho g x$, $P' = P_o + \rho g y$ por lo que : $P - P' = \rho g (x - y) = a(x - y)$. Como $PV = \text{cte}$ para el volumen de aire que se encuentra en el tubo: $P_o L S = P(L - y + x)$. Utilizando las dos ultimas ecuaciones, se llega a que: $P = P' + aL(1 - P_o/P')$. Considerando la característica lineal del sensor, se tiene que: $P = b(V + \text{offset})$ por lo que utilizando la expresión que vincula P y P' y considerando $P_o = 1$, se llega a que: $P = aL(1 - a/b(V + \text{offset})) + b(V + \text{offset}) = ax + P_o$, por lo que se llega a que $x = qv + t - r/(V + \text{offset})$. Lo que difiere en el ultimo factor de la característica esperada, el cual es no despreciable para tensiones (o profundidades) chicas, de aquí que la característica si se podrá considerar lineal para profundidades grandes.

tensión para distintas profundidades en el rango de 0 a 50 cm procedimos a determinar en Matlab por el método de los mínimos cuadrados, la recta que mejor aproxima a estas medidas.

Una vez encontrada la recta que mejor aproxima a las medidas, esta servirá para luego utilizar en la programación del microcontrolador una tabla con rangos de valores basados en esta recta.

Los valores obtenidos fueron próximos a los esperados del diseño. Se obtuvo un voltaje de alrededor de 1.7 V para una profundidad de 48cm, y se observó que se empezaban a obtener lecturas no nulas a partir de los 4 cm.

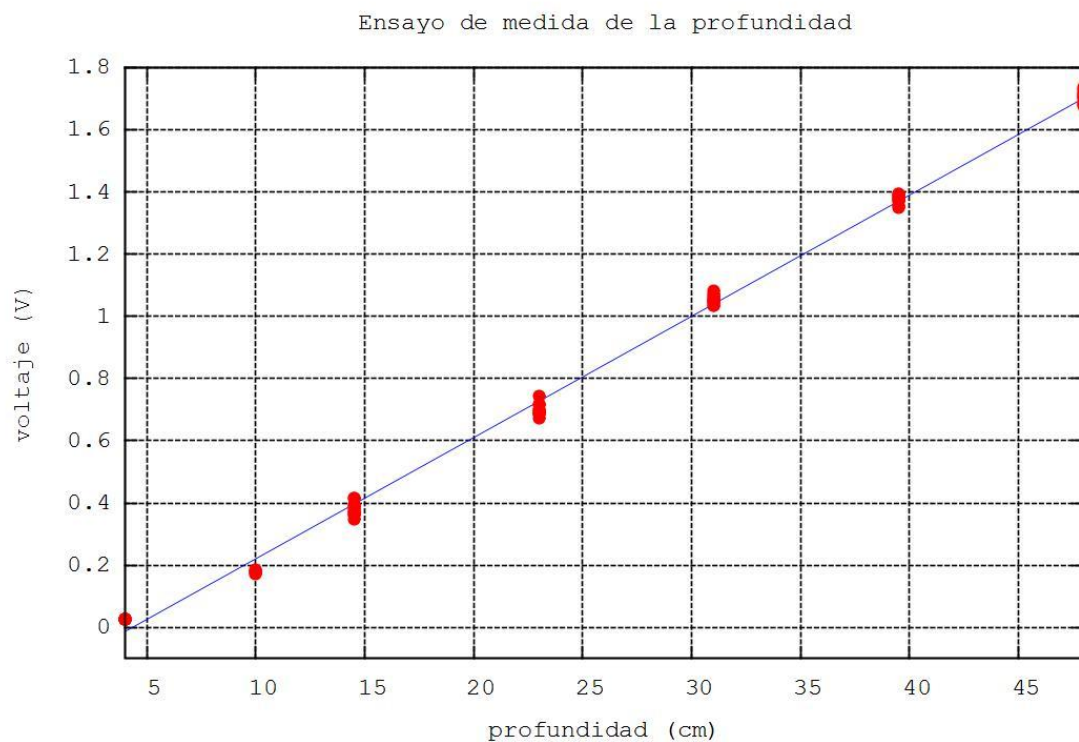


Figura 2.5. Resultados obtenidos en el ensayo.

2.1.4. Conclusión

El sensor funcionó para medir profundidades de hasta 50 cm (que es hasta donde fue ensayado), obteniéndose valores de tensiones adecuados para su lectura desde el microcontrolador. Se verificó el comportamiento lineal del sensor y se determinó mediante el método de los mínimos cuadrados la recta que mejor aproxima a las muestras provenientes de los ensayos.

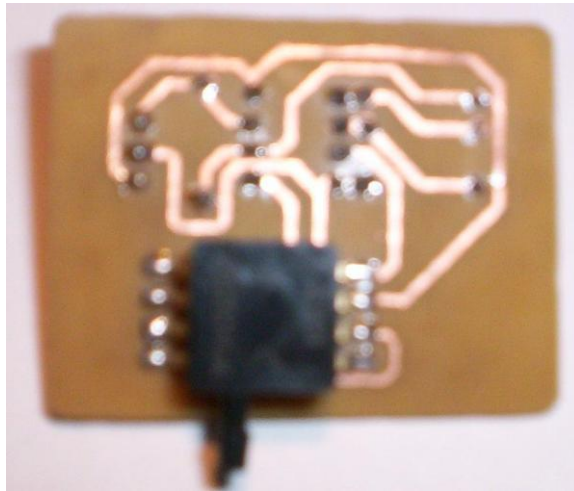


Figura 2.6. Circuito impreso (dimensiones 3 x 4 cm).

2.2. Ensayo de los sensores de distancia

2.2.1. Elección de la electrónica asociada

El sensor ensayado fue el GP2Y0A21YKOF, de Sharp, que fue descrito a rasgos generales en la sección de elección del sistema de sensores.



Figura 2.7. Fotografía del sensor utilizado

Permite medir distancias entre 6 y 80 cm, y posee una característica voltaje-distancia no lineal (siendo el voltaje inversamente proporcional a la distancia al objeto), como ya fue mostrado en secciones previas.

Posee además la propiedad de dar una salida que no necesita ser amplificada, a diferencia de la salida de los sensores de presión.

2.2.2. Descripción del ensayo

Para ensayar el sensor bajo el agua, procedimos a colocar el sensor en un recipiente traslúcido como se muestra en la imagen, e ir midiendo las salidas de tensión para distintas distancias a un objeto (una tabla que fuimos moviendo). Este ensayo lo repetimos unas 4 veces para tener datos de mayor confiabilidad.

Repetimos el mismo ensayo fuera del agua para comparar resultados, el procedimiento realizado en este caso fue análogo al descrito previamente. Esto fue realizado para comparar el comportamiento del sensor en ambos medios. Por motivos de falta de espacio en el caso del ensayo en el agua, solo llegamos a poder medir distancias de hasta el orden de los 50 cm.

La fuente de alimentación fueron pilas, produciendo una tensión de alimentación de 5.40 V.



Figura 2.8. Vista general del ensayo

2.2.3. Análisis de los resultados

Para el ensayo realizado con el sensor en el agua:

A partir de las medidas, verificamos que la característica distancia – tensión de salida del sensor es similar a la dada en la hoja de datos. Como disponíamos de varias lecturas de tensión para distintas distancias en el rango de 10 a 50 cm procedimos a determinar en Matlab por el método de los mínimos cuadrados no lineales, la curva que mejor aproxima a estas medidas (en este caso buscamos una hipérbola).

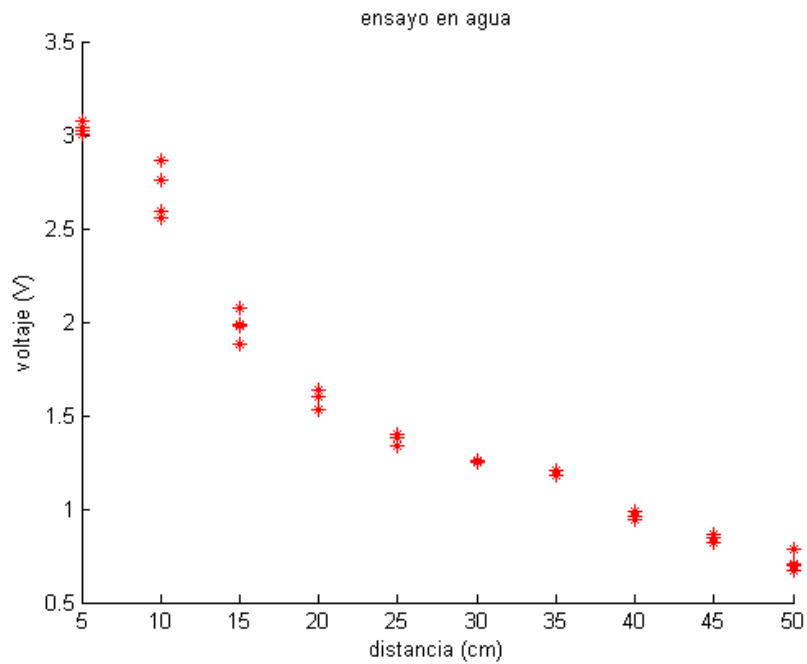


Figura 2.9. Característica voltaje – distancia en el agua

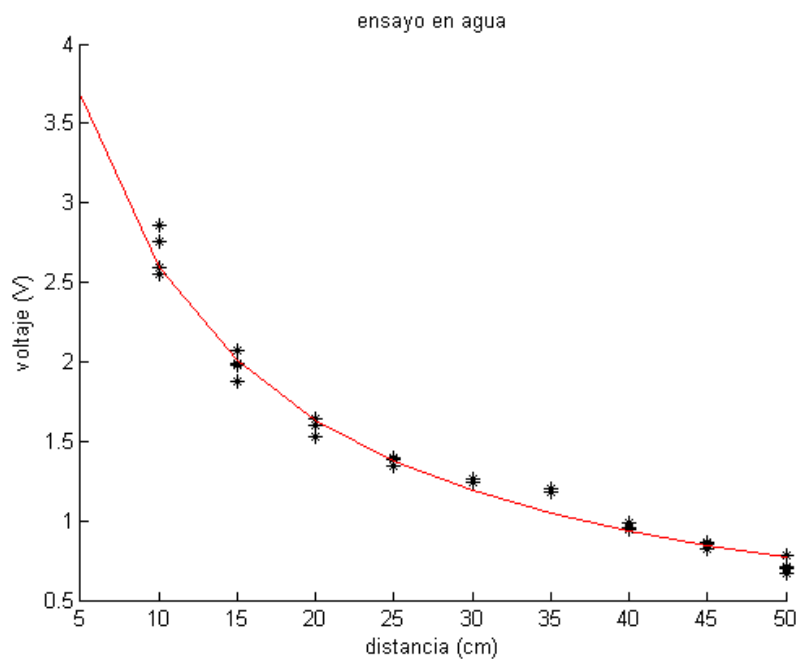


Figura 2.10. Hipérbola que mejor aproxima a las muestras

Una vez encontrada la curva que mejor aproxima a las medidas, esta servirá para luego utilizar en la programación del microcontrolador una tabla con rangos de valores basados en esta curva.

Con respecto a los valores obtenidos de realizar el ensayo fuera del agua, estos se muestran en la siguiente gráfica:

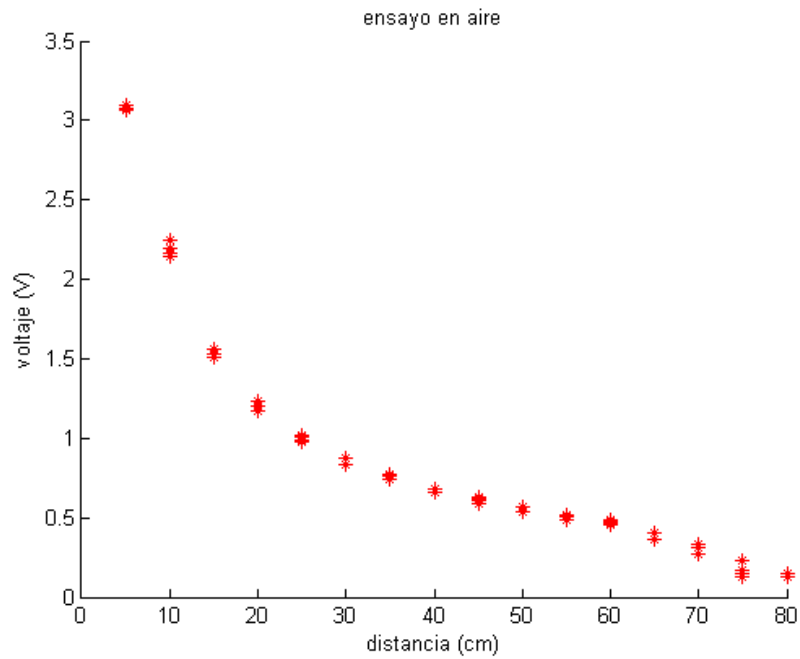


Figura 2.11. Característica voltaje – distancia en el aire

Los valores obtenidos fueron más o menos los mismos que cuando el ensayo fue realizado fuera del agua.

En la siguiente gráfica se comparan los resultados, los cuales confirman que estos sensores son más que aptos para medir distancias bajo el agua.

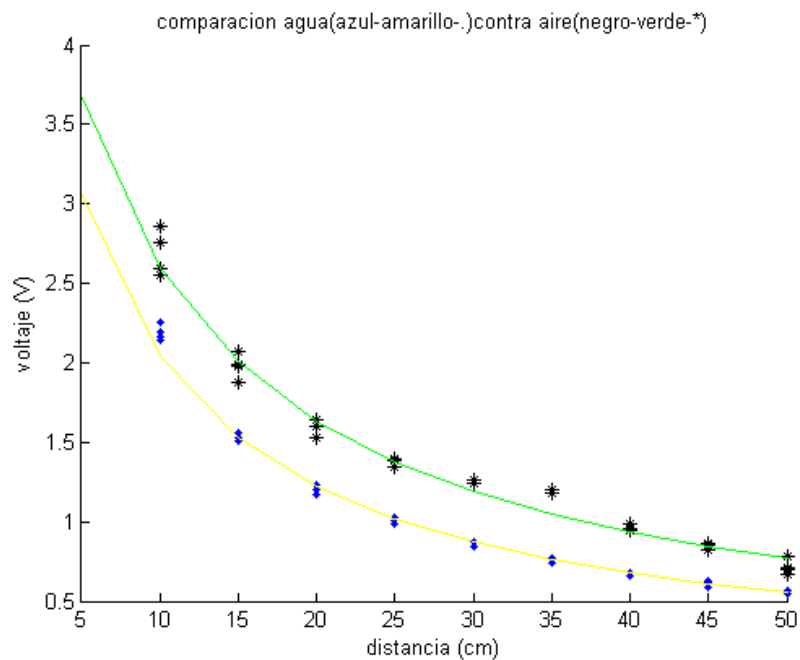


Figura 2.12. Resultados obtenidos en el ensayo, dentro y fuera del agua.

La curva inferior corresponde al comportamiento sobre el agua, mientras que la inferior corresponde al comportamiento bajo el agua.

2.2.4. Conclusión

El sensor funcionó para medir distancias de hasta 50 cm bajo el agua (en realidad es de esperar que llegue hasta los 80 cm como en el aire, pero por falta de espacio, este fue el mayor valor que pudimos medir), obteniéndose valores de tensiones adecuados para su lectura desde el microcontrolador. Se verificó el comportamiento del sensor tanto dentro como fuera del agua y se determinó mediante el método de los mínimos cuadrados no lineales la curva que mejor aproxima a las muestras provenientes de los ensayos. El comportamiento del sensor fue más que satisfactorio, tanto sobre como bajo el agua.

3. Sistema de Control de Profundidad

El mecanismo a utilizar es sencillo, y consiste de una perita de goma que contiene agua, la cual es presionada por un servomotor. La idea básica es variar la masa total del pez y, en consecuencia la densidad. Es bien sabido que cuando la densidad del pez iguale la del agua se obtiene un equilibrio en el que el pez se sumerge completamente. Naturalmente, si ahora le quitamos masa (mediante presión a la perita) el pez tenderá a ir a la superficie y contrariamente si agregamos masa al pez.

El servomotor fue elegido dado que es más simple con él controlar la posición angular del eje, con respecto a otros motores convencionales. A continuación se describen el funcionamiento y la conexión de los servomotores, destacándose la característica de la Modulación de Ancho de Pulsos (PWM).



Figura 3.1. Foto del servo motor.

Un servomotor es un dispositivo en forma de caja negra al que llegan tres cables. Contiene un pequeño motor, una caja de engranajes, un potenciómetro de un valor aproximado de 5K y un pequeño circuito integrado. Este motor eléctrico en miniatura ataca a la magnitud que se ha de controlar: el giro y posicionamiento del eje del motor.

A su vez, el movimiento de rotación angular del motor modifica la posición del potenciómetro interno, que controla un monoestable también integrado en el servomotor. El eje del motor puede ser girado hasta una posición angular específica mediante una señal de control. Mientras se mantenga esta señal de control, el servomotor mantendrá la posición angular del eje. Si la señal de control cambia, también cambia la posición de eje.

3.1. Conexiones

Los servomotores tienen tres cables: el de masa (-), el de alimentación (+) y el de la señal de control. El positivo se conecta a + 5 y el de señal de control a una fuente de pulsos variables entre 1 y 2 milisegundos de duración (figura 3) que se repiten con una frecuencia de unos 12-20 ms [13]. Los cables de los servos siguen casi siempre el mismo código de colores; por ejemplo en los Futaba color rojo (V+), negro (masa) y blanco (señal de control). Los fabricantes JR y Graupner colocan el cable de la señal de

control de color naranja, mientras que algunos servos Sanwa tienen el cable de masa de color azul.

3.2. Funcionamiento

La velocidad del motor, así como la dirección del movimiento de los servos se controla mediante servo pulsos modulados en amplitud [14]. En la Figura 3.2 se muestra la forma de los servo pulsos. El servomotor convierte los pulsos en un movimiento mecánico. La magnitud del giro del eje del servo es proporcional a la anchura del pulso que llega por la línea de control. Este tipo de pulsos está formado por una señal digital que se genera aproximadamente cada 20 milisegundos. La anchura de estos pulsos va de un mínimo de 1 ms a un máximo de 2 ms. Aunque la relación anchura del pulso y la posición del eje no está estandarizada, lo normal es que trenes de pulsos de 1,5 ms lleven el eje del servo al centro de su rango, anchura neutra. Si la anchura del pulso es de 1 ms, el servomotor se posiciona en el extremo izquierdo, mientras que si el pulso tiene una anchura de 2 ms la posición del servo es el extremo opuesto. Esta técnica se conoce como modulación por anchura de pulso, en inglés PWM (*Pulse Width Modulation*). El servomotor trabaja comparando la anchura del pulso de entrada con la anchura del pulso producido por el *timer* interno. A su vez, el período del *timer* interno es controlado por el potenciómetro acoplado al eje del servo. La diferencia entre la anchura del pulso de entrada y la anchura del pulso interno se utiliza como señal de error. El motor girará modificando la posición del potenciómetro de retroalimentación. Cuando llega el siguiente pulso se vuelve a realizar la comparación, comprobando de forma continua la posición del eje y realizando también constantemente las correcciones necesarias en la posición del mismo. Como se ha podido apreciar, se trata de un bucle de retroalimentación negativa. Si la posición del potenciómetro no se iguala con la posición deseada del eje, el motor se moverá hacia adelante o hacia atrás, hasta que la posición del potenciómetro sea equivalente a la posición deseada del eje. En este momento la corriente del motor se apaga [13].

3.3. Modulación por ancho de Pulsos

La modulación por ancho de pulsos (o PWM, de *pulse-width modulation* en inglés) es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (por ejemplo sinusoidal o cuadrada).

El **ciclo de trabajo** de una señal periódica es el ancho relativo de su parte positiva en relación al período. Matemáticamente [15]:

$$D = \frac{\tau}{T}$$

D es el ciclo de trabajo

τ es el tiempo del período en que la función es positiva (ancho del pulso)

T es el período de la función

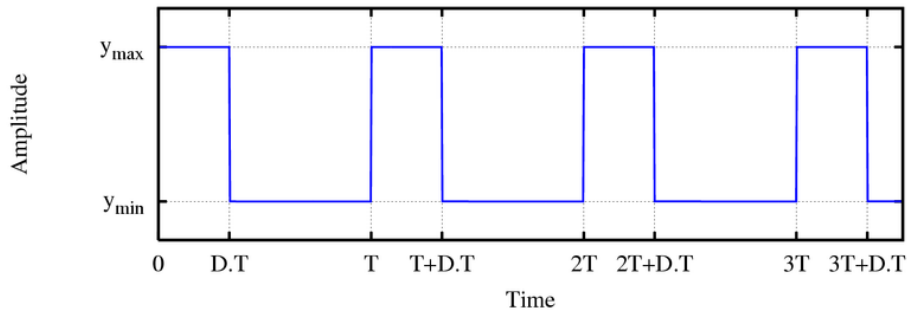


Figura 3.2. PWM

En la actualidad, existen muchos circuitos integrados que integran la función PWM, además de otros muy particulares para lograr circuitos funcionales que puedan controlar fuentes conmutadas, controles de motores, controles de elementos termoelectrónicos, choppers para sensores en ambientes ruidosos y algunas otras aplicaciones. Se distinguen por fabricar este tipo de integrados de compañías como Texas Instruments, National Semiconductor, Maxim, y algunas otras más.

3.4. En relación a los motores

La modulación por ancho de pulsos (PWM o *Pulse Width Modulation* en inglés) es una técnica utilizada para regular la velocidad de giro de los motores eléctricos. Mantiene el par motor constante y no supone un desaprovechamiento de la energía eléctrica. Se utiliza tanto en corriente continua como en alterna, como su nombre lo indica, al controlar: un momento alto (encendido o alimentado) y un momento bajo (apagado o desconectado), controlado normalmente por relevadores (baja frecuencia) o mosfet (alta frecuencia) [16].

Otros sistemas para regular la velocidad modifican la tensión eléctrica, con lo que disminuye el par motor; o interponen una resistencia eléctrica, con lo que se pierde energía en forma de calor en esta resistencia.

Otra forma de regular el giro del motor es variando el tiempo entre pulsos modulación por frecuencia de pulsos de duración constante.

En los motores de corriente alterna también se puede utilizar la variación de frecuencia. La modulación por ancho de pulsos también se usa para controlar servomotores, los cuales modifican su posición de acuerdo al ancho del pulso enviado cada un cierto período que depende de cada servo motor. Esta información puede ser enviada utilizando un microprocesador [16].

3.5.Ensayos del servomotor

3.5.1. Elección de la electrónica asociada

El servomotor SO3T 2BB fue elegido el cual posee una capacidad de torque de 7 kg/cm.



Figura 3.3. Servomotor SO3T 2BB

3.5.2. Descripción del ensayo

Para poder ensayar este dispositivo se realizó un programa en Assembler para el microcontrolador 16F876, con el cual se hace variar la tensión de entrada al conversor A/D por medio de un potenciómetro el cual regula el duty cycle de la señal PWM.

Tabla 1. Valores obtenidos

Tiempo (ms.)	Posición angular (grados)
0.6	0
0.8	0
1.3	45
1.7	90
2.1	135
2.5	180

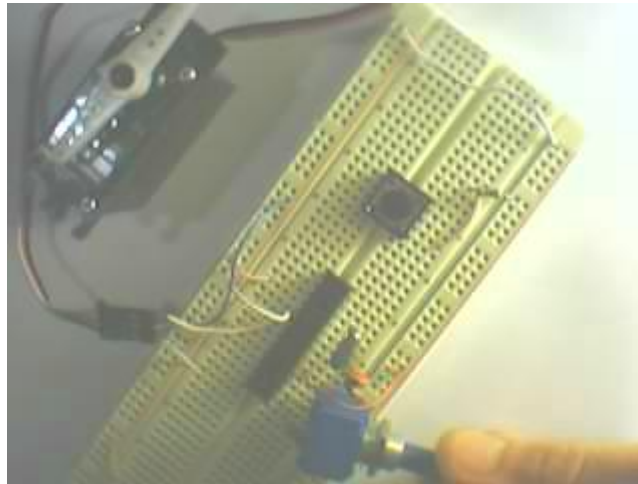


Figura 3.4. Generación de PWM mediante un Microcontrolador

Para generar pulsos PWM se generó un código en Assembler que se basa en rutinas Delay, estableciendo alternadamente valores lógicos 0 y 1 respetando los tiempos para controlar un servomotor. De esta manera podemos controlar la amplitud del ángulo de giro, que va a hacer contacto con la perita de goma e intercambiar la masa de agua para obtener el equilibrio deseado. Se adjunta en el anexo el código en Assembler que genera los pulsos PWM. Vale mencionar que variando la duración de los delays es posible elegir diferentes amplitudes de giro.

3.6. Sistema de Control de Movimiento

Para el sistema de dirección del pez se utilizará un servomotor al igual que para el sistema de control de profundidad, ya que posee mejor precisión en movimientos pequeños con respecto a otros motores.

4. Elección de microcontroladores a utilizar

4.1.Introducción

La arquitectura del sistema es la que permite procesar los datos obtenidos de los sensores previamente descritos en las secciones anteriores, y de esta manera decidir qué acciones mecánicas (movimientos) seguir para continuar con una navegación autónoma. Para que el usuario pueda fácilmente comprobar el estado del sistema podríamos decir que la comunicación con una PC deja de ser un opcional y se convierte en una necesidad.

Esta también es útil para poder eventualmente guiar al pez a control del usuario. Se buscará también realizar una programación en el circuito (por ejemplo ICSP) del microcontrolador para de esta manera no tener que estar en todo momento armando y desarmando al pez con los peligros que esto conlleva. Vale la pena resaltar que en la mayoría de los proyectos de tesis o de final de carrera encontrados, esta comunicación y esta programación en circuito no está presente (o están presentes parcialmente o bajo fuertes restricciones).

4.2.Opciones de arquitectura del sistema

La opción más intuitiva es la de utilizar un microcontrolador, el cual sea responsable de realizar todas las funciones de sensado del medio, comunicación y control de motores, además de llevar a cabo el control en cuanto a toma de decisiones de trayectorias a seguir.

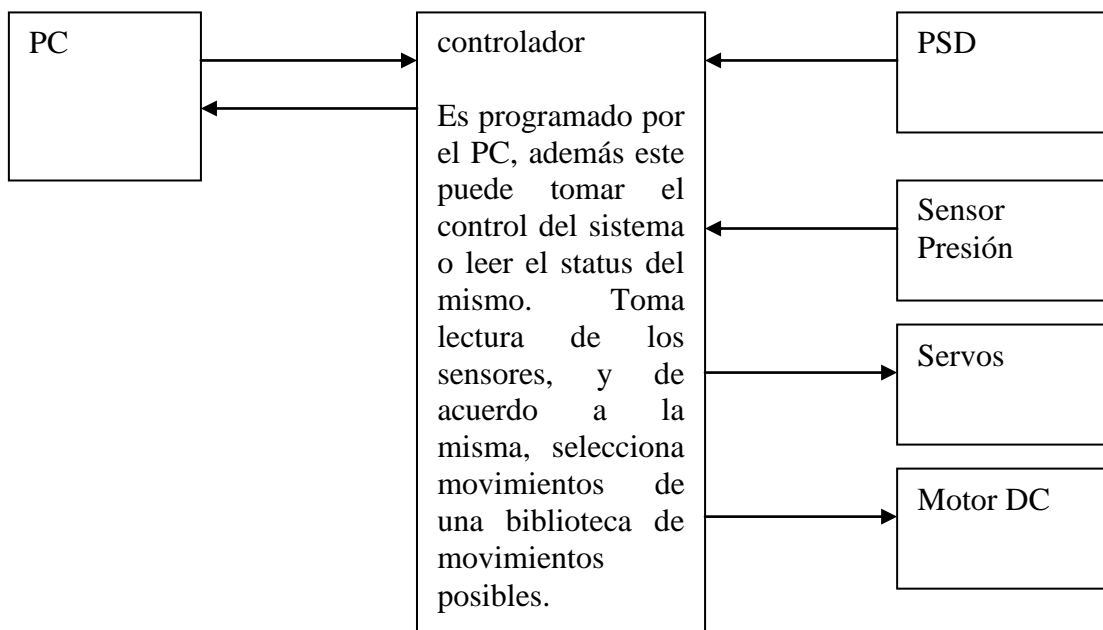


Figura 4.1. Arquitectura con un solo controlador

Esta arquitectura es similar a la realizada en [17] utilizando un microcontrolador Motorola 68HC11, [18] utilizando un CPU Pentium III, y [19] utilizando un microcontrolador Motorola M68332.

Una variante de la arquitectura anterior, que requiere más de un controlador dentro del pez, es la siguiente:

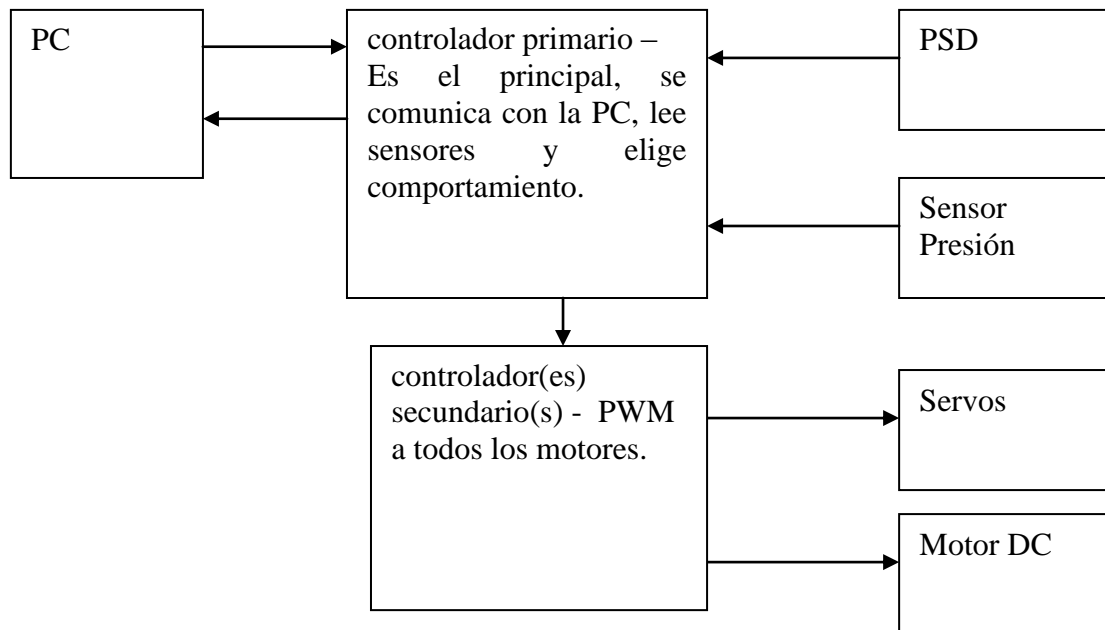


Figura 4.2. Arquitectura con varios controladores

Esta arquitectura es similar a la realizada en [20] Microcontrolador: principal: microcontrolador18F2580, secundario: microcontrolador16F768A.

Vale la pena observar que en la primera arquitectura, usualmente son utilizados microcontroladores potentes, ya que se puede llegar a requerir una gran cantidad de memoria y de entradas y salidas en uso para poder trabajar todo a la vez.

En la segunda arquitectura propuesta, ya a costa de tener más microcontroladores, se simplifica la complejidad de los mismos, siendo el principal un microcontrolador de mediana potencia (en cuanto a memoria y cantidad de entradas-salidas), y los secundarios microcontroladores simples).

Una tercera opción, que sería simplemente realizar la toma de los datos y PWM de los motores en el pez y realizar toda la toma de decisiones en una CPU externa.

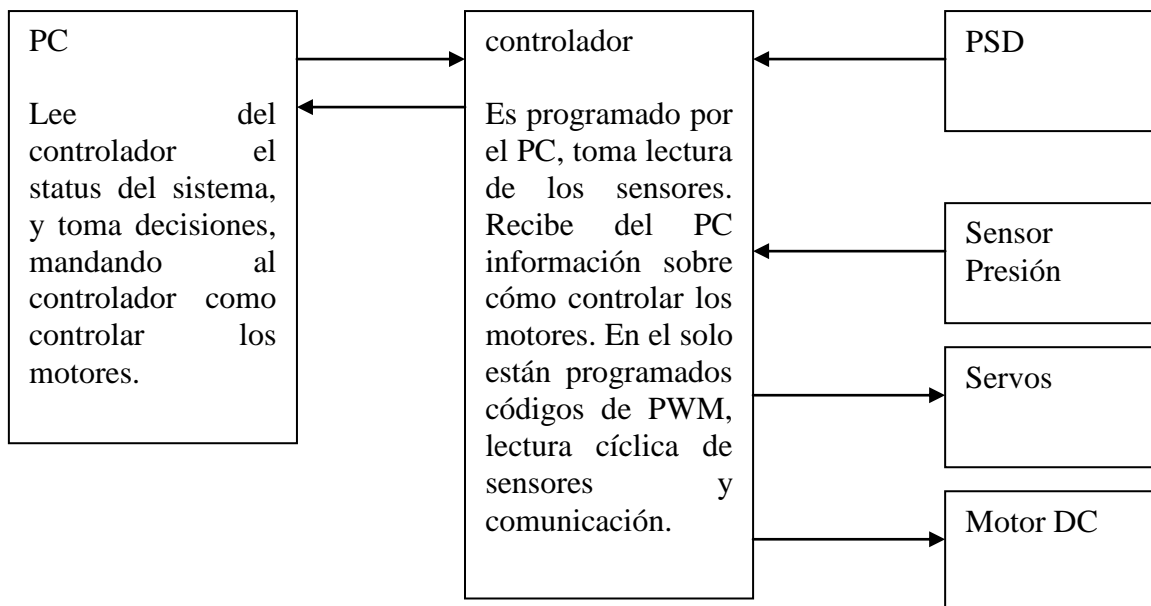


Figura 4.3. Arquitectura con control externo

Esta es la arquitectura a la que apuntan los trabajos del NMRI, de [21] y la utilizada por investigadores de la universidad de Essex, Inglaterra en sus primeros prototipos [1] y [22] (ya que después pasaron a una arquitectura similar a la primera, aunque con variantes, utilizando un micro Gumstix).

Esta arquitectura tiene la ventaja de que no se requiere complejidad en los microcontroladores dentro del pez, aunque sí se requiere una rápida comunicación con la PC. En la PC se pueden desarrollar sistemas de control complejos, obteniéndose resultados muy buenos en cuanto a similitud con la realidad.

4.3. Estudio de diversos microcontroladores y “Single Board Computers” para su posible utilización en el proyecto

Una componente muy importante para la arquitectura del sistema, es la elección del microcontrolador a utilizarse en la misma. Para ello, realizamos una investigación de los diversas opciones de microcontroladores disponibles en el mercado, donde obviamente, por lo vasto que es el mismo, y las muchas alternativas que hay no pudimos explorarlo en su totalidad. Lo que se muestra a continuación es una tabla comparativa de microcontroladores y single board computers, donde a nuestro entender se encuentran por lo menos dos representantes de cada gama en cuanto a funcionalidad.

Tabla 8.1 – Comparación de distintos microcontroladores y SBC										
Fabricante	Modelo	Tipo	Consumo	Tamaño	Precio	Precio con programador	Memoria programa	Memoria datos	Entradas analógicas	Salidas PWM
Rabbit	LP3500	SBC	20mA	9.3x6.6	199	399	512K	512K	8	3
Rabbit	BL1800	SBC	100mA	8.9x6.4	99	139	256K	128K	1	2
Rabbit	RCM4400W (wi-fi)	Modulo	80mA /450mA	4.7x7.2	119	149	512K	512K	0	4
Rabbit	RCM4010	Modulo	125mA	4.7x2.1	75	99	512K	512K	0	2
Microchip	PIC24HJXXX	Micro	-----	-----	-----	-----	128K	8K	8	8
Microchip	PIC18F87J10	Micro	-----	-----	-----	-----	128K	4K	15	2
Microchip	PIC18F4550	Micro	6mA	-----	-----	-----	32K	2K	13	2
Atmel	Atmega2560	Micro	20mA	-----	-----	-----	256K	8K	16	4
Atmel	Atmega128	Micro	40mA	-----	-----	-----	128K	8K	8	3
Gumstix	Connex200	SBC	120mA	8x2	109	-----	64MB sdram	16MB flash	-----	-----
Gumstix	con Atmega128	placa	-----	8x4	49	-----	-----	-----	-----	-----
Gumstix	Robostix wi-fi pack	SBC + placas	450 mA en com.	8x4	239	-----	-----	-----	-----	-----

4.4.Comparación de las distintas alternativas.

De las distintas arquitecturas posibles y tipos de microcontroladores y SBC, surgen a la vista tres opciones, las cuales se describen a continuación:

1) Utilizar una SBC con capacidades de lectura de señales analógicas y de salidas PWM a los motores. Esto correspondería en cierta forma con el primer tipo de arquitectura del sistema presentado, aunque también posee dependiendo del sistema a usar ciertos matices de la segunda.

La SBC a utilizar puede ser una de las siguientes:

- Rabbit en las distintas alternativas que se muestran en la tabla anterior
- Gumstix (con placa Robostix)

Del análisis de las funcionalidades de las mismas, se observa lo siguiente:

- Las SBC de rabbit no poseen ni memoria ni funcionalidad comparable a las de una Gumstix. Se observa que en un caso se está trabajando con una computadora con un sistema operativo y varios MB de memoria, mientras en el otro la capacidad de los controladores es comparable a la de la placa Robostix de Gumstix.
- Las SBC de Rabbit no son mucho más económicas (ver tabla), vale la pena aclarar que se muestran en la misma aquellos modelos de los distintos fabricantes que resultaron interesantes en su relación precio – capacidad y funcionalidades. Se puede decir sin temor a equivocarse que la opción de utilizar una Gumstix resultaría mucho más funcional y no mucho más cara.
- Con respecto al tamaño, el cual también es de importancia para nuestra aplicación, se observa que el de las Gumstix es menor a la de las Rabbit.
- En cuanto al consumo, sin ser del modelo ya diseñado para bajo consumo de Rabbit que es el LP3500, el consumo de las Gumstix es comparable a los demás modelos o SBC de Rabbit. Pero como la velocidad de la Gumstix está en el orden de 200 MHz, comparado con 20 o 40 MHz de Rabbit, se observa que en sí estas son mucho más eficientes que las ultimas desde el punto de vista de consumo versus funcionalidad.
- Es importante observar que en el caso de la Gumstix, se está considerando un pack con Wi-Fi incorporado, por lo que todo el asunto de la comunicación con la PC quedaría resuelto de utilizar esta SBC.

En conclusión, resulta más potente, funcional, y no mucho más caro utilizar una SBC Gumstix que una Rabbit.

2) Utilizar simplemente un microcontrolador, con la mayor cantidad de memoria posible y realizar nosotros nuestra placa con comunicación hacia la PC incluida.

La ventaja de esta alternativa sería su costo, aunque tendría grandes inconvenientes como por ejemplo que la memoria sería mucho mas reducida que la de la opción previamente tratada. Se agrega la dificultad de realizar ciertos circuitos electrónicos de mediana complejidad, por ejemplo para la comunicación con el exterior, lo que finalmente lleva a más tiempo y trabajo.

3) Utilizar varios microcontroladores individuales realizando tareas en paralelo. Esta opción posee las mismas desventajas que la segunda. Aunque tendría la ventaja comparado con esta de exigir menos a cada microcontrolador.

4.5. Elección de arquitectura y microcontrolador

Básicamente por los motivos ya expuestos, que ahora se detallaran nuevamente para que queden más claros para el lector:

- Es la opción más potente y pequeña dentro de las SBC.
- No requiere tanta electrónica adicional como las demás opciones.
- Ya posee comunicación wi-fi con PC externa; en casi todos los otros casos, hay que implementar la misma.
- No requiere compra de programadores, ni de software adicional.
- Cumple con todos los requisitos de funcionalidad en cuanto a entradas analógicas y salidas PWM sólo con la placa Robostix.
- No es mucho más caro que las demás opciones.

Resulta la mejor opción utilizar un SBC Gumstix, junto con la placa Robostix, dentro del pack de robótica de Gumstix. La arquitectura será similar a la segunda, mientras que en cuanto a los programas, será como la primera (será todo trabajado como una unidad). Esto se describirá con mayor detalle en las secciones posteriores).

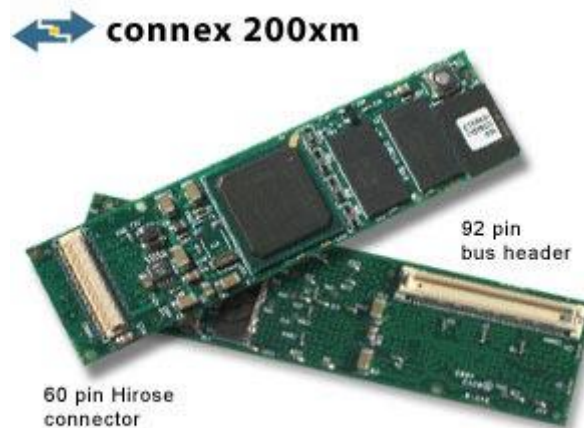


Figura 4.4. Motherboard Gumstix

5. Modelo Matemático del Sistema

5.1. Resumen

En esta sección se presentan la implementación, surgiendo desde las ecuaciones que gobiernan el problema (bajo ciertas hipótesis), de un simulador útil para comprender la mecánica de movimiento del pez. Se muestran los resultados obtenidos, y la realización de algoritmos sencillos de control para mejorar por ejemplo los tiempos de respuesta del sistema y seguir ciertas trayectorias, entre otros.

5.2. Modelo Utilizado

A diferencia de lo que es usual en los robots con vehículo mediante ruedas, los modelos cinemáticos son más complicados en el caso presente de un pez robótico, ya que debemos considerar la interacción entre ellos y su medio de locomoción, que en este caso es el agua.

Del estudio de la literatura especializada [23], encontramos el siguiente modelo simplificado para nuestro sistema:

$$ma_t = F_{ty} - D_v \quad (5.1)$$

$$V_r = 1.2f \sin \theta \quad (5.2)$$

Siendo θ el ángulo que forma el eje sobre el cual se dan las oscilaciones de la cola con el cuerpo del pez, m la masa del pez, y f la frecuencia de oscilación de la cola,

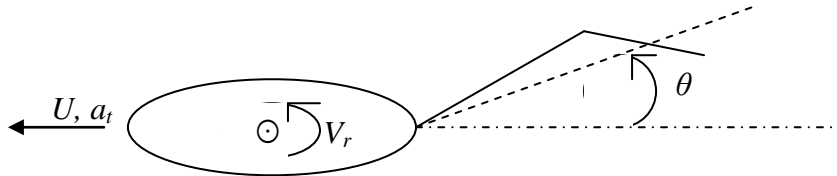


Figura 5.1. Esquema del pez

donde para modos caranguiforme o subcaranguiforme el modelo hidrodinámico relacionado con la generación de empuje se asocia con el método de la masa añadida: cuando una onda pasa por detrás del pez, el momento de ésta cambia por el movimiento de su cola, lo que causa una fuerza de reacción desde el agua al pez F_{thrust} . Esta fuerza a su vez se descompone en una componente lateral F_{tx} y una componente de empuje la cual contribuye al movimiento de proporción del pez $F_{ty} = F_{thrust} \cos \theta$ [24].

Se observa que a su vez hay una velocidad máxima, la cual ocurre cuando el rozamiento viscoso es igual al empuje, bajo la hipótesis de que el mismo no depende de la velocidad del pez. Se puede determinar una expresión para este, ya que esta velocidad depende de la frecuencia de oscilación de la cola, de las amplitudes máximas de las oscilaciones, y de la geometría del pez.

El parámetro D_v representa el rozamiento viscoso, que será la única fuerza que consideraremos oponiéndose al empuje. Este será determinado mediante la expresión estándar, la cual considera contribuciones por flujo laminar y turbulento:

$$D_v = 0.5C_f S U^2 \rho \quad (5.3)$$

donde S es la superficie de la cola en cuestión, ρ la densidad del agua, y C_f el coeficiente de rozamiento, el cual depende del número de Reynolds, que se define como:

$$Re = LU / \nu,$$

donde L es el largo de la cola y ν es la viscosidad cinética del agua, que tiene un valor tímicrocontroladoro de $1.12 \text{ mm}^2 / \text{s}$.

Los coeficientes de rozamiento por flujo laminar y turbulento (cuya suma da C_f), a su vez son: $1.328 Re^{-0.5}$ y $0.074 Re^{-0.2}$ respectivamente.

Finalmente, la velocidad a considerarse (U), corresponde a la velocidad en la dirección principal de desplazamiento, la cual cumple con la siguiente relación:

$$S_t = fA / U \quad (5.4)$$

donde S_t corresponde al número de Strouhal, y se encuentra entre 0.25 y 0.35, y A es la amplitud de las oscilaciones de la cola. Así es que se puede determinar F_{thrust} a partir del máximo rozamiento el cual ocurre a la máxima velocidad U_{max} , como:

$$F_{thrust} = 0.5C_{f \max} S U_{\max}^2 \rho \quad (5.5)$$

Es así que finalmente se escribe:

$$\dot{U} = F(U) \Big|_{f=cte, A=cte} \quad (5.6)$$

Este modelo es utilizado en el simulador realizado por el grupo de la universidad de Essex, el cual nosotros consideramos para realizar un simulador que nos permita ensayar los comandos de control a utilizar. Los resultados obtenidos en las primeras simulaciones con este simulador presentaron resultados muy similares a aquellas disponibles en la literatura, tanto para giros como para desplazamientos lineales [25] y [26]. Las velocidades máximas obtenidas y tiempos de respuesta también fueron muy similares, lo cual muestra que un simulador de esta clase se ajusta a un primer modelo de prueba con el cual trabajar.

La ecuación diferencial fue resuelta por el método de Euler hacia delante en nuestro simulador. Obtuvimos la respuesta al escalón del sistema bajo la frecuencia de oscilación de la cola.

Vale la pena observar que hay una cota superior para f dada por la velocidad máxima de los servomotores, que son los que actúan sobre las articulaciones.

Con respecto a la velocidad angular, ésta puede ser obtenida mediante la expresión comentada al principio que proviene de datos estadísticos, y es presentada en la

referencia [9.4]. Esta expresión es válida bajo la hipótesis de que la duración de la aceleración angular es muy pequeña como para considerarla.

Resumiendo, vemos que podemos actuar sobre la velocidad lineal y la velocidad angular de nuestro sistema alterando el ángulo del offset de la cola, la amplitud y frecuencia de oscilación de la misma.

5.3. Implementación en Matlab

Utilizando las ecuaciones anteriormente descriptas, procedimos a realizar nuestra propia versión del simulador del pez robot. Para ésto, definimos primeramente rutinas que integran por el método de Euler hacia adelante la ecuación diferencial (no lineal en U) [27]. Realizamos rutinas que llaman a esta función, a saber la rutina que implementa la respuesta al escalón con el pez partiendo del reposo, y aplicándosele un escalón unitario en f- esto conduce como consecuencia a una variación en U, la cual es graficada en el tiempo, y se muestra en la figura 9.2.

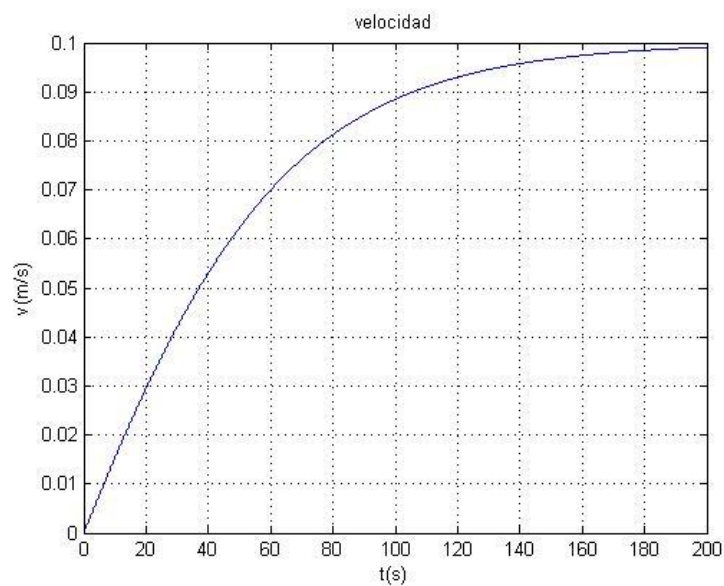
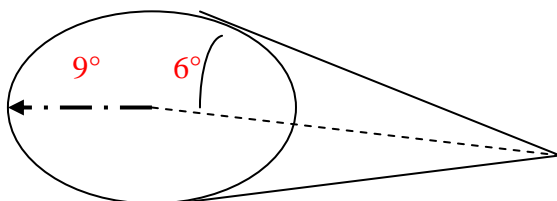


Figura 5.2. Respuesta al escalón unitario.

5.4. Modelado computacional de los sensores

Nuestra simulación agrega a lo expuesto un modelo de los sensores de distancia, fundamentales para la correcta evolución de su movimiento.

Con esta finalidad fue necesario determinar experimentalmente la región de sensibilidad de los mismos (zona del espacio donde el sensor detecta objetos). Como conclusión se ha de observar que el sensor detecta objetos en un ángulo menor a $\pm 9^\circ$ en su sentido horizontal, y $\pm 6^\circ$ en el sentido vertical.



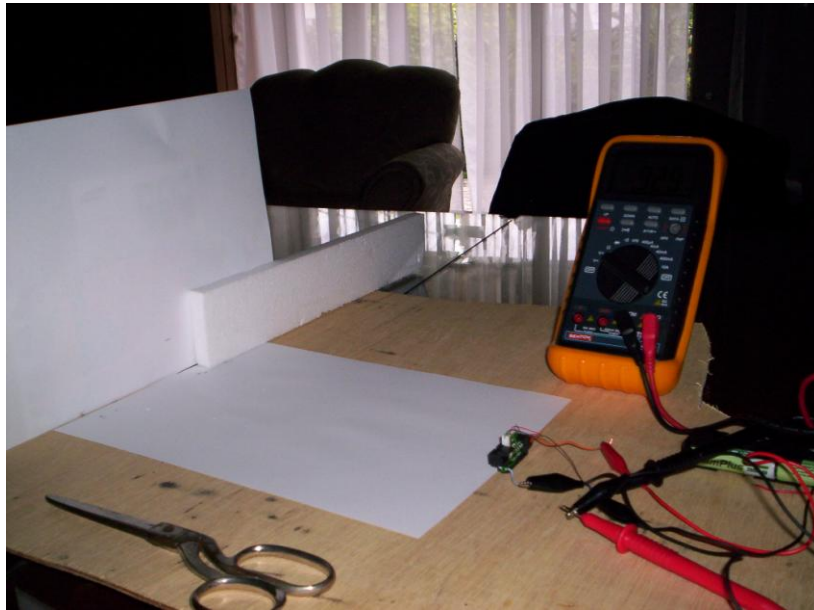


Figura 5.3. Medidas para la determinación del cono de sensibilidad (en la foto se muestra el sensor en el sentido horizontal)

El modelo de los sensores de distancia sólo detectarán objetos presentes dentro de este ángulo sólido de forma estricta, agregándose la limitación de alcance de una distancia de 60 centímetros (las hojas técnicas especifican con mayor optimismo un alcance de 80 centímetros, mas sencillas pruebas mostraron inferiores magnitudes).

Agregamos a su vez ruido gaussiano a estas medidas, con un parámetro de potencia variable, y representamos la característica no lineal de los mismos, haciendo que éste ruido afecte más a los valores mayores de distancia que a los menores.

5.5.Primer intento de control: Rutina para seguir la pared.

Realizamos un código que toma las lecturas de los sensores (cada cierto intervalo de tiempo), y a partir de las mismas (comparando con las anteriores), corrige el ángulo de la cola del pez, de modo de no chocar contra la pared, y seguir la misma a una distancia no mayor en lo posible a una elegida.

Para ésto, se realizó el siguiente algoritmo:

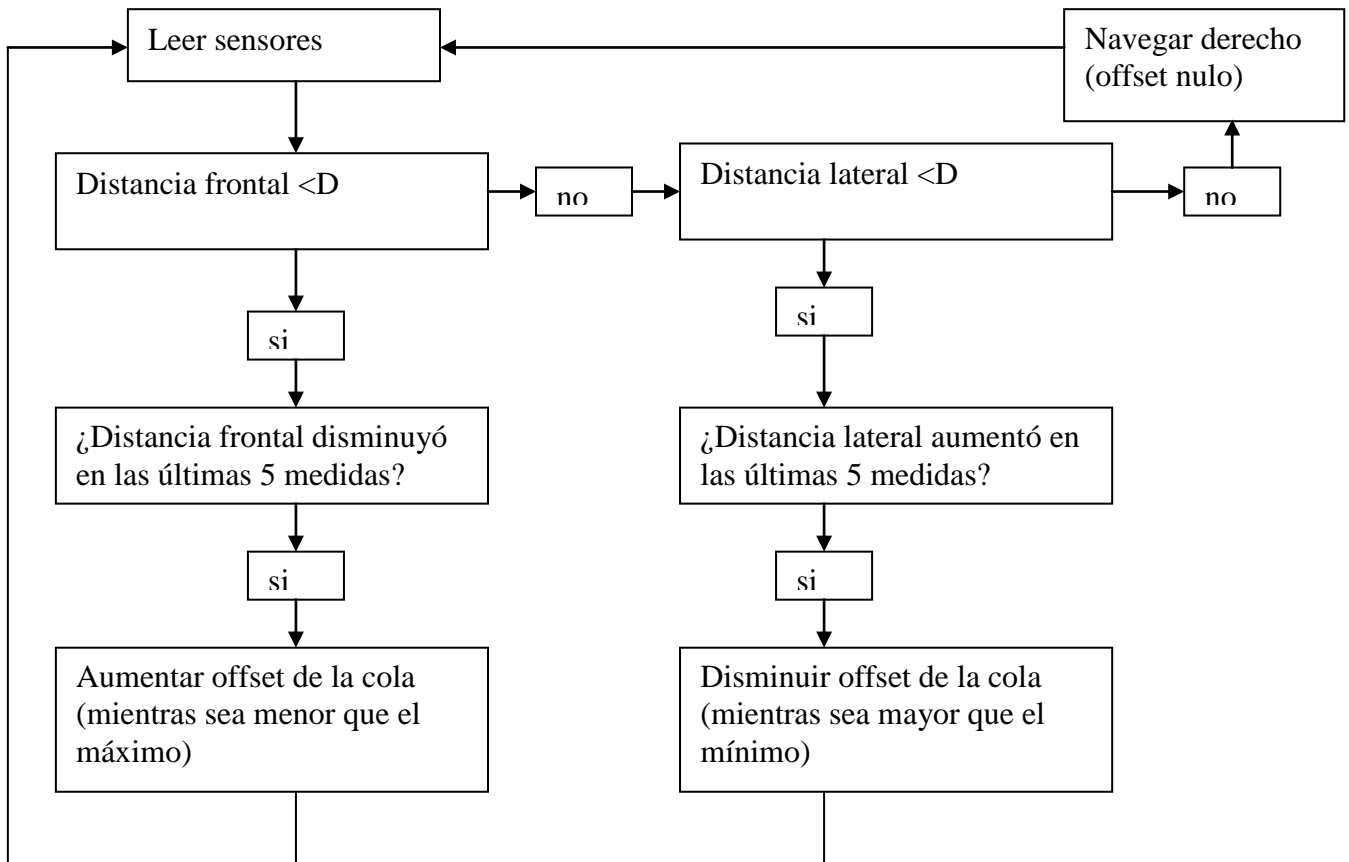


Figura 5.4. Esquema del algoritmo utilizado para navegar siguiendo a la pared.

Este algoritmo fue probado para diversos puntos de partida y velocidades (dadas por las frecuencias máximas de oscilación de la cola), observándose en todos los casos que el pez eludió efectivamente las paredes (para un tanque simulado de forma rectangular). A continuación se muestran los resultados obtenidos para un movimiento a una frecuencia de oscilación de la cola de 1Hz.

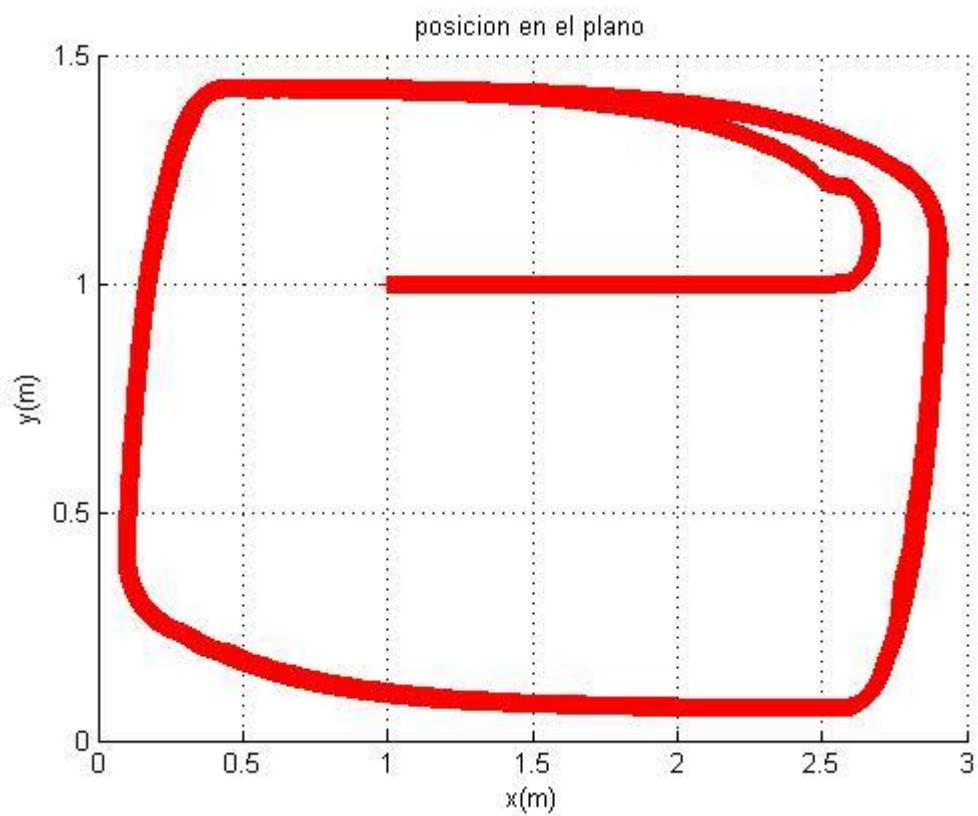


Figura 5.5. Trayectoria seguida por el pez en el simulador.

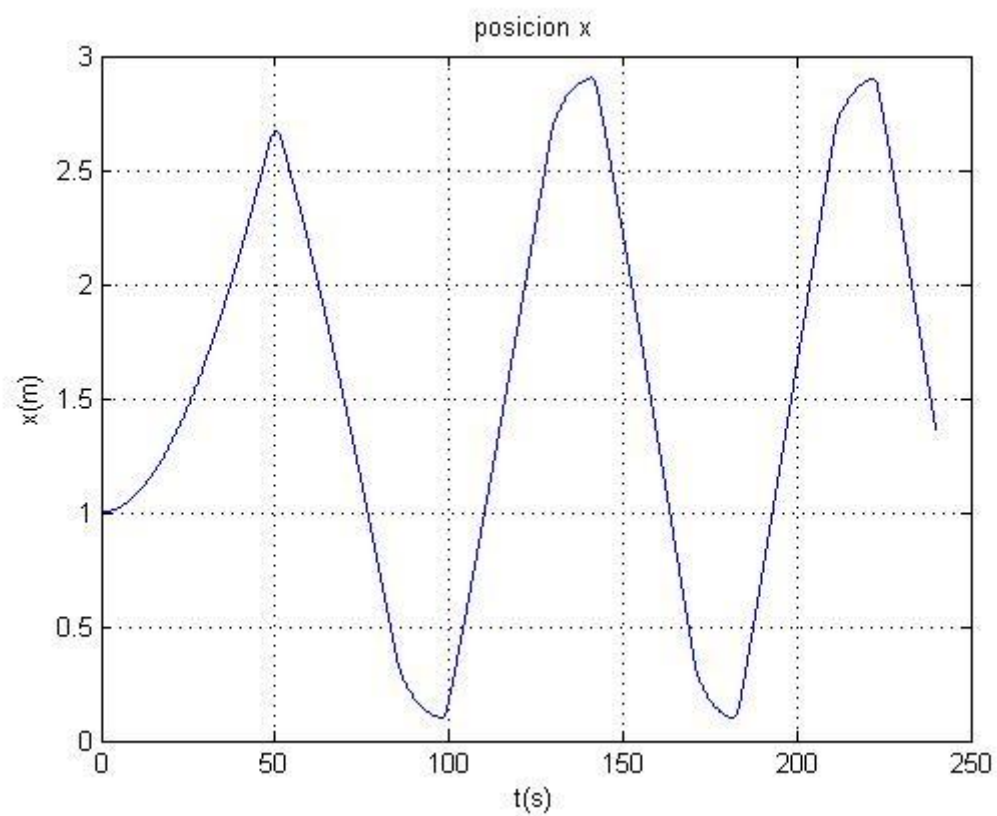


Figura 5.6. Coordenada x como función del tiempo.

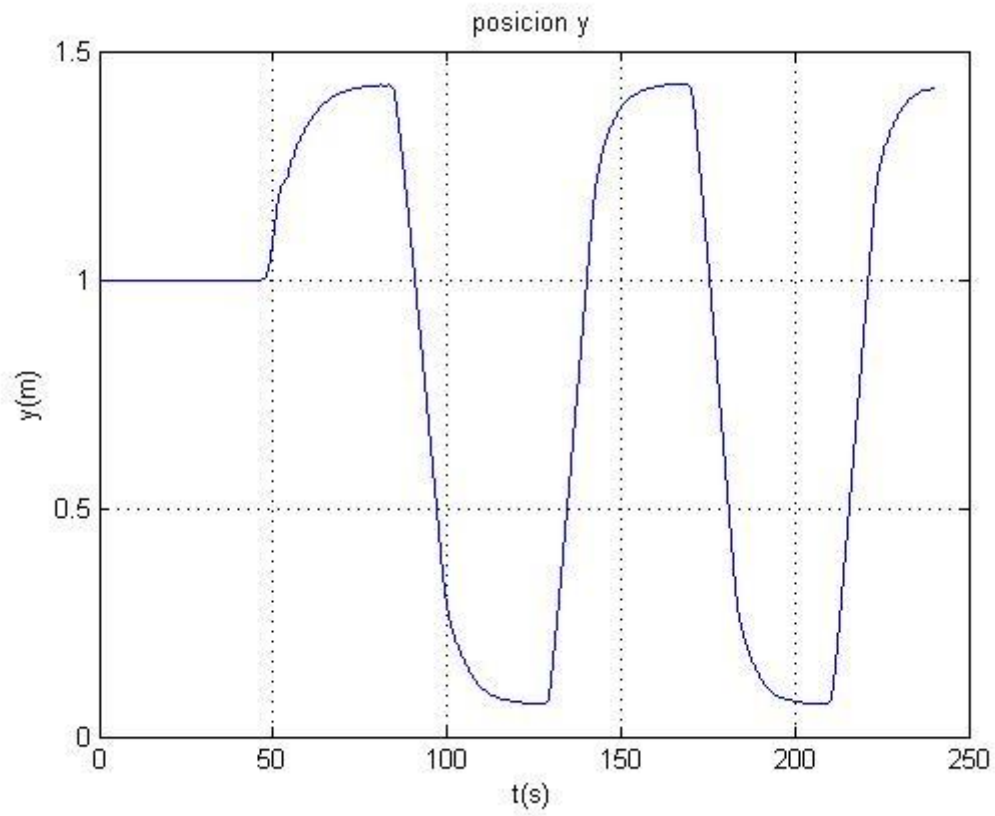


Figura 5.7. Coordenada y como función del tiempo.

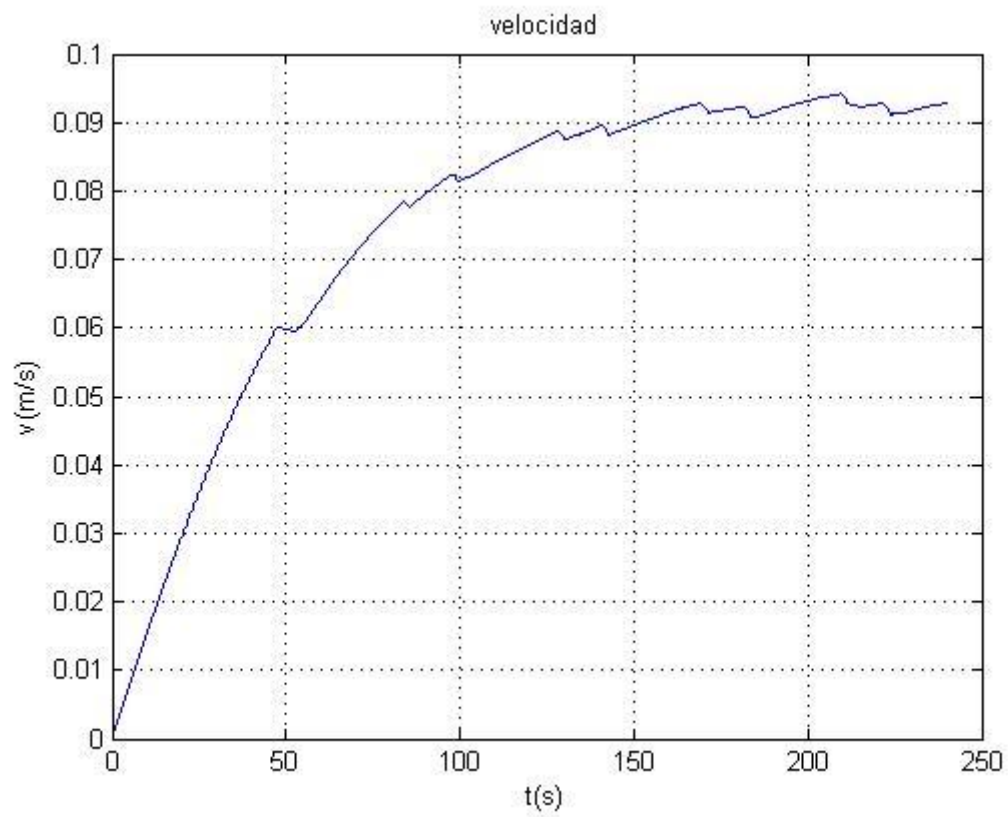


Figura 5.8. Velocidad en función del tiempo.

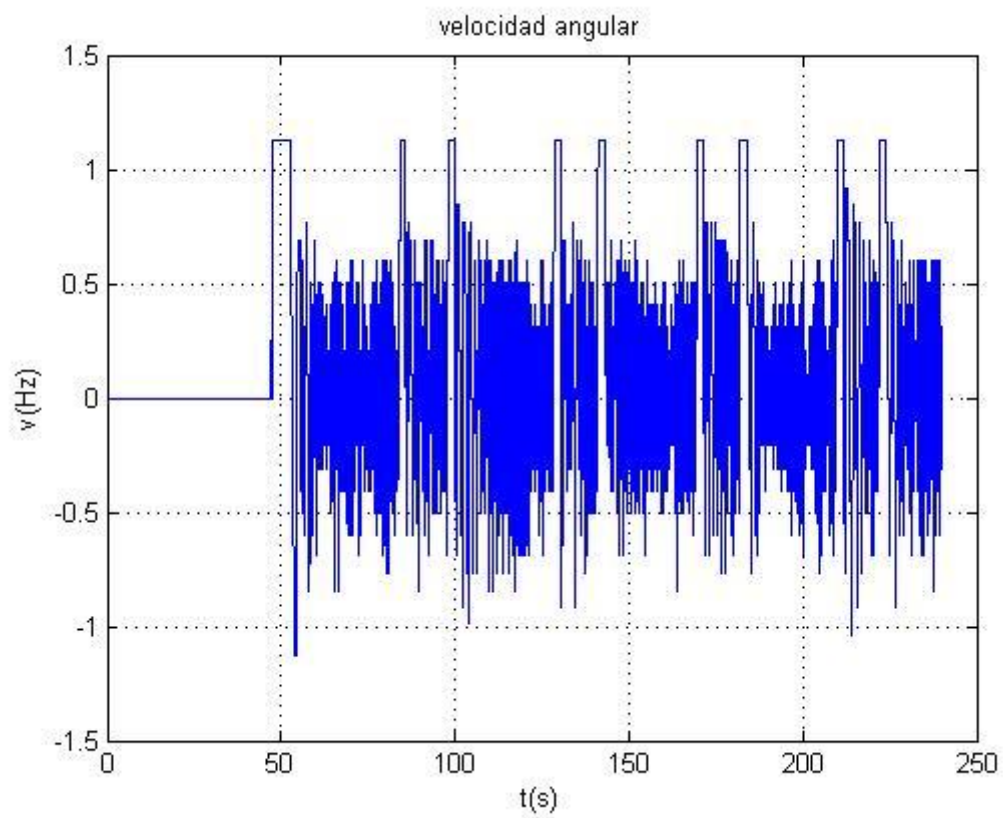


Figura 5.9. Velocidad angular en función del tiempo.

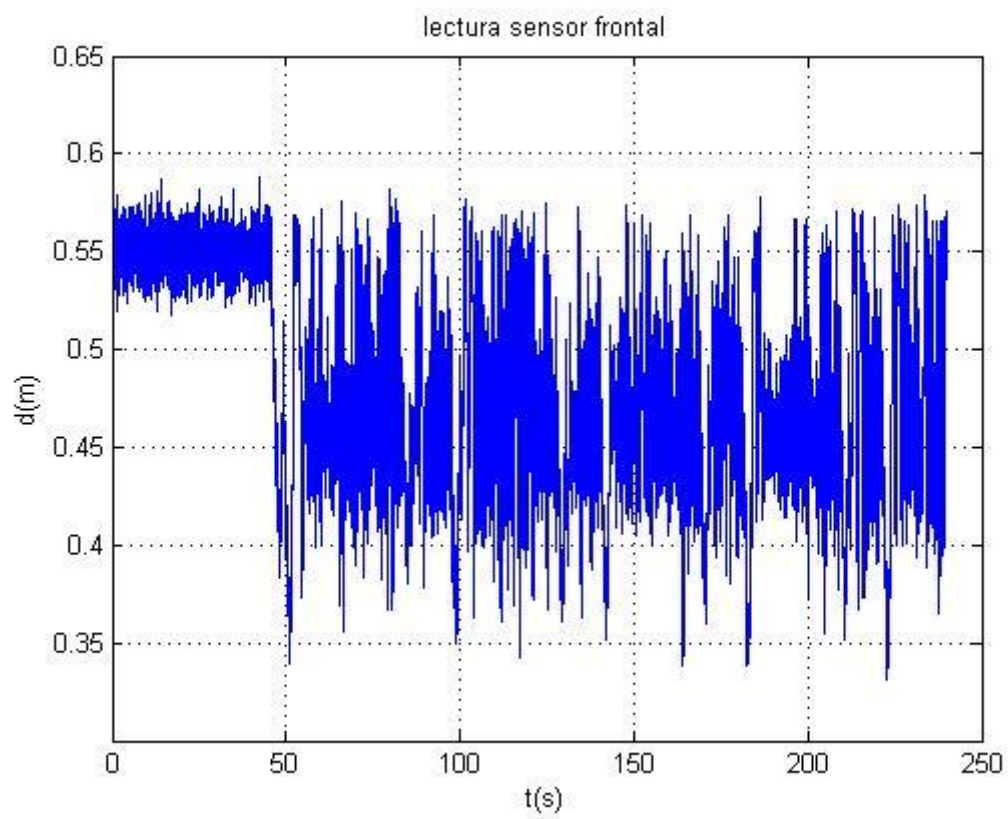


Figura 5.10. Lectura del sensor frontal en función del tiempo.

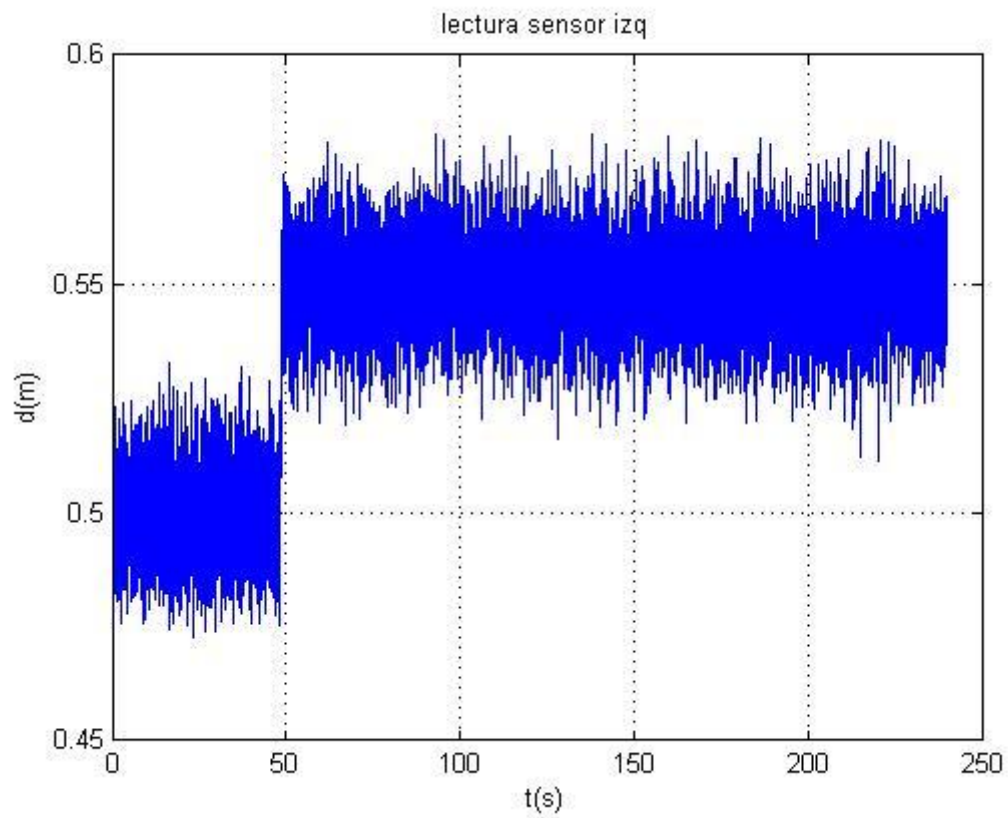


Figura 5.11. Lectura del sensor izq.

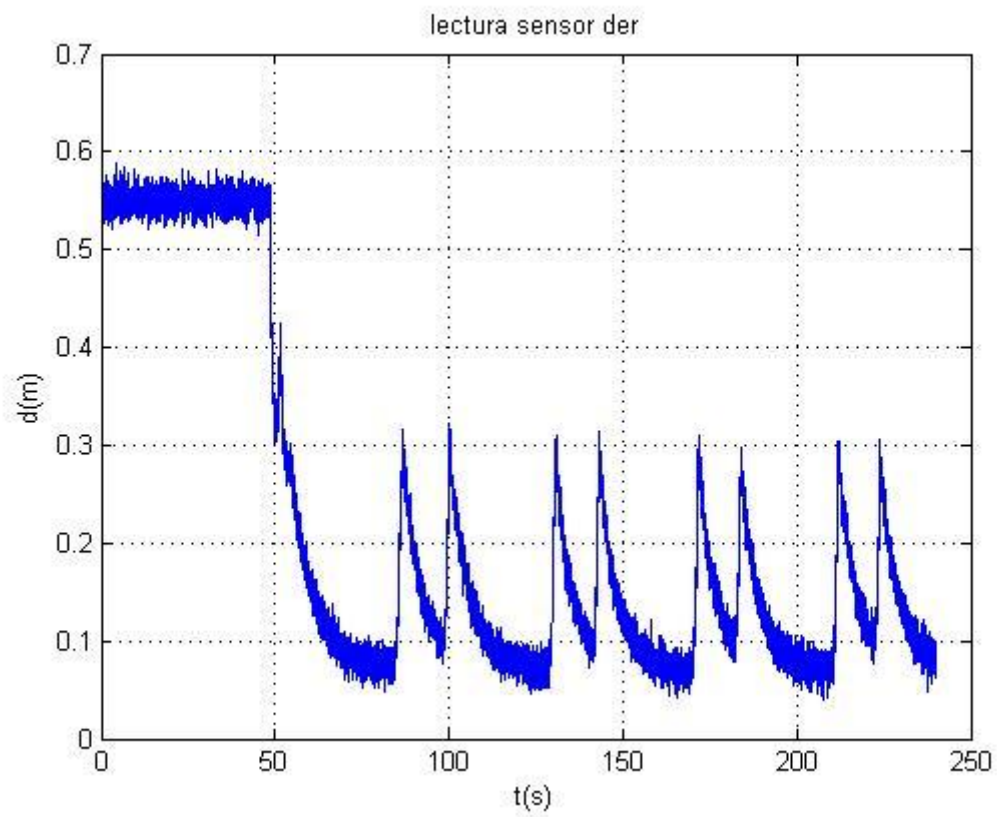


Figura 5.12. Lectura del sensor der.

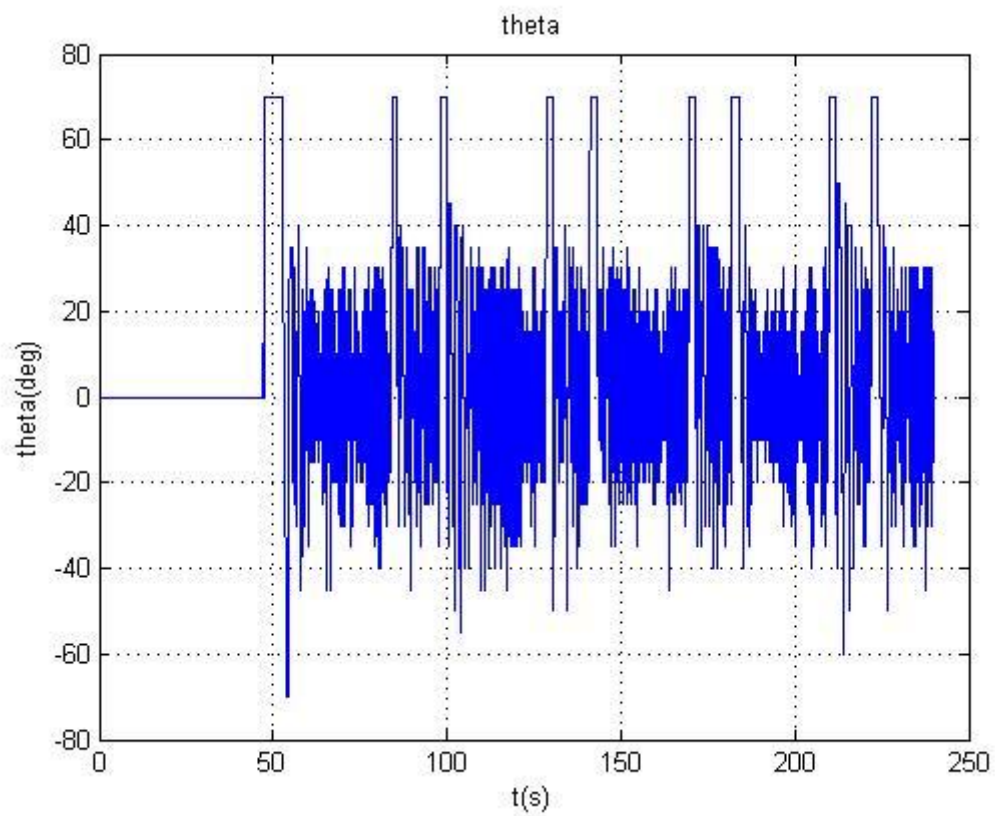


Figura 5.13. Angulo del eje de desfase de la cola en función del tiempo.

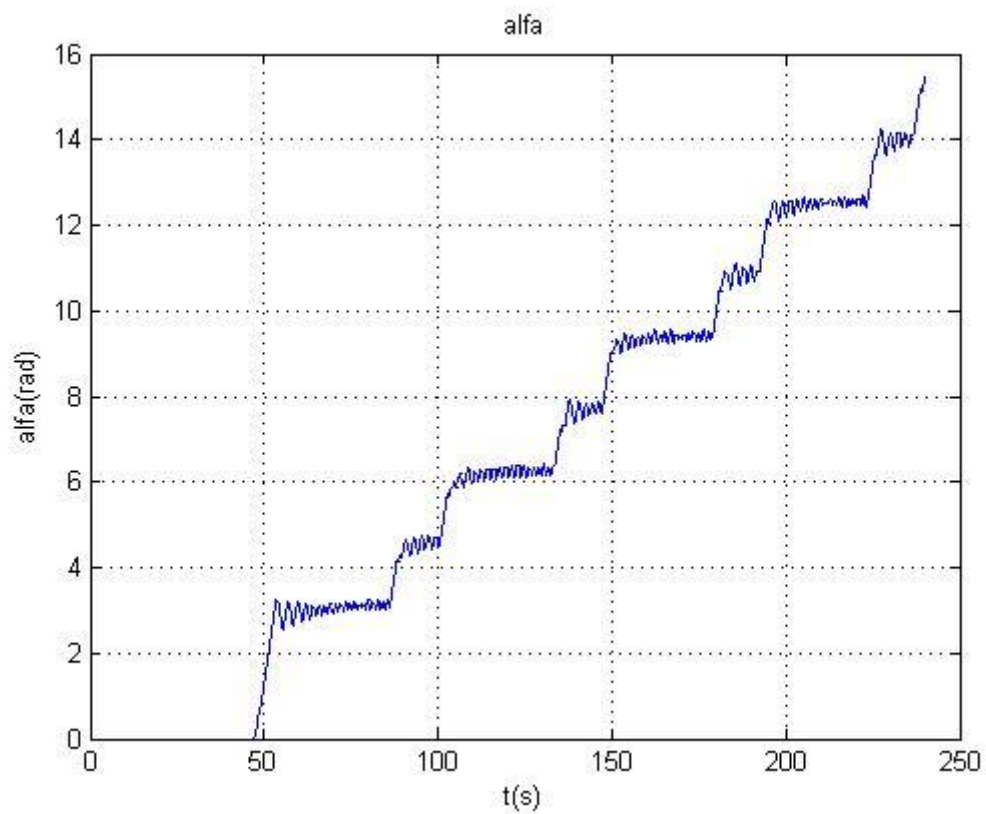


Figura 5.14. Orientación como función del tiempo.

6. Comportamientos del Pez

6.1.Introducción

Esta fase tiene por objeto definir distintos estados en los cuales el pez se puede hallar. En todo el proceso nos regimos por la conducta inherente a un pez en su hábitat natural, teniendo presente a su vez que la diversidad de misma causa un compromiso entre la dificultad de implementación y la similitud con la realidad.

Procedemos entonces fraccionando el problema de la planificación de la conducta del pez en cierta cantidad de tareas, las cuales van a depender estrechamente de distintos modos en los que el pez se puede hallar. Estos conceptos nos llevan a una solución, y son los que nos van a ocupar a continuación.

6.2.Movimientos Simples

Los movimientos simples son aquellos que se pueden generar de manera directa a partir del comando de los motores, y respuesta a los estímulos. Nos referimos entonces al andar derecho, girar tanto a la izquierda como la derecha, y también subir y bajar. Por último, la ausencia de movimiento también entra en esta categoría, como un caso especial (vale observar que este movimiento se caracteriza por velocidad nula en todos los motores, independientemente de la dirección del pez).

6.3.Atributos

Los atributos a los distintos movimientos son específicamente las amplitudes, fases y frecuencias de los motores, la dirección de nadado y velocidad lineal del pez. Como se verá posteriormente, esta última será, por practicidad, el atributo más utilizado para caracterizar los distintos comportamientos del pez.

6.4.Tareas

Las tareas son actividades limitadas en el tiempo que se definen a partir de la observación de las características generales del nadado de un pez biológico. Se busca entonces reunir cualidades básicas del movimiento de los peces de forma tal que en su conjunto determinen clásicos comportamientos de los mismos. Más específicamente, nuestro pez robótico va a lograr realizar 7 tareas posibles y disjuntas, a saber:

- Seguir la pared
- Descanso en la cercanía de un borde
- Ir y volver
- Nadar en un círculo
- Nadar sin un propósito
- Nadar chocándose contra la pared
- Encontrar borde

6.5. Sensores activos en distintas tareas

Surge con naturalidad una vez definidas las tareas, la pregunta de cómo realizarlas. Proponemos aquí una breve discusión de los sensores que se presentan activos en cada tarea, así como también criterios de duración de las mismas y uso de movimientos simples para cada una. Esto a su vez genera pautas para la programación del pez:

1) Seguir la pared

Esta tarea se va a producir mientras existan lecturas de los sensores de distancias. Existen claramente dos casos disjuntos: la pared se halla a la izquierda o bien a la derecha del pez (estamos descartando que el pez tenga paredes en su cercanía en ambos lados a la vez, algo que asociamos con “trampas”, que hasta el momento no se consideran). Luego, en un primer caso el sensor delantero y el izquierdo captan el estímulo en cuestión; el derecho junto con el central en el segundo caso.

Los movimientos simples considerados para efectuar correctamente esta tarea son el nadar derecho, cambiar de nivel y girar. El giro está presente no sólo para cambios bruscos en el ángulo de la pared, sino que es imprescindible inclusive ante paredes planas también, pues la oscilación de la cola plantea leves alejamientos y acercamientos, que se corrigen con pequeños giros. El cambio de nivel brinda libertad en el mayor libertad al robot en el seguimiento, y requiere actividad del sensor de presión (aunque se puede prescindir de este movimiento para paredes simples).

Por último, el fin de esta tarea va a estar dado si se presenta alguna falla, como es cuando el sensor lateral activo deja de notar pared, y se pierde en consecuencia la misma, o bien cuando expira el tiempo del estado asociado a esta tarea, que como veremos más adelante, no vamos a permitir que una tarea sea indefinida en el tiempo.

2) Descanso contra un borde

Aparenta ser un hábito frecuente en los peces esta tarea, que como su nombre lo indica es bien pasiva. Sin embargo, no se debe perder la actividad de lectura de los sensores, dado que no queremos que durante el descanso el pez reciba un impacto contra el borde. Naturalmente, esta tarea viene de la mano de la búsqueda del borde, algo que se debe efectuar previamente (este condicionamiento se cuantificará mediante la definición de probabilidades de transición más adelante).

Para caracterizar la actividad de los sensores durante esta tarea, se debe buscar reducir la velocidad al mínimo, en el caso ideal con completamente quieto, y también como recién se mencionó, sensar la distancia con los bordes, cuya condición de constante es condición necesaria mas no suficiente para la quietud.

La terminación de esta tarea va a indicar que el pez “ya descansó”, y se va a asociar con una variable aleatoria exponencial.

3) Ir y volver

También es una tarea que se repite en muchas especies de peces. Para la implementación de esta tarea resulta crítico el retorno, dado que es preciso aquí cambios de 180 grados en la dirección, requiriendo reducciones en la velocidad para lograr tal objetivo. Vamos a admitir también cambios de nivel, lo que agrega la acción del mecanismo de profundidad y lecturas del sensor de presión.

Se va a dar una transición a otra tarea cuando exista un error, como puede ser el choque al percibir cercanía a una pared e intentar realizar el giro de 180 grados erráticamente, o bien ante la expiración del tiempo de esta tarea.

4) Nadar en círculos

Esta tarea tiene muchas limitaciones para su aplicación. La primera es que el radio de curvatura de la trayectoria a recorrer debe estar limitado por la geometría del espacio de navegación y se reduce con la proximidad del centro de la trayectoria con la vecindad de la frontera de este espacio. Se busca entonces imprimir un sesgo constante en la cola, siempre que no se detecten bordes, en cuyo caso se impone el cambio de tarea. Otra limitación es la dificultad de la estabilidad, para no progresar en trayectorias de espiral. Es evidente que se requerirá de la actividad de los sensores de distancia para el constante monitoreo de presencia de bordes. También se desea que el pez pueda cambiar de radios con el tiempo durante esta tarea, lo que agrega la necesidad de realización de giros con cambios simultáneos en la velocidad.

Por último, la transición a otra tarea se va a dar ante la cercanía con la pared o bien luego de un tiempo aleatorio.

5) Nadar sin un propósito

En esta tarea deben permanecer activos los sensores de distancia en todo momento, para esquivar bordes, como también el sistema de profundidad. Se aplican todos los movimientos simples, por lo que el pez aquí puede nadar derecho, girar, cambiar de nivel o también no moverse.

La finalización de esta tarea es en todos los casos, mediante expiración de un tiempo variable.

6) Nadar chocándose contra la pared

¿Por qué definir una tarea tal? Vale destacar que el pez va a realizar distintas tareas con distintas probabilidades, por lo que vamos a impedir que el pez realice muy seguido esta tarea. Esta situación del pez se caracteriza por la inactividad de los sensores de distancia, y como consecuencia, una movilidad errática.

Su final se define mediante una variable aleatoria de tiempo.

7) Encontrar borde

Esta última tarea resulta ser muy importante, pues recordemos que el pez no descansa sin antes hallar un borde, tampoco va a seguir la pared sin encontrarla previamente. Se caracteriza por el hecho de seguir una trayectoria rectilínea en busca de la pared. Aunque el acercarse a la misma precisaría en otros dispositivos veloces de la actuación sobre la velocidad, las reducidas magnitudes de la misma en este robot no muestra que esta sea crítica.

Todos los sensores deben estar activos para encontrar bordes. Por último, esta es la única tarea que finaliza en forma determinística: una vez hallado el borde, realizamos la transición a otra tarea.

6.6.Modos

Las tareas no las va a efectuar el pez de la misma manera a través del tiempo, sino que se van a pronunciar diferencias en sus atributos como son la velocidad y cambios en la dirección de acuerdo con estados de ánimo, o modos en los que se encuentra el pez, los cuales van variando con el tiempo, y buscan imprimir carácter propio al pez y expresión de las emociones.

Es en esta definición de modos que presenta el pez es donde se halla el principal desafío o compromiso entre lo verosímil del movimiento de nuestro robot en comparación con lo real, y de la dificultad de implementación. Aquí señalaremos brevemente dos posibles enfoques, y optaremos por aquél que sea el más prometedor y simple de aplicación.

6.6.1. Primer enfoque

Este enfoque plantea una diversidad de estados de ánimo, y define el concepto de estados a partir de pares tareas-ánimo. Renuncia a las tareas de ir y volver, como también a la de descanso en un borde, por lo que trabaja con 5 tareas en total. Los estados de ánimo son los siguientes:

- Animo de jugar: la dificultad de este ánimo radica en qué tipo de tareas asignarle al pez para que exteriormente se aprecie su juego. La más apropiada de las definidas aparenta ser nadar sin un propósito con variaciones en la velocidad, en la que la probabilidad de recorrer trayectorias circulares es condicionada favorablemente cuando el pez juega.
- Animo de búsqueda de presa, “acelerado”: el pez se debe hallar en este estado de ánimo únicamente cuando desea saciar su apetito, algo que ocurre pocas veces en un día. Naturalmente, la probabilidad de seguir la presa está condicionada a la existencia de la misma en las proximidades del pez.

Por lo anteriormente mencionado, este estado debe poseer baja probabilidad.

- Animo de conquista: se asocia este estado con el período de celo, en el que puede permanecer días con alta probabilidad una vez que se entra en este estado (algo que se puede tratar con un valor esperado mayor en el tiempo de transición de estado). Una vez que sacia su apetito sexual, comúnmente pasa a un estado de reposo, o en algunas ocasiones, de hambre.
- “Alocado”: en este estado el pez puede llegar a dar vueltas muy rápidamente, al punto de perder la estabilidad. Condiciona favorablemente a proseguir con un descanso.
- Luego de comer, “satisfecho”: este estado viene acompañado usualmente de tareas de seguir la pared en paralelo y de descanso.
- Animo de sueño: los peces duermen a pesar de no tener párpados. Normalmente reducen su actividad durante la noche, y al dormir se colocan en una esquina del acuario muy quietos. Este estado de ánimo está condicionado favorablemente luego de comer y durante la noche, cada 24 horas.

Dado que tenemos 5 tareas y 6 estados de ánimo diferentes, existe un total de 30 estados definibles, aunque hay eventos que son incompatibles, como dormir y jugar, y otros. Se reduce la complejidad asignando probabilidad no nula a la mitad de los estados, a saber:

e_1 : descansar satisfecho
 e_2 : descansar con ánimo de sueño
 e_3 : seguir en paralelo la pared con ánimo de búsqueda de presa
 e_4 : seguir en paralelo la pared con ánimo de conquista
 e_5 : seguir en paralelo la pared “alocadamente”
 e_6 : seguir en paralelo la pared, satisfecho
 e_7 : buscar la pared con ánimo de jugar
 e_8 : buscar la pared “alocadamente”
 e_9 : recorrer trayectorias circulares con ánimo de jugar
 e_{10} : recorrer trayectorias circulares con ánimo de conquista
 e_{11} : recorrer trayectorias circulares “alocadamente”
 e_{12} : nadar sin propósito, con ánimo de jugar
 e_{13} : nadar sin propósito, con ánimo de búsqueda de presa
 e_{14} : nadar sin propósito, con ánimo de conquista
 e_{15} : nadar sin propósito y alocado

La probabilidad de hallarse en un estado depende únicamente y por simplicidad del estado anterior. Definiendo estas probabilidades de transición se obtiene así una cadena de Markov. Se necesita también definir las variables aleatorias del tiempo de permanencia de cada estado.

6.6.2. Segundo enfoque

Aquí se plantean menos modos que en el enfoque anterior. A pesar de no resultar exhaustivo en la presentación del carácter del pez, muestra 3 modos marcadamente diferentes, que son de mayor simplicidad a la hora de la programación y posterior implementación de estados.

Los tres modos definidos son:

- Buscar presa
- Descansar
- Modo de exhibición

La idea es que el primer modo tenga asociado el atributo de velocidad alta, el descanso evidentemente bajo y aplique velocidades medias en el modo de exhibición.

Además, y lo que lo diferencia del enfoque anterior, es que a modos distintos se definen distribuciones de probabilidad distintas a las tareas. A su vez, se definen probabilidades de transición entre los modos, que imprimen al pez un dinamismo diferente.

Esto implica que se va a definir cadenas de Markov en dos niveles: por un lado, definir una cadena para los modos; además cada modo tiene asociado una cadena de Markov para las tareas. Por ejemplo, con el pez en el modo de descanso, va a ser más probable la tarea de encontrar borde y descansar en el borde que otras; ahora si el pez está en modo de exhibición va a ser más probable la tarea de nadar siguiendo la pared que la de descansar.

Cada estado va a estar determinado por los dos niveles, es decir, por un modo y una tarea. A partir de lo anterior debe quedar claro que, cada estado está determinado únicamente dados un modo y una tarea, aunque no definamos probabilidades de

transición entre estados, sino que por simplicidad hemos definido probabilidades de transición entre modos, y entre tareas.

6.6.3. Comparación de los enfoques

- Fidelidad con la realidad

Dada la diversidad de estados de ánimo y el número de estados, el primer enfoque se muestra más fiel a la realidad. De todas maneras, con sólo 3 modos, el segundo enfoque permite ofrecer suficiente variabilidad en el dinamismo, disociando claramente los modos rápidos de los lentos.

- Complejidad

El primer enfoque es claramente más complejo. Observar que el número de estados en este enfoque es de 15, lo cual hace un total de $\binom{15}{2} 2! = 210$ probabilidades de transición

a definir. Por el otro lado, en el segundo enfoque se definen 3 modos, y una cadena de Markov entre ellos debe definir 6 probabilidades. Para cada uno de los 3 modos, se tienen 7 tareas, que precisa de $\binom{7}{2} 2! = 42$ probabilidades de transición, totalizando

$3 \cdot 42 + 6 = 132$ definiciones. Este número es menor al primero, sin considerar el hecho que hay tareas que se pueden asociar con probabilidad nula a ciertos modos.

El hecho de asignar modos con cambios en los atributos simplifica mucho la programación, ya que es fácil reutilizar los programas cambiando no mucho más que sus valores.

- Claridad de la solución

En el segundo enfoque se distinguen claramente dos niveles, que son modos y tareas. La programación se puede modular en tareas, y los atributos, incluyendo aquí la velocidad media y las probabilidades, se pueden redefinir para cada uno de los tres modos.

Dado que en el primer enfoque no se puede pensar en que los estados de ánimo subyacen las tareas o viceversa, la estructuración de la solución pragmática no es clara.

Por estas razones consideraremos de aquí en más el segundo enfoque, procediendo a definir las probabilidades de transición y las tareas posibles para cada modo.

6.7. Plan de distinción de los estados de ánimo

El estado de ánimo en el que se encuentre el pez actuará modificando las probabilidades de transición entre las tareas dentro de un modo y a su vez cambiando las velocidades de ejecución de las tareas. En un instante determinado de tiempo, el pez se encontrará en un modo, y con un estado de ánimo, el cual alterará las variables ya mencionadas.

6.8. Análisis Temporal

El tiempo lo vamos a modelar discreto, de forma tal que planteamos o no transición a una tarea o permanencia en la misma regularmente con la finalización de cada ranura.

Para la selección de las ranuras de tiempo nos basamos en la respuesta mecánica en el tiempo del pez, que anticipamos, es en el orden de segundos. Se selecciona entonces una ranura $T_{st}=2s$ para los cambios en las tareas. Puesto que no deseamos cambios bruscos una vez que exista transición entre modos, vamos a tomar la unidad de ranura de tiempo para la transición de los modos como un múltiplo de T_{st} ; por el momento consideremos $T_{sm}=10 \cdot T_{st}$. Vale destacar que estos parámetros se pueden modificar a medida que se realicen ensayos.

6.9. Asignación de probabilidades de transición

6.9.1. De los Modos

Observar que la probabilidad de que exista transición a un modo distinto en el instante $t = k$ (medido en time slots o ranuras) coincide con la de permanencia consecutiva por $k-1$ ranuras de tiempo, y cambio en la ranura k . Esto se modela como una distribución geométrica de parámetro p_i , donde el índice $i: 1 \leq i \leq 3$ indica el modo (buscar presa, descansar o exhibición, respectivamente).

Para un modo dado, si queremos que permanezca en el mismo una media de x time slots (donde x es un entero), recordemos que la esperanza de la distribución geométrica es:

$$x = \sum_{k=1}^{\infty} kp(1-p)^{k-1} = -p \frac{\partial}{\partial p} \sum_{k=1}^{\infty} (1-p)^k = -p \cdot \left(\frac{-1}{p^2} \right) = \frac{1}{p}$$

por lo que basta con asignar una probabilidad de transición inversa al valor esperado de ranuras, o al tiempo medio deseado de time slots de permanencia del pez en el modo en cuestión.

Demos igual permanencia media a los modos de exhibición y de descanso, asignándoles $4T_{sm}$ al caso de búsqueda de una presa y $2T_{sm}$ de media a la búsqueda de una presa. (observar que ésto no significa que descanse más tiempo que el que dedica a buscar presa, sino que una vez entrado a este modo toma en media más tiempo en cambiar a otro). Luego, las probabilidades de transición son a partir del valor esperado de una geométrica, los siguientes: $p_{11} = \frac{1}{2}$, $p_{22} = p_{33} = \frac{3}{4}$.

Modo 1: Búsqueda de presa

Modo 2: Descanso

Modo 3: Exhibición

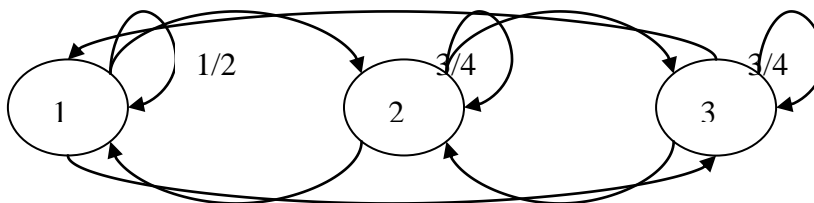


Figura 10.1. Probabilidades de Transición

Definamos p_{ij} como la probabilidad de que en la ranura de tiempo inmediata el pez se halle en el modo j , estando actualmente en el modo i . Hasta ahora asignamos valores a los lazos de permanencia de los modos p_{ii} , $i=1,2,3$.

Una ligadura inherente a las probabilidades es que a partir de un modo dado, se continúa en otro (o el mismo) con probabilidad 1; esto significa que:

$$\sum_{j=1}^3 p_{ij} = 1 \quad \forall i = 1..3.$$

Aún tenemos 3 grados de libertad. Para fijarlos, damos prioridad temporal al estado de exhibición, y luego al descanso, con ciertas condiciones.

- Consideración de tareas a distintos modos

Para definir valores a las probabilidades, vamos a dar prioridad a las siguientes tareas, para cada comportamiento:

Modo	Tarea
1) Buscar presa	Encontrar borde Seguir el borde Nadar chocando contra bordes Ir y volver Nadar sin un propósito
2) Descansar	Descansar en un borde Nadar en círculos Encontrar la pared
3) Exhibición	Ir y volver Seguir el borde Encontrar el borde Nadar sin un propósito

6.10. Programación

Vamos a aplicar el concepto de modularización realizando un programa para cada una de las tareas. Para comenzar y por razones de simplicidad, prescindiremos de cambios en la profundidad como también del cambio dinámico de los atributos (que vale recordar, es lo que intenta imprimir distintos estados de ánimo al pez).

Realicemos primeramente siete pseudocódigos titulados con cada una de las tareas. La idea es que en una etapa posterior, se realizará un programa principal que retenga el modo actual, y alterne entre los valores 1,2 y 3 con el tiempo, según las probabilidades dadas en la cadena de Markov de los modos definida anteriormente. A su vez, se bifurquen casos según otra variable entera que contenga un entero entre 1 y 7, y llame en tiempos correspondientes (dependiendo de la tarea actual) a las distintas tareas, que se deberán sortear de acuerdo con la distribución de probabilidad que posee la tarea en el modo dado.

Procedemos realizando primeramente siete pseudocódigos titulados con cada una de las tareas. Posteriormente daremos nombres a las distintas funciones que se requieren definir para ser usadas en el programa principal. Por último, mostraremos el código del programa principal, el cual llama a funciones de las distintas tareas según sorteos en tiempos regulares.

Tarea 1: Seguir el borde

Como se mencionó anteriormente, en el caso de tener tanto distancias lateral como frontal inferiores a D, podemos estar en el caso que se haya perdido el borde, momento en el cual se debe llamar a una función auxiliar que lo encuentre nuevamente.

En pseudocódigo, quedaría así:

Leer sensores

```
Si Dfrontal última < D entonces
{ Si Dfrontal disminuyó en las 5 últimas medidas
Mientras el offset de la cola sea menor que el máximo
( aumentar offset de la cola )
Volver a leer sensores }
```

```
En otro caso
Si Dlateral < D entonces
{ Si Dlateral aumentó en las 5 últimas medidas
Mientras el offset de la cola sea mayor que el mínimo
( disminuir offset de la cola )
Volver a leer sensores }
```

```
En otro caso // Dfrontal >= D y Dlateral >= D; entra a transitorio
{ Mientras Dfrontal >= D
( navegar derecho // con offset nulo
Leer Dfrontal )
Volver a leer sensores }
```

Tarea 2: Descansar contra un borde

El primer tramo de esta tarea va a consistir de hallar un borde. En lo sucesivo va a avanzar paralelo al mismo igual que en la tarea anterior, pero reduciendo paulatinamente la velocidad (o equivalentemente la frecuencia de oscilación).

El pseudocódigo tiene similitudes con el anterior, y es el siguiente:

Leer sensores

```
Si Dfrontal última < D entonces
{ Si Dfrontal disminuyó en las 5 últimas medidas
Mientras el offset de la cola sea menor que el máximo
( aumentar offset de la cola
disminuir f-de-oscilación )
Volver a leer sensores }
```

```
En otro caso
Si Dlateral < D entonces
{ Si Dlateral aumentó en las 5 últimas medidas
Mientras el offset de la cola sea mayor que el mínimo
( disminuir offset de la cola
disminuir f-de-oscilación )
Volver a leer sensores }
```

```
En otro caso // Dfrontal >= D y Dlateral >= D búsqueda de borde
{ Mientras Dfrontal >= D
( navegar derecho // con offset nulo
Leer Dfrontal )
Volver a leer sensores }
```

Tarea 3: Ir y Volver

La dificultad aquí se halla en el reconocimiento de los 180 grados al volver. Por el momento daremos una cierta cantidad de giro x, que será hallada en forma óptima mediante simulación.

Consideremos el siguiente pseudocódigo:

Leer sensores

```
Si Dfrontal > D' entonces
( navegar derecho
Si f > fmedia entonces
disminuir f
En otro caso
aumentar f
Volver a leer sensores
```

```
En otro caso
Si Dizq < Dmáx //debemos rotar a la derecha "una cantidad x"
{ Durante x iteraciones //ciclo for
(girar derecha // offset de cola a la izquierda
reducir f )
```

```

Volver a leer sensores }

Si Dder<Dmáx // caso análogo al anterior
{ Durante x iteraciones
  (girar izquierda
   reducir f )
Volver a leer sensores }

Si Dizq=Dmáx y Dder=Dmáx // el pez está por chocar derecho
{ elegimos izq. O der. Al azar
  Durante x iteraciones
  (girar el sentido obtenido al azar
   reducir f )
Volver a leer sensores }

```

Observaciones:

- El pseudocódigo tiene 3 parámetros variables: D', fmedia y x
- Dmáx no es parámetro, pues es la distancia máxima que llega a reconocer los sensores
- Si el doble del radio de curvatura de giro es inferior que Dmáx, el pez no va a chocar, salvo que entre en una “trampa”, definida mediante la condición siguiente: Dfrontal<D' y Dlaterales<Dmáx. En este caso la única manera de salir es mediante retroceso, que sería posible con aletas, o bien chocando de la manera más delicada posible y tomar un ángulo de salida.

Tarea 4: Nadar en círculos

Se anticipa aquí el problema de detectar un borde al describir un arco de circunferencia. Es evidente que el radio a describir depende de la geometría de la región de nadado y también de las limitaciones físicas del pez (tamaño, velocidad angular del motor, etc.), por lo que no es posible aquí un tratado independiente de la región. El algoritmo consta de giros simples que permiten realizar arcos o circunferencias en el mejor de los casos, y en caso de hallar borde se aleja del mismo y vuelve a describir otros arcos o circunferencias.

Este algoritmo se resume en el siguiente pseudocódigo:

```

Sortear izquierda o derecha
Guardar sorteo en variable x
Leer sensores
Si Dizq=Dmáx y Dder=Dmáx //estamos lejos de bordes laterales
{ Realizar offset de la cola según sorteo
  Si f>fmedia entonces
    disminuir f
  En otro caso
    aumentar f
  Volver a leer sensores
}

En otro caso // Estamos cerca de un lateral
Si Dizq<Dmáx //
{ girar derecha // offset de cola a la izquierda

```

```

        Asignar "derecha" a la variable x
        Si  $f > f_{media}$  entonces
            disminuir  $f$ 
        En otro caso
            aumentar  $f$ 
        Volver a leer sensores
    }

    Si  $D_{der} < D_{m\acute{a}x}$  // caso análogo al anterior
    { girar izquierda // offset de cola a la derecha
        Asignar "izquierda" a la variable x
        Si  $f > f_{media}$  entonces
            disminuir  $f$ 
        En otro caso
            aumentar  $f$ 
        Volver a leer sensores
    }

```

Tarea 5: Nadar sin un propósito

Como ya se ha mencionado, todos los sensores deben permanecer activos, con la primera función de evitar choques (y más adelante variar la profundidad). Se procederá alternando la aplicación de movimientos simples (navegar derecho, girar a ambos laterales), con libertad salvo ante la presencia de obstáculos, en los que se realizará un giro apropiado.

En pseudocódigo, quedaría así:

```

        Leer sensores

        Si  $D_{frontal} > D'$  entonces
            Si  $D_{izq} > D_{m\acute{a}x}$  y  $D_{der} > D_{m\acute{a}x}$ 
                ( sortear frecuencia
                sortear giro //incluye navegar derecho
                volver a leer sensores )
            En otro caso
                Si  $D_{lateral} < D$  entonces
                    { Si  $D_{lateral}$  aumentó en las 5 últimas medidas
                    Mientras el offset de la cola sea mayor que el mínimo
                        ( disminuir offset de la cola )
                    Volver a leer sensores }

        En otro caso //si la distancia lateral es pequeña
        Si  $D_{izq} < D_{m\acute{a}x}$  //debemos rotar a la derecha "una cantidad x"
            { Durante x iteraciones //ciclo for
            (girar derecha // offset de cola a la izquierda
            reducir  $f$  )
            Volver a leer sensores }

```



```

Si Dder<Dmáx           // caso análogo al anterior
{ Durante x iteraciones
  (girar izquierda
   reducir f )
  Volver a leer sensores }

Si Dizq=Dmáx y Dder=Dmáx // el pez está por chocar derecho
{ elegimos izq. O der. Al azar
  Durante x iteraciones
  (girar el sentido obtenido al azar
   reducir f )
  Volver a leer sensores }

```

Tarea 6: Nadar chocando contra bordes

El pseudocódigo asociado a esta tarea es simple, ya que no hay actividad alguna de los sensores infrarrojos, y el pez no responde ante estímulos. Se sortea entonces la frecuencia de oscilación (permaneciendo dentro de valores válidos o posibles por el motor) y varía entre giros a la derecha, izquierda y avances rectos:

```

          Inicio:
          Si f>fmáx
            (reducir f)
          En otro caso
            Si f<fmín
              (aumentar f)
            En otro caso
              (sortear f) // aumentar o disminuir por sorteo

Sortear giro           //incluye navegar derecho
Girar o andar derecho según el sorteo
Retornar al inicio

```

Observaciones:

- Más adelante analizamos la terminación de la realización de esta tarea mediante las distribuciones de probabilidad de cada modo, por lo que no se va a tener un loop infinito como el anterior.
- Aún no se consideran las variaciones en la profundidad del pez que posteriormente también será aleatoria para esta tarea.

Tarea 7: Encontrar borde

Esta tarea reutiliza código de la tarea 2 ya que consiste en alcanzar el borde, una vez obtenido el objetivo la tarea culmina

Leer sensores

Si $D_{\text{frontal}} \text{ última} < D$ entonces
{ Si D_{frontal} disminuyó en las 5 últimas medidas
Mientras el offset de la cola sea menor que el máximo
(aumentar offset de la cola)

Volver a leer sensores }

En otro caso
Si $D_{\text{lateral}} < D$ entonces
{ Si D_{lateral} aumentó en las 5 últimas medidas
Mientras el offset de la cola sea mayor que el mínimo
(disminuir offset de la cola)
Volver a leer sensores }

En otro caso // $D_{\text{frontal}} \geq D$ y $D_{\text{lateral}} \geq D$ //búsqueda de borde
{ Mientras $D_{\text{frontal}} \geq D$
(navegar derecho // con offset nulo
Leer D_{frontal})
Volver a leer sensores }

6.11. El Programa principal

Admitiendo que tenemos las tareas como funciones realizadas, en el cuerpo del main se debe llamar a las distintas tareas según el modo y la tarea sorteados. Dado que los sorteos se van a efectuar luego de un tiempo T_{st} de 5 segundos para las tareas y de $T_{\text{sm}} = 10 * T_{\text{st}}$ para los modos, surge la necesidad de realizar una función de retardo con base por ejemplo de 10 milisegundos: `void delay10ms(){...}`. Observando que en cada ranura de tiempo sólo importa saber el resultado de un sorteo (sea en modos o en tareas), el programa principal debe llamar a una función

`int sorteo(double p1, double p2,.....double p7){...}` que retorne el entero 1 con probabilidad $p1$, el 2 con probabilidad $p2$ y así sucesivamente. Esta manera de realizar el sorteo es muy útil, pues si estamos sorteando entre los modos simplemente llamamos mediante la línea:

`modo = sorteo (x1,x2,x3,0,0,0,0);`

donde los valores $x1$, $x2$ y $x3$ dependen del valor que tenía el modo antes del sorteo.

Resulta conveniente por razones de legibilidad, definir constantes que representen con palabras los distintos modos y tareas, y estas últimas como variables enteras.

7. Primera implementación

7.1. Implementación

El primer prototipo que realizamos contó simplemente con un servo y un microcontrolador pic, y fue realizado para ver el movimiento de éste en el agua.



Figura 7.1. Primer prototipo

Luego, construimos un segundo prototipo de pez robot, que consta de dos cajas de acrílico como se ve en la foto. Una tiene en su interior los sensores de distancia (uno frontal y dos laterales), un servomotor que actúa sobre la primera junta, las baterías y el microcontrolador utilizado (pic 16F876). La segunda caja, contiene un solo servomotor, el cual actúa sobre la segunda junta. Las dimensiones aproximadas de éste son: 24 cm (largo) x 10 cm (alto) x 6 cm (ancho).



Figura 7.2. vista lateral del prototipo

7.1.1. Componentes utilizados

Los componentes utilizados en este prototipo fueron los siguientes:

- Un servo motor mini
- Un Servo motor común
- Cuatro baterías de 2800 mAh
- Un pic modelo 16F876
- Tres sensores SHARP GP2XX

7.1.2. Arquitectura del sistema

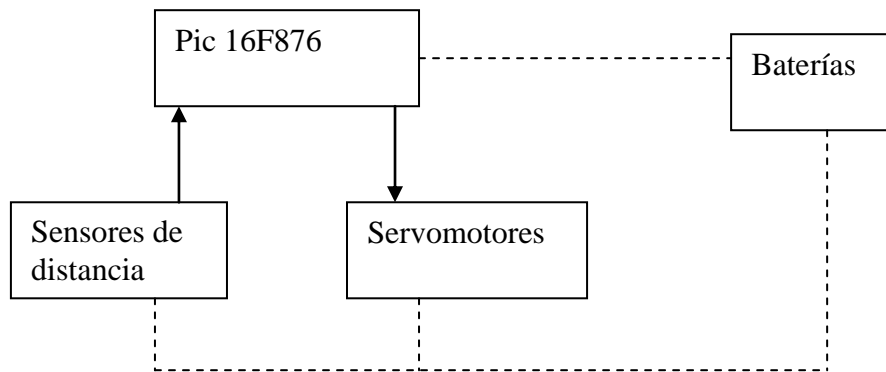


Figura 7.3. Esquema de la arquitectura del sistema

7.2. Control

Para este prototipo, programamos patrones de nadado, cada uno de los cuales puede ser visto como una función de la posición del pez y del tiempo:

$$\theta_i = \theta_i(x, t)$$

donde θ_i es el ángulo relativo de la i -ésima junta, y (x, t) representan la posición y el tiempo respectivamente.

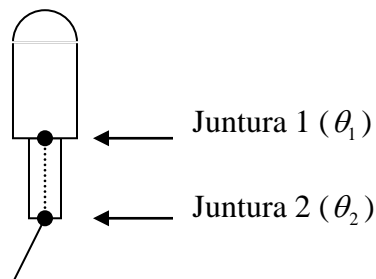


Figura 7.4. Esquema de la estructura, vista superior

De las tareas anteriormente descritas, fueron implementadas en este prototipo las siguientes :

i. Encontrar la pared

El pez nada en forma rectilínea (para lo cual producimos oscilaciones simétricas de la cola en torno a su eje) hasta encontrar la pared.

ii. Seguir la pared

El pez trata de mantenerse nadando siguiendo a la misma. En caso de perderla va corrigiendo la trayectoria para acercarse, y en caso de llevar una trayectoria de colisión corrige para alejarse.

iii. Nadar evadiendo obstáculos

Similar al anterior, pero en este caso si pierde de vista a la pared, continúa nadando.

Los algoritmos de control fueron similares a los ya descritos en secciones anteriores. Vale la pena comentar que cada tarea ocupa aproximadamente un 25% de la memoria del pic. La programación fue realizada en lenguaje C, y luego compilada en assembler. Finalmente, ya que no hay comunicación externa con una PC, cada vez que se quiere implementar una nueva tarea, se debe retirar el pci y re-programarlo.

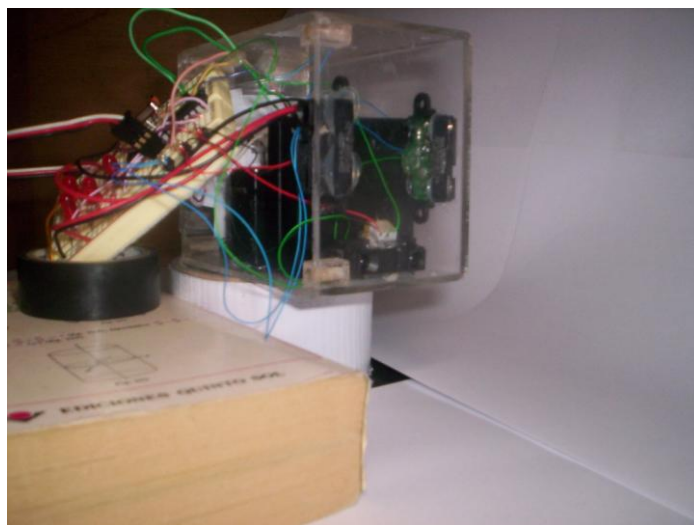


Figura 7.5. Vista del prototipo durante los ensayos previos en tierra.

7.3. Algunos resultados

Realizamos ensayos de las tareas anteriormente detalladas en una piscina estructural (con dimensiones 3 m x 1.5 m). En las siguientes figuras se observan algunas de las trayectorias observadas durante la realización de estos ensayos:

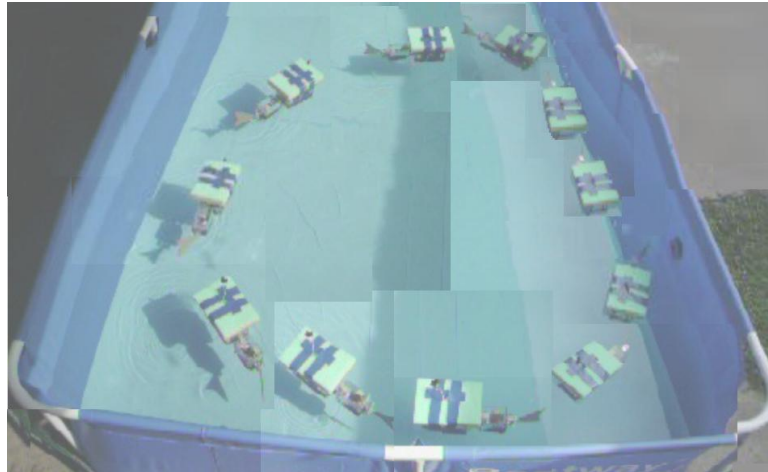


Figura 7.6. El pez nadando evadiendo las paredes de la piscina

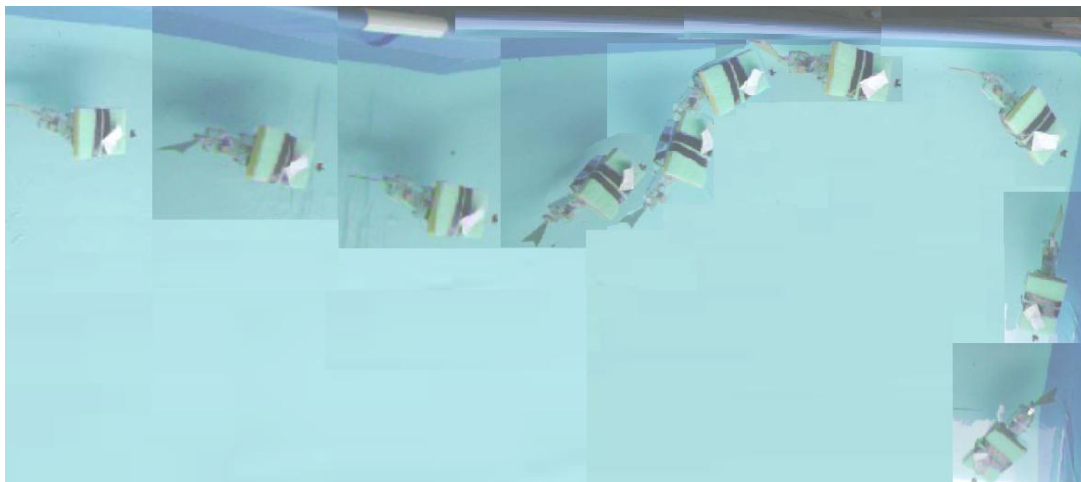


Figura 7.7. El pez siguiendo la pared

7.4. Principales inconvenientes y dificultades

Un inconveniente encontrado, fue el sellado del prototipo para que no le ingrese agua en su interior, lo cual aunque al principio fue difícil, luego resultó sencillo con la práctica. Otro inconveniente no menor fue el de la distribución de masa para lograr la estabilidad del mismo (cosa que fue solucionada en parte, colocando pesas en ciertos lugares, también un flotador en la parte superior).

7.5. Conclusiones

El prototipo realizado funcionó correctamente según lo esperado. Los algoritmos de control de las tareas primeramente ensayados en el simulador de Matlab con pequeñas modificaciones produjeron en un prototipo real buenos resultados.

7.6. Divulgación

Lo realizado en este prototipo, fue presentado en un poster en la primera Escuela de Verano Latinoamericana IEEE RAS en Robótica, realizada en la ciudad de Santiago de Chile del 12 al 14 de Diciembre de 2007. En este evento, obtuvimos el segundo lugar en la competencia de posters.

El jurado de dicha competencia, estuvo integrado por:

- Dr. Kim Boyer, Ohio State University.
- Dra. Alicia Casals, UPC.
- Dr. Javier Ruiz del Solar, Universidad de Chile



Figura 7.8. Poster presentado en la Primera Escuela de Verano Latinoamericana IEEE RAS en Robótica.

8. Baterías

8.1. Baterías Ni-Cd (Nickel Cadmium)

Las baterías de Ni-Cd (Níquel-Cadmio) constan de dos polos, uno positivo con hidróxido de níquel y uno negativo de hidróxido de cadmio. A lo largo de la vida de la batería, su capacidad máxima va disminuyendo debido al efecto memoria.

La potencia nominal de las baterías de Ni-Cd es de 1.2V, pero totalmente cargadas ofertan una potencia de 1.4 V durante un corto período de tiempo. Se consideran totalmente descargadas cuando su voltaje es inferior a 1.1V por celda.

El voltaje de las baterías de Ni-Cd tiende a caer de golpe, quedando descargadas de un momento para otro después de un período considerable de utilización.

La vida de las baterías de Ni-Cd es de más o menos 1000 ciclos de carga y debido a su pequeña resistencia interna (100-200 mOhm) tiene una gran alta tasa de descarga .

Una batería de Ni-Cd recién cargada puede perder cerca del 10% de su energía en las primeras 24 horas, disminuyendo posteriormente a una tasa de 1% a 3% por día. Después de unos días ésta tasa va decrementándose hasta llegar a ser mínima. Debido a esto, una batería de Ni-Cd cargada no estará descargada antes de 3 meses como mínimo en condiciones normales de 20° de temperatura.

Las baterías de Ni-Cd no se deben cargar más de 1 vez al día ya que se calientan al cargarse o al usarlas y no se deberían de volver a cargar hasta que se enfriasen a temperatura ambiente [28-29].

Ventajas	Inconvenientes
Son baratas	Son contaminantes
Buena capacidad y autonomía	Tienen efecto memoria
Tienen una gran vida útil	Pierden un 10% de carga las primeras 24 horas
Alta descarga debido a su pequeña resistencia interna.	Se autodescargan
Tienen un tiempo menor de carga que otras baterías recargables	No se pueden cargar varias veces al día

Precauciones Ni-Cd

Para almacenar sin uso las baterías Ni-Cd, es recomendable que estén descargadas, ya que aunque existe tasa de autodescarga, ésta se va decrementando con el tiempo hasta hacerse prácticamente despreciable.

Se han de cargar solamente el tiempo necesario, un exceso de carga puede ser más perjudicial que el efecto memoria.

No se debe descargar la batería a menos 0,9 Voltios (por elemento) ya podrían dañarla. Esto se llama descarga profunda.

8.2. Baterías NiMH (Níquel Metal Hydride)

Las baterías de níquel metal hidruro (NiMH) son algo más caras que las de Ni-Cd por los metales que se usan para fabricarlas y no poseen efecto memoria (sí lo poseen, pero muy pequeño) y se caracterizan por su elevada densidad de energía por volumen y peso (30 % más que las de Ni-Cd a igualdad de tamaño y peso).

La potencia nominal de las baterías de NiMH es de 1.2V, pero totalmente cargadas ofertan una potencia de 1.4 V durante un corto período de tiempo. Se consideran totalmente descargadas cuando su voltaje es inferior a 1.1V por celda.

En cuanto a la vida de éstas, es inferior a las de Ni-Cd, más o menos entre los 400 y 600 ciclos de carga.

La baja resistencia interna de estas baterías permite altos índices de carga. Actualmente es habitual cargarlas en tasas de 3 a 5 Amperios, pero se están probando cargarlas a más de 9 Amperios con buen resultado. Este tipo de baterías se pueden cargar varias veces al día. De hecho, los fabricantes indican que se pueden cargar 3 veces al día, siendo la segunda y tercera carga mejor porque ésta tendrá más autonomía una vez cargada. Eso sí, entre carga y descarga, hay que dejar enfriar las baterías a temperatura ambiente.

Se trata de una tecnología no especialmente adaptada para la carga permanente ya que el NiMH es electroquímicamente más susceptible a la sobrecarga que el Ni-Cd.

Padecen una alta autodescarga (2% a 8% día), siendo su tasa mucho mayor que las de Ni-Cd debido a los átomos de Hidrógeno en fuga.

En las Ni-MH si se dejan 1 mes sin usar, en la primera carga-descarga no dan todas sus prestaciones, por ello interesa usarlas al menos una vez en entrenamientos libres antes de emplearlas en los oficiales o en carrera [28-29].

Ventajas	Inconvenientes
Tienen más densidad de energía que las de Ni-Cd por lo que a igual capacidad pesan menos y tienen menos volumen.	Menos durabilidad de ciclos que las de Ni-Cd
No son contaminantes	Muy propensas a las sobrecargas
No tienen efecto memoria (Lo tienen pero despreciable)	Necesita más tiempo de carga que una de Ni-Cd
Se pueden recargar varias veces al día	No ofertan una tasa de descarga tan grande como Ni-Cd
Tienen posibilidad de cargarse a altas tasas de amperaje.	-

Precauciones NiMH

Para almacenar sin uso las baterías NiMH, es recomendable que se almacenen con un 40% ó 50% de carga.

No descargar nunca, por debajo de 0,9 Voltios por batería.

Aunque no padece el efecto memoria, es aconsejable no iniciar un proceso de carga antes de que haya perdido un 50% de su capacidad. De vez en cuando, es aconsejable que se descargue completamente antes de ponerlo a recargar (Al menos cada 3 meses).

8.3. Baterías LiIon (Litio-Ión)

Las baterías de Litio-Ión (LiIon) tienen altísima densidad energética , apenas efecto de autodescarga (0.3% día) y sin efecto memoria. Con estas baterías se consigue alcanzar una capacidad nominal 2 veces superior a las de Ni-Cd y 1,5 veces superior a las de NiMH. Una batería de LiIon pesa menos de la mitad que una batería de Ni-Cd de igual capacidad.

El proceso completo de carga, como en el caso de las de NiMH, dura más tiempo que las de Ni-Cd.

Debido a que las baterías de LiIon tienen una resistencia interna considerable (150-250 mOhm) no ofertan alta descarga, por lo que no es recomendable usarlas para alimentar elementos que precisen altas demandas energéticas (Motores de tracción de eléctricos) si no que para servos y elementos que precisen una demanda energética media.

En cuanto a la vida de éstas, es muy inferior a las de Ni-Cd, más o menos entre los 300 y 500 ciclos de carga [28-29].

Ventajas	Inconvenientes
Tienen más densidad energética que las de Ni-Cd y NiMH por lo que a igual capacidad pesan menos de la mitad y poco volumen.	Menos durabilidad de ciclos de vida que las de Ni-Cd y NiMH
Apenas autodescarga	Propensas a las sobrecargas
No tienen efecto memoria	Necesita más tiempo de carga que una de Ni-Cd
Se pueden recargar varias veces al día	Ofertan una tasa media de descarga
-	Son muy contaminantes

Precauciones NiMH (Litio-Ión)

Aunque no padece el efecto memoria, es aconsejable no iniciar un proceso de carga antes de que haya perdido un 50% de su capacidad. De vez en cuando, es aconsejable que se descargue completamente antes de ponerlo a recargar (Al menos cada 3 meses).

Para almacenar sin uso las baterías LiIon, es recomendable que se almacenen con un 40% ó 50% de carga.

8.4. Comparativa de baterías recargables

Características	Ni-Cd	NiMH	LiIon
Potencia nominal por célula	1.2V	1.2V	3.6V
Precio	Económicas	Algo más caras que las de Ni-Cd	Precio elevado
Ciclos de vida (duración)	1000 ciclos	400-600 ciclos	300-500 ciclos
Almacenamiento de la batería sin uso	Descargadas	Almacenar con 40% ó 50% de carga	Almacenar con 40% ó 50% de carga
Capacidad	Buena capacidad	30% más de capacidad que una de Ni-Cd a igualdad de peso y tamaño	2 veces más capacidad que una de Ni-Cd y 1,5 más que una NiMH a igualdad de peso y tamaño
Energía Densidad (Wh/kg)	40-60	60-80	80-100
Resistencia Interna (mOhm)	100-200	200-300	150-250
Máximo número de cargas por día (recomendable)	1	3	3
Sobrecarga	Perjudicial	Muy perjudicial, son muy sensibles	Perjudicial
Descarga	Alta descarga (Baja Resistencia)	Buena descarga (Resistencia más grande que Ni-Cd)	Descarga media (Por lo que no se usan para la tracción de eléctricos, sólo receptores y servos, etc)
Autodescarga (Sin usar)	De 1 a 3 % día	De 2 a 8% día	0,3% día
Recarga	Se necesitan varias horas	Necesita más tiempo que una de Ni-Cd	Necesita más tiempo que una de Ni-Cd
Efecto Memoria	Sí	No (Realmente tienen un pequeño efecto memoria que es despreciable)	No
Reciclaje	Reciclable, contaminante	Desechable, no contamina	Reciclable, muy contaminante

Temperatura de operación	Buena	Sensiblemente mejor operación que Ni-Cd a Baja y a alta temperatura	Buena
Perfil de descarga	Similar a NiMH	Similar a NiMH	-

8.5. Precios

Algunos de los precios actuales del mercado son:

Tipo	Capacidad (mAh)	Voltaje de la celda (V)	Precio (U\$S)	Vendedor	Marca	Tamaño
Ni-CD	700	1.2	1	www.mercadolibre.com.ar	Ulster	AA
NIMH	2000	1.2	2	www.CDRmedios.com.uy	Novacell	AA
NIMH	2500	1.2	2.5	www.CDRmedios.com.uy	Novacell	AA
NIMH	3000	1.2	2.23	www.CDRmedios.com.uy	Novacell	C
NIMH	1800	1.2	4	Devoto	Sanyo	AA
NIMH	2500	1.2	2.5	Devoto	Sumoll	AA
Li_Ion	710	3.7	16	www.deremate.com.uy	Motorola	40 mm x 25 mm X 5 mm
Li-Ion	11000	3.7	47.5	www.all-battery.com	Tenergy	185 mm x 72 mm x 7.8 mm
Li-Ion	16000	3.7	69.99	www.all-battery.com	Tenergy	170mm x 135 mm x 6.8 mm

Lo que se puede observar de la tabla que las baterías mas económicas son las de NI-Cd pero también son las de menor capacidad por celda, también vale la pena mencionar que son las mas aptas para el uso con motores eléctricos de tracción por su baja resistencia interna. Por otro lado se tienen las baterías de NI-MH que siguen siendo económicas, su capacidad de carga es casi 4 veces mayor a las anteriores y aunque su resistencia interna es mayor a las de NI-Cd pueden ser usadas para alimentar motores de tracción. Antes de mencionar las de Li-Ion vale la pena destacar que con 4 celdas de 1.2 V se llega a una tensión de 4.8 V, el cual es el voltaje utilizado en la mayoría de los microcontroladores y sensores. Por último, la batería de Li-Ion tiene una tremenda capacidad de carga en comparación con las dos clases anteriores, pero su precio es bastante mayor y cabe destacar 2 inconvenientes para nuestro uso, el primero es que son celdas de 3.7 V lo

cual implica que para llegar a 5V habría que poner 2 celdas de estas y luego un regulador o conversor DC-DC lo cual implicaría un consumo extra, el otro problema es que no son recomendables para el uso de motores eléctricos de tracción.

8.6. Detalles de consumo

Dispositivo	Consumo (mA)	Tiempo aproximado de operación (en porcentaje de operación)	Cantidad en operación	Consumo total estimado(mAh)
Sensor de distancia SHARP GP2Y0A21YK0F	30	100	5	150
Sensor de presión MPX2010	6	100	1	6
Servomotor	140(operación) 20 (stand by)	60	2	177.6
Motor CC	50	100	1	50
Robostix Wi-fi Pack	450	100	1	450

Consumo total estimado : 833.6 mAh

Autonomía

Utilizando un banco de 4 baterías de NI-MH de 2500 mAh el cual se encargaría de alimentar

-Sensores de distancia	150 mA
-Sensor de presión	6 mA
-Servomotores	177.6 mA
-Motor CC	50 mA
-Robostix	20 mA

Darían un total a 403.6 mA, lo cual equivale a una autonomía de aproximadamente 6.2 horas

Si usamos otro banco de 2500 mAh el cual se encargaría de todas las placas Gumstix menos la Robostix, lo que equivale a un consumo de 430 mAh brindando una autonomía de 6 horas.

8.7. Circuito propuesto para el cargador

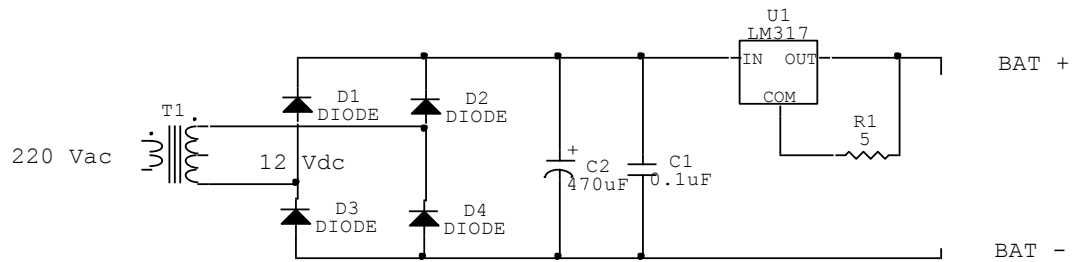


Figura 8.1. Circuito para cargar las baterías.

El circuito de la figura es un regulador de corriente de 250 mA que se encarga de realizar esta función es el integrado LM317. El mismo asegura entre sus patas COM y OUT una diferencia de tensión de 1,25 V. Además como la corriente por la pata COM se considera despreciable, la corriente queda determinada por la resistencia entre COM y OUT, algo a tener en cuenta es la potencia disipada por la resistencia la cual en este caso esta en el orden de 0,3 W. Por esta razón la resistencia debe ser de al menos $\frac{1}{2}$ W. La primera parte del circuito es simplemente un puente rectificador de onda completa y 2 condensador de filtrado

9. Diseño de la arquitectura de software y hardware

9.1. La arquitectura del Hardware

Los componentes de Hardware a utilizar son los siguientes:

- PC
- Gumstix (PC/104)
- Wifi-Stix (placa de comunicación Wi-Fi)
- Robostix (microcontrolador ATMEL)
- Sensores de distancia SHARP
- Sensor de Presión Freescale
- Servomotores
- Baterías

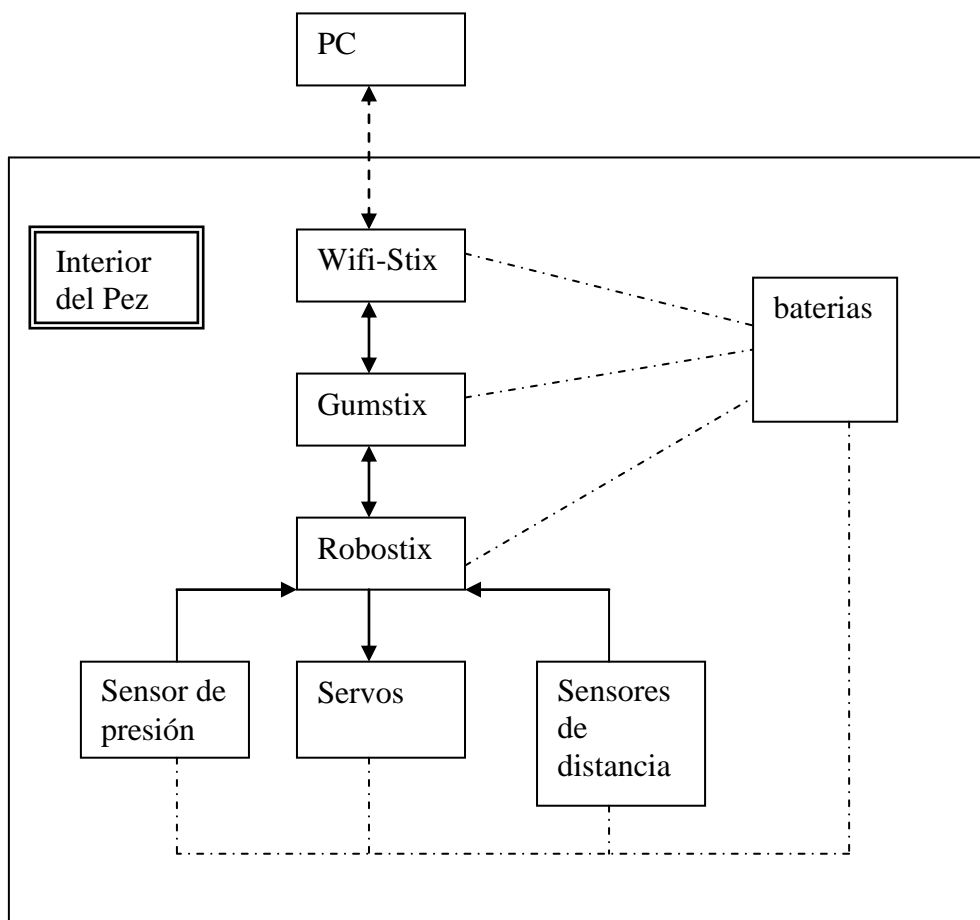


Figura 9.1. Arquitectura del Hardware

¿Qué realiza cada uno de los componentes?

PC:

- Función principal: ser la puerta de entrada al Gumstix permitiendo de esta forma al usuario ejecutar, dejar de ejecutar, o alterar los programas que en él se estén ejecutando.

- Función secundaria: Desplegar dinámicamente de forma amigable para el usuario los datos recogidos por los sensores, y el estado de los actuadores y del pez. Almacenar estos datos.

Gumstix:

- Función principal: ejecutar programa de selección de comportamientos, estados de ánimo y tareas. Ejecutar los programas de control de las tareas.
- Función secundaria: obtener del Robostix los datos de los sensores y parámetros de control de los actuadores, comunicar éstos al PC.

Robostix:

- Función principal: enviar datos al Gumstix.

9.2. La Arquitectura del Software

Detalles de las implementaciones de los programas:

Gumstix:

-Elige estado de ánimo.
-Elige modo.
-Elige tarea.
-Ejecuta la tarea con sus parámetros.
-Mientras se esté ejecutando una tarea en un cierto modo y estado de ánimo, toma los datos del Robostix y hace un archivo con éstos para ser leído por la PC.

Robostix:

- envía cuando se lo pida los datos al Gumstix.

PC:

-Lee archivo de datos realizado por el Gumstix.
-Actualiza gráficas (históricas e instantáneas).
-Actualiza registro histórico.

El programa principal residente en la PC se llamará programa de despliegue o PdD.

El programa principal residente en el Gumstix se llamará programa principal del pez o PPdP, además existirán por lo menos un programa de control o PdCont (por tarea), y un programa de comunicación con el Robostix o PdComuGR.

El programa residente en el Robostix se denominará programa de comunicación con el Gumstix o PdComuRG.

Obviamente estos programas a su vez invocarán a otros durante su ejecución.

Cosas a guardar en memoria:

Robostix: Lecturas de los sensores (memoria de corto plazo).
Gumstix: Archivo con las lecturas de los sensores.
PC: Registro histórico de las lecturas de los sensores.

9.2.1. Descripción a nivel del usuario:

-Se debe correr una hiper-terminal del Gumstix. En ésta el usuario realiza las siguientes cosas:

Ejecuta el programa principal dentro del Gumstix.
Finaliza la ejecución del programa principal (resetea al pez).
Altera a su voluntad el programa principal.

-El usuario puede observar además los datos de los sensores y el comportamiento del pez, ejecutando el programa de despliegue.

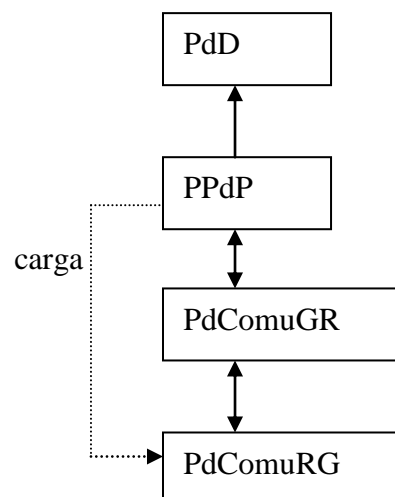
¿Qué brinda éste al usuario?

- Registro histórico de datos
- Posibilidad de acceder o graficar datos de este registro.
- Opción de inicializar este registro, y eventualmente detener la adquisición de datos.
- Visualización de gráficas en tiempo real con la evolución temporal de los datos.
- Permite eventualmente imprimir los datos en forma clara para analizar.

9.2.2. Lenguajes de implementación de los códigos de los programas:

Robostix: C/Assembler
Gumstix: C
PC: Matlab

9.2.3. Diagrama del flujo de la información a nivel de software:



9.2.4. Detalle de los programas implementados

- i2c-io: modificación realizada por nosotros del programa original i2c-io disponible en la página de Gumstix [30] que lee y/o escribe en los puertos y registros del Robostix. Actúa como una función capaz de ser invocada desde otros programas. Esto es un equivalente a lo que fue anteriormente denominado PdComuGR.
- Programas de las tareas (mismo nombre que tarea respectiva): toma como dato el tiempo de ejecución y las posiciones iniciales de los servos. Realiza la ejecución de la tarea respectiva invocando a i2c-io para tomar las lecturas de los sensores y actuar sobre los motores. Implementa el algoritmo de control de la tarea correspondiente. También otra cosa que realiza es periódicamente llamar a la función que crea el archivo .dat con los datos a desplegar (despliegue).
- despliegue: crea un archivo .dat con información sobre el estado actual de nuestro sistema.
- prog_principal: realiza el sorteo de los estados de ánimo, modos y tareas, ejecutando estas según corresponda. Este es el programa que se encuentra compilado, y es el programa a ejecutar en el Gumstix (PPdP). Este programa utiliza todas las funciones que se detallaron anteriormente y algunas más que también implementamos se encuentran disponibles en los archivos que acompañan a esta documentación.
- El programa de despliegue, el cual es residente en la PC, está compuesto de dos programas que deben ser ejecutados a la vez: uno que copia los archivos desde el Gumstix a la PC y lleva un registro histórico de la evolución del sistema, y otro que despliega en tiempo real los datos.

9.3. Tareas realizadas para la implementación de estos programas y para iniciar la programación del Gumstix

1. Ensamblaje y encendido del gumstix
2. Establecimiento de conexión serie con PC
3. Seteo de comunicación wi-fi
4. Establecimiento de comunicación wi-fi con PC
5. Creación de un simple archivo en el gumstix (archivo de texto con vi)
6. Instalación de la versión R1161 del gumstix buildroot en la PC
7. Compilación con el buildroot de programa hola mundo
8. Envío de archivos desde y hacia el gumstix mediante scp
9. Ejecución del programa hola mundo en el gumstix
10. Establecimiento de conexión SPI con el robostix
11. Cargamos programa que implementa i2c en el robostix
12. Establecimiento de comunicación i2c entre gumstix y robostix
13. Modificación de la función en C que lee y/o escribe registros y puertos del robostix
14. Desarrollo en el gumstix de un script que prende y apaga un led temporizado

15. Desarrollo en C de un programa que prende y apaga un led temporizado
16. Desarrollo de un script en la PC que copia a esta un archivo .dat con los datos de los sensores del gumstix, actualizando un historial con estos. Prueba.
17. Implementación de nuestras propias subrutinas de delay y random pues las de C no corren en el gumstix.
18. Desarrollo de un programa en C que realiza la decisión de estados de ánimo, modos y tareas, y llama a la tarea correspondiente. Prueba.
19. Desarrollo de un programa en C que prueba el conversor AD. Calibración de los sensores.
20. Desarrollo de un programa en C que configura PWM de los motores. Prueba del mismo con un motor en el robstix. Calibración.
21. Desarrollo de una función en C que dado un ángulo en grados de la posición de un servo, lleva este a la misma.
22. Desarrollo de una función en C para movimiento sinusoidal del servo.
23. Desarrollo de una función en C para realizar la tarea encontrar pared.
24. Prueba de comunicación wi-fi en el agua.
25. Establecimiento de una comunicación mediante pscp entre un sistema Windows y el gumstix.
26. Implementación de la interfaz gráfica (primitiva) para despliegue de los resultados en tiempo real en Windows.
27. Implementación de una función en C para realizar la tarea descansar contra el borde.
28. Implementación de una función en C para realizar la tarea seguir trayectoria poligonal.
29. Implementación de una función en C para realizar la tarea nadar contra el borde.
30. Prueba final de comunicación wi-fi en el agua. Resultado de profundidad mayor a 20 cm.
31. Implementación de una función en C para realizar control de profundidad (con un servo).
32. Implementación de una función en C para realizar la tarea ir y volver
33. Implementación de una función en C para realizar la tarea de descansar contra un borde.
34. Implementación del programa final uniendo todas las partes realizadas.
35. Prueba general del programa
36. Implementación definitiva de la interfaz grafica para el despliegue de datos.

10.El Enlace Inalámbrico

10.1. Introducción

Es imprescindible para nuestra aplicación la existencia de un enlace inalámbrico que permita tanto comandar los movimientos del pez en respuesta a estímulos, como también programarlo y recibir información del mismo para ser luego desplegada y analizada mediante la PC. El gran desafío aquí presente es la inmersión de nuestro robot en agua, medio con propiedades muy diferentes a las del aire que es por donde existe la generalidad de enlaces de radiofrecuencia, al ser un muy buen dieléctrico.

En este apartado se estudia la viabilidad de un enlace inalámbrico entre una antena de una notebook WiFi compatible y la antena Wifistix, que viene junto con el pack Gumstix que se utilizará en este proyecto.

El texto se ordena como sigue: se hace un breve repaso de la propagación electromagnética, atendiendo parámetros como la atenuación y profundidad de penetración. Luego se estudian las propiedades del agua doméstica, clasificándola según distintas frecuencias como conductor o dieléctrico. Más adelante se analizan los terminales de la comunicación: ganancia y sensibilidad de las antenas, tasas de transmisión y margen en dB destinado a las pérdidas. Finalmente se estudian las pérdidas tanto por atenuación como reflexión en los cambios de medios, para obtener una respuesta por sí o no de la viabilidad.

Vale destacar que este análisis teórico va a estar sujeto a sucesivos ensayos mediante la inmersión de la antena Wifistix, y se tiene en cuenta ante adversidades la alternativa de utilizar el primer enlace exitoso para el prototipo de pez anterior, por lo que esta etapa no nos impedirá el avance, mas puede requerirnos conexiones especiales y estudio de compatibilidades, en el segundo caso.

10.2. Propagación electromagnética en un conductor

Consideremos una onda electromagnética propagándose sobre un medio conductor lineal e isotrópico. Usando la ecuación de Maxwell del vector divergencia del desplazamiento eléctrico y que la carga neta dentro de un conductor es 0 se tiene:

$$\nabla \cdot \vec{E} = \nabla \cdot \left(\frac{1}{\epsilon} \vec{D} \right) = \frac{1}{\epsilon} \nabla \cdot \vec{D} = \frac{\rho}{\epsilon} = 0 \quad (10.1)$$

Aplicando ahora la identidad matemática: $\nabla \times \nabla \times \vec{X} = \nabla(\nabla \cdot \vec{X}) - \nabla^2 \vec{X}$ al campo eléctrico obtenemos entonces que $\nabla \times \nabla \times \vec{E} = -\nabla^2 \vec{E}$. Ahora recurriendo a las siguientes

relaciones de Maxwell: $\nabla \times \vec{H} = \vec{J} + \frac{\partial \vec{D}}{\partial t}$, y $\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$ resulta que:

$$\nabla \times \nabla \times \vec{E} = \nabla \times \left(-\mu \frac{\partial \vec{H}}{\partial t} \right) = -\mu \frac{\partial}{\partial t} (\nabla \times \vec{H}) = -\mu \frac{\partial}{\partial t} \left(\vec{J} + \epsilon \frac{\partial \vec{E}}{\partial t} \right) = -\mu \frac{\partial}{\partial t} \left(\sigma \vec{E} + \epsilon \frac{\partial \vec{E}}{\partial t} \right) = -\nabla^2 \vec{E}$$

Se ha utilizado la linealidad del medio como también que $\vec{J} = \sigma \vec{E}$. En conclusión resulta la ecuación de onda para \vec{E} en un medio conductor:

$$\nabla^2 \vec{E} - \mu \epsilon \frac{\partial^2 \vec{E}}{\partial t^2} - \mu \sigma \frac{\partial \vec{E}}{\partial t} = 0 \quad (10.2)$$

En el caso de una onda plana y uniforme (en la que el campo eléctrico depende únicamente de su dirección de propagación x) y admitiendo una expresión fasorial:

$$\vec{E}(x,t) = \vec{E}_o e^{-\gamma x} e^{j\omega t} = e^{-\alpha x} \vec{E}_o e^{j(\omega t - \beta x)} \quad (10.3)$$

Al sustituir esta expresión en la ecuación de onda se desprende que la atenuación α es igual a:

$$\alpha = \omega \sqrt{\frac{\mu\epsilon}{2} \left(\left(1 + \left(\frac{\sigma}{\omega\epsilon} \right)^2 \right)^{1/2} - 1 \right)} \quad (10.4)$$

Notemos aquí que para un buen dieléctrico se tiene que $\frac{\sigma}{\omega\epsilon} \ll 1$, mientras que en un

buen conductor $\frac{\sigma}{\omega\epsilon} \gg 1$.

En nuestra aplicación va a ser de gran interés la “profundidad de penetración” δ , que se define como la profundidad en la que la amplitud de la onda se atenúa en un factor de $1/e$.

Entonces $e^{-\alpha\delta} = e^{-1} \rightarrow \delta = \frac{1}{\alpha}$. Esto es tan solo un repaso de propagación. Más detalles se hallan en la referencia [31].

10.3. Propiedades del agua doméstica

Dada la escasez de estudios del agua doméstica para la propagación de ondas electromagnéticas a muy altas frecuencias, nos limitaremos aquí obteniendo aproximaciones basadas en ensayos que estudian la propagación en agua realizados en [2]. Estos consideran antenas que trabajan a frecuencias de 0.9 y 1GHz, y se concentran en parámetros del agua tales como la conductividad σ , permitividad dieléctrica ϵ y velocidad de propagación, entre otros.

A partir de estos ensayos se procederá entonces estimando los parámetros de propagación relevantes en el agua, a una frecuencia de 2.4GHz, que es en la que vamos a trabajar en este proyecto. Su procedimiento consistió en situar las antenas 3100 (0.9GHz) y 3101 (1GHz) de GSSI y medir los tiempos dobles de propagación de la reflexión de la onda en planchas metálicas. Los valores útiles para nuestra aplicación y registrados en el ensayo realizado en [32] fueron los siguientes:

f de antena	ϵ	$\sigma (mS/m)$	$v(cm/ns)$	$\alpha(m^{-1})$	$\delta(cm)$
0.9GHz	81	400	3.329	8.37	11.95
1GHz	81	400	3.329	8.48	11.8

En otro ensayo en el que se varió la profundidad se obtuvo lo siguiente:

Prof (cm)	10	20	30	40	50
$v(cm/ns)$	3.40	3.30	3.50	3.40	3.38
ϵ	78.6	81.3	75.3	77	81.3

Al calcular la desviación estándar de cada parámetro se ve que $\epsilon = 78.7 \pm 0.9$ y $v = 3.4 \pm 0.6 cm/ns$.

A falta de más datos y observando las tablas anteriores utilizaremos $\varepsilon = 79\varepsilon_0$, $\sigma = 0.4 S/m$, $f = 2.4GHz$ y $\omega = 2\pi f$, donde ε_0 es la permitividad eléctrica del vacío. Con éstos valores resulta que $\frac{\sigma}{\omega\varepsilon} = 0.038 \ll 1$, por lo que consideraremos al agua doméstica como un buen dieléctrico a nuestra frecuencia de trabajo.

10.4. Terminales

- a. El Router.

El radio existente en el router es una tarjeta con las siguientes características:

3.2 Functions:

- Protocol : Compatible with IEEE 802.11b /g Standard
- Modulation Technique :
 - 802.11b : DSSS (Direct Sequence Spread Spectrum) with BPSK (1Mbps), QPSK (2Mbps), and CCK (5.5 and 11Mbps)
 - 802.11g : OFDM
- Transfer data rate : **802.11g** : 54, 48,36,24, 18,12,9,6Mbps, auto-fallback
802.11b : 11, 5.5, 2,1Mbps, auto-fallback
- Media access protocol : CSMA/CA with ACK
- Antenna: Built-in antenna connector
- Output Power : 15~18dBm(CCK), 10~14dBm(OFDM)
- RF sensitivity : @PER <0.08 ,11Mbps < -84dBm(typical)
 @ PER <0.08 ,54Mbps < -70dBm(typical)
- Security : 64/128-bit WEP Encryption
 64/128-bit TKIP Data Encryption
 64/128-bit AES Data Encryption

Normalmente la misma cuenta con una pequeña antena de entre 2dBm (pero en este caso esa antena la vamos a despreciar). Para maximizar el alcance vamos a suponer que la misma se conectará en 802.11b y por tanto la potencia mínima de salida será de 15dBm y que la sensibilidad será la tímicocontroladora de -84dBm*.

- b. El terminal Wifistix:

Contiene una placa de radiofrecuencia de 100mW, más una pequeña antena de 2dBm.

10.5. Estudio de Pérdidas de propagación

La señal electromagnética en el uplink (de Wifistix a notebook) debe atravesar primeramente distintos medios (silicona-agua y agua-aire), con sus respectivas pérdidas por reflexión, y además cada uno de estos medios tiene asociada una atenuación. Vale destacar que la atenuación dominante es evidentemente la del agua, y por ello las pérdidas por el corto trayecto del aire y la silicona serán bien despreciados. También se debe resguardar cierto margen en dBi para ocultar al ruido, siempre presente en todo sistema de telecomunicación.

a. Margen para el ruido.

Asumiremos que la velocidad del enlace en el peor caso puede llegar a 1Mbps (802.11b), que es la menos exigente en cuanto al margen de ruido, requiriéndose una señal con 4 o 5dBm por encima del mismo. Supongamos que el ruido de fondo estará en el entorno de los -100dBm. La señal en recepción debe tener mayor ganancia para ocultar al ruido.

Si consideramos entonces un margen de ruido de 6dBm, vamos a requerir que la señal tenga una potencia de al menos -94 dBm.

* Este análisis sigue un paralelismo al utilizado en la referencia [33].

Para el caso en que el radio no sea capaz de “captar” una señal de esa potencia (por tener una sensibilidad peor a esta cifra) se usará la sensibilidad de la tarjeta como peor caso. Es importante notar que si bien la tarjeta no será capaz de captar el ruido por su propia sensibilidad, procuraremos ese margen debido al propio ruido que genera la tarjeta y a la variabilidad de los enlaces.

Por lo tanto en el caso de enlaces con un receptor de baja sensibilidad usaremos como límite admitido de recepción la sensibilidad menos los 6dBm que esperamos de margen de ruido.

Cuando el receptor es el router, la potencia mínima que tiene que llegar para que pueda ser considerado el alcance de la misma es de -78dBm.

b. Pérdidas por atenuación

Retomemos aquí el ensayo realizado en agua doméstica, mediante el cual consideramos $\varepsilon = 79\varepsilon_0$, $\sigma = 0.4 S/m$, $f = 2.4GHz$ y $\omega = 2\pi f$. Sustituyendo en la expresión de atenuación se tiene que $\alpha = 11.99m^{-1}$, o de manera equivalente: ¡¡ La profundidad de penetración es de 8.3 centímetros!! . Esto es contundente, y refiere a que en 8.3 centímetros propagados en agua, el campo eléctrico se reduce a un 37% de su amplitud original.

c. Pérdidas por reflexión

Por lo visto de las propiedades del agua doméstica, admitimos que el es buen dieléctrico a 2,4GHz, por lo que el cambio de medio aire-agua es entre dieléctricos. Aplicando la ley de Snell, la de conservación de energía y la conservación de la componente tangencial del campo eléctrico en el cambio de medios (ver [32] por mayores detalles), se obtiene la siguiente relación entre el campo eléctrico incidente y el reflejado:

$$\frac{E_r}{E_i} = \frac{\left(\frac{\varepsilon_2}{\varepsilon_1}\right)\cos(\theta) - \sqrt{\left(\frac{\varepsilon_2}{\varepsilon_1}\right) - \sin^2(\theta)}}{\left(\frac{\varepsilon_2}{\varepsilon_1}\right)\cos(\theta) + \sqrt{\left(\frac{\varepsilon_2}{\varepsilon_1}\right) - \sin^2(\theta)}} \quad (\text{polarización vertical}), \quad (10.5)$$

o bien:

$$\frac{E_r}{E_i} = \frac{\cos(\theta) - \sqrt{\left(\frac{\varepsilon_2}{\varepsilon_1}\right) - \sin^2(\theta)}}{\left(\frac{\varepsilon_2}{\varepsilon_1}\right)\cos(\theta) + \sqrt{\left(\frac{\varepsilon_2}{\varepsilon_1}\right) - \sin^2(\theta)}} \quad (\text{polarización horizontal}), \quad (10.6)$$

donde θ es el ángulo de incidencia y los subíndices refieren a los medios 1 (original) y 2 (destino).

Evaluemos en el caso de medios 1-agua y 2-aire con incidencia normal ($\theta = 0$):

$$\frac{E_r}{E_i}(\theta = 0) = \frac{\left(\frac{1}{79}\right) - \sqrt{\left(\frac{1}{79}\right)}}{\left(\frac{1}{79}\right) + \sqrt{\left(\frac{1}{79}\right)}} = -0,7977 \text{ (polarización vertical),}$$

$$\frac{E_r}{E_i}(\theta = 0) = 0,7977 \text{ (polarización vertical).}$$

Esto anterior expresa que apenas un 20% del campo eléctrico atraviesa el medio agua-aire con incidencia normal. Con polarización vertical es posible hallar un ángulo incidente que anula el campo reflejado (“ángulo de Brewster”). Despejando se obtiene que $\theta_b = \arcsen \sqrt{\frac{79}{80}}$, que no es práctico dada su proximidad con el ángulo recto.

d. Vínculo entre la potencia y el campo eléctrico

Recordemos que el campo eléctrico en un dipolo corto toma la siguiente expresión (referencia básica [4]): $E(\theta, \varphi) = j\omega\mu \frac{e^{-jkr}}{4\pi r} \sen(\theta)\bar{I}l$. En un dipolo corto real, su potencia

radiada es $P_{rad} = \frac{\pi}{12} \eta I^2 \left(\frac{l}{\lambda}\right)^2$. Obsérvese que $P_{rad} = K |E(\theta, \varphi)|^2$, o equivalentemente en dBm: $G(dBm) = 10\log_{10} K + 20\log_{10} |E(\theta, \varphi)|$

e. Pérdidas totales y profundidad máxima

La potencia mínima de un router en recepción es de -78dBm, mientras que la potencia de emisión de la placa de radiofrecuencia en el Wifistix es de 20dBm, por lo que se tienen entre terminales un rango de 98dBm que debe ser superior a las pérdidas. Dentro de las mismas, las dominantes se hallan en el agua, siendo tanto de atenuación como reflexión. Estimemos en 3dBm la caída potencia por efecto de la propagación en unos metros de aire entre el router y la superficie de agua, y 10dBm perdidos por la reflexión en el cambio de medio agua-aislante del pez y atenuación en este aislante. Entonces el resto de las pérdidas se destinan al agua, quedando un margen de $S = 85dBm$. Sea x la profundidad a la que se sumerge la antena Wifistix, y $|E_o|$ la magnitud inicial del campo eléctrico. Para una incidencia normal recordemos que su magnitud se reduce a un 20,23 por ciento. Entonces las potencias inicial y final en dBm son, respectivamente:

$$dBm_o = 10\log_{10} K + 20\log_{10} |E_o|,$$

$$dBm_f = 10\log_{10} K + 20\log_{10} (0,2023 |E_o| e^{-11,99x})$$

La diferencia entre las potencias inicial y final en dBm no debe superar la sensibilidad, por lo que $S \geq dBm_o - dBm_f$. Despejando para x se obtiene finalmente que: $x \leq 68,3cm$.

10.6. Experimento

Hemos realizado un simple experimento sumergiendo el Gumstix con su antena Wifistix dentro de un frasco hermético y en agua, todo dentro de una hielera rectangular de 30 centímetros de altura y 30cmx25cm de base. Se comprobó que el led del Wifistix permaneció encendido al ser sumergido y también al mover el frasco en todos los espacios de la hielera, por lo que la comunicación es posible.

10.7. Conclusiones

El cálculo con el que se obtiene los 68,3 centímetros de profundidad es optimista, pues se basa en la sensibilidad tímicrocontroladora de un router. Sin embargo, nuestro experimento nos permite asegurar que un enlace satisfactorio es posible al menos a 12,5 centímetros (la mitad del menor de los lados, pues la onda atraviesa las paredes. Esto es muy valioso para nuestra aplicación, y una vez determinada la profundidad máxima en un enlace experimental procuraremos no superarla, para evitar problemas de cobertura intermitente, pérdidas de información y otras contingencias.

CAPITULO 2

Manual de Usuario

1. Requisitos previos:

- Computadora personal con sistema operativo Linux o Windows. Se recomienda Windows 2000 o superior y Linux Fedora 6.
- Router inalámbrico de 55Mbps o mayor velocidad.
- Al menos 512 MB de Memoria RAM libre en Linux o 2 GB en Windows.
- Se recomienda muy fuertemente tener una conexión a internet para poder consultar dudas en la página de Gumstix y también bajar los programas a utilizar.

2. Software a instalar:

Como el Gumstix corre sobre un sistema operativo basado en Linux, es necesario para poder compilar los programas a ser cargados de tener instalada en su PC alguna plataforma Linux. Se recomienda en especial tener Linux Fedora ya que esta versión de Linux es la que se referencia como usada en la página de Gumstix.

Por más información, recurra a la página web de Gumstix:
http://docwiki.gumstix.org/index.php/Software_on_gumstix

Instrucciones a seguir para la instalación sobre sistema operativo Windows:

W.1. Instale WMare Player

(para bajar e instalar este programa ir a : www.vmware.com/products/player)

W.2. Baje la imagen virtual de Fedora Core 6 de la página:
www.thoughtpolice.co.uk/vmware

W.3. Instale esta en el WMare Player

W.4. Siga las instrucciones de Linux

Instrucciones a seguir para la instalación sobre sistema operativo Linux:

Para poder compilar sus programas en la PC externa, usted deberá instalar el gumstix buildroot, luego mediante SCP usted podrá transferir archivos entre su PC y el Gumstix.

Instalación del buildroot en su PC:

L.1. En su PC Instalar los siguientes paquetes de Linux:

autoconf, automake, bison, byacc, flex, gcc, gcc-c++, make, makeinfo, ncurses-devel, subversion, texinfo, zlib-devel

L.2. En una terminal de su PC usar los siguientes comandos:

% `svn co http://svn.gumstix.com/gumstix-buildroot/trunk gumstix-buildroot`

Y luego:

% `cd gumstix-buildroot`

% `make defconfig`

% `make menuconfig`

% `make`

Aquí ya tendrá instalado el buildroot.

Por mayor información sobre este procedimiento, consulte:

<http://docwiki.gumstix.org/index.php/Buildroot>

Por información posterior en como crear y compilar programas vaya a :

http://docwiki.gumstix.com/index.php/Software_development_kit

http://docwiki.gumstix.com/index.php/Gumstix_programming

http://docwiki.gumstix.com/index.php/Sample_code

3. Inicialización de la Comunicación entre el Gumstix y el PC

Para conectarse al Gumstix lo primero que hay que hacer es:

Luego de que se tiene la plataforma Linux ejecutándose abrir una terminal en la misma. Hay 2 formas previstas para conectarse, una es a través de un cable que se conecta entre el Twener y un puerto serie de la PC y la otra es usando el protocolo WI-FI entre la placa wifi stix y un router wireless.

Si se elige el primer método se debe realizar lo siguiente:

Conecte el cable entre el puerto serie de su PC y el conector serie del gumstix. Siga las directrices de las siguientes páginas:

- En Linux:

http://docwiki.gumstix.org/index.php/Connecting_via_Serial_-_Linux

- En Windows (ya que también será posible conectarse desde este sistema operativo):

http://docwiki.gumstix.org/index.php/Connecting_via_Serial_-_Windows

En cambio si se opta por una conectividad inalámbrica lo que se debe hacer es:

Primero que nada debe conocerse la dirección IP que el router wireless le a asignado al Gumstix, sabiendo esta puede verificarse escribiendo en el internet browser la dirección IP del Gumstix, si todo va bien debería aparecer la siguiente pantalla:



Figura 1. Pantalla que aparece al estar conectado su PC con el Gumstix.

El siguiente paso sería abrir una ventana terminal, esto se hace en el menú:

Aplicaciones ➔ Accesorios ➔ Terminal

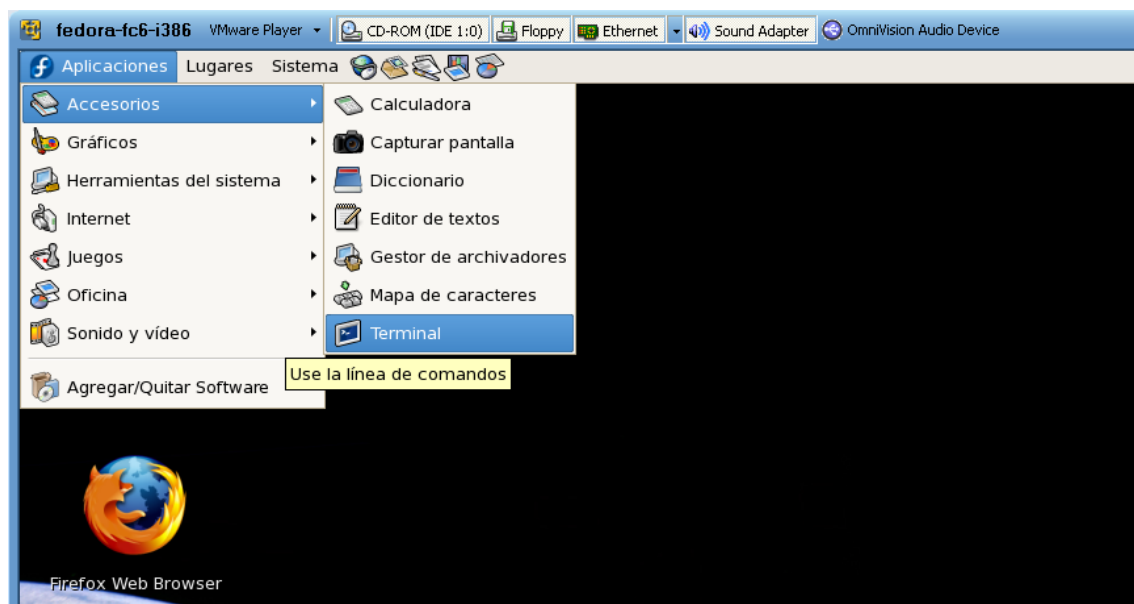


Figura 2. Como abrir una terminal en Linux Fedora.

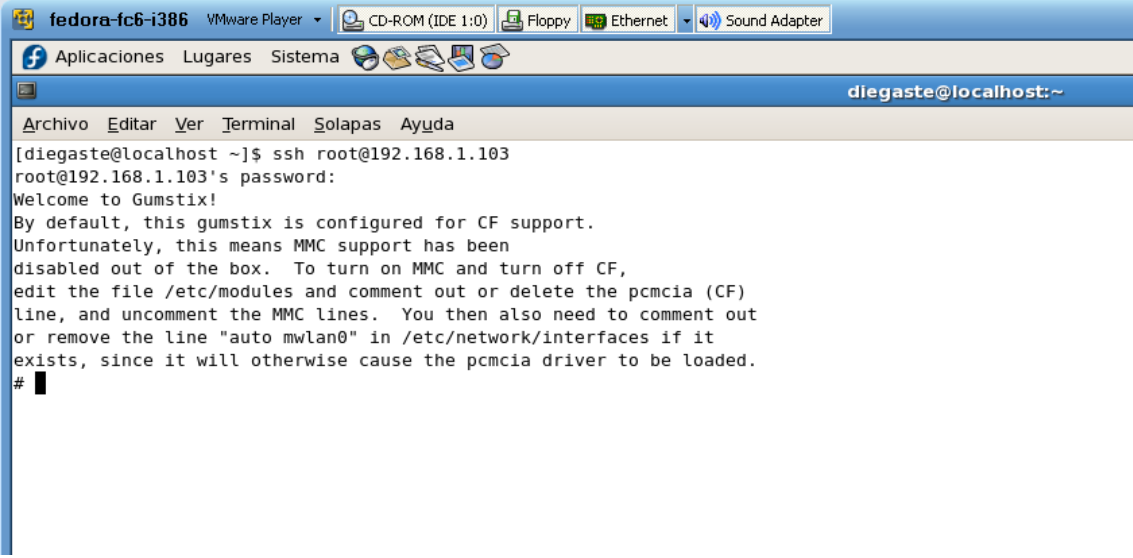
Una vez abierto el terminal, para conectarse al Gumstix, debe escribirse en el prompt:

```
$ ssh root@[dirección IP del Gumstix]
```

y dar enter.

Enseguida se le pedirá el password, el cual por defecto es: Gumstix.

Si el procedimiento fue correcto se debería visualizar la siguiente pantalla:

The image shows a terminal window titled 'fedora-fc6-i386' running inside a 'VMware Player'. The terminal displays the command 'ssh root@192.168.1.103' and the subsequent login process. The user 'root' is prompted for a password, and upon successful login, a 'Welcome to Gumstix!' message is shown. The message explains that by default, Gumstix is configured for CF support, but MMC support has been disabled. It provides instructions on how to enable MMC support by editing the file '/etc/modules' and commenting out or deleting the 'pcmcia (CF)' line, and also to comment out or remove the line 'auto mwlan0' in '/etc/network/interfaces' if it exists, to prevent the pcmcia driver from being loaded. The terminal ends with a '#' prompt and a cursor.

```
fedora-fc6-i386 VMware Player
Aplicaciones Lugares Sistema
diegaste@localhost:~
Archivo Editar Ver Terminal Solapas Ayuda
[diegaste@localhost ~]$ ssh root@192.168.1.103
root@192.168.1.103's password:
Welcome to Gumstix!
By default, this gumstix is configured for CF support.
Unfortunately, this means MMC support has been
disabled out of the box. To turn on MMC and turn off CF,
edit the file /etc/modules and comment out or delete the pcmcia (CF)
line, and uncomment the MMC lines. You then also need to comment out
or remove the line "auto mwlan0" in /etc/network/interfaces if it
exists, since it will otherwise cause the pcmcia driver to be loaded.
#
```

Figura 3. Terminal donde se observa lo que aparece en el momento de conectarse al Gumstix por SSH para trabajar en él.

En este caso el router le había asignado al Gumstix la dirección IP: 192.168.1.103 . De aquí en más se estará trabajando en el sistema operativo del Gumstix, siempre y cuando se opere en esta ventana terminal.

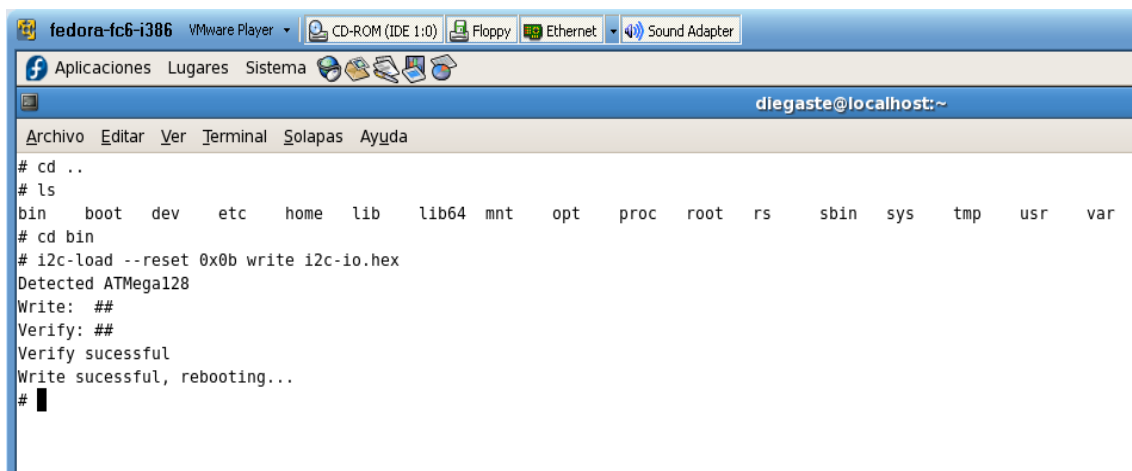
4. Inicialización de la Comunicación entre el Gumstix y el Robostix

El Gumstix y el Robostix se comunican por i2c, para esto hay que inicializar la comunicación. Para inicializar la misma lo que se hace es cargar una especie de bootloader al Robostix desde el Gumstix el cual contiene todas las directivas para comunicarse via i2c, como este se carga en la memoria volátil del atmega128 (micro-controlador del Robostix) esta operación debe hacerse cada vez que se energiza el Gumstix y Robostix.

Esto se hace en la hiperterminal donde esté trabajando en el Gumstix instanciando los siguientes comandos:

```
# cd ..
# cd bin
# i2c-load - reset 0x0b write i2c-io.hex
```

Lo que usted vería en pantalla si la operación es exitosa, se muestra a continuación:



```
fedora-fc6-i386 VMware Player
Aplicaciones Lugares Sistema
diegaste@localhost:~
Archivo Editar Ver Terminal Solapas Ayuda
# cd ..
# ls
bin boot dev etc home lib lib64 mnt opt proc root rs sbin sys tmp usr var
# cd bin
# i2c-load --reset 0x0b write i2c-io.hex
Detected ATMegal28
Write: ##
Verify: ##
Verify successful
Write successful, rebooting...
#
```

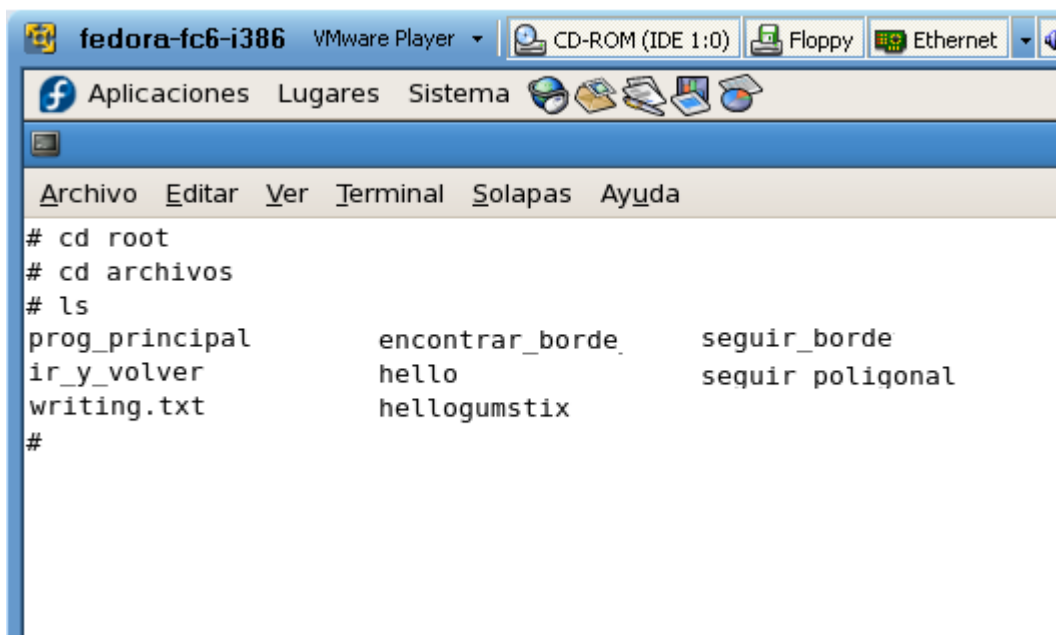
Figura 4. Cargado exitoso de la comunicación i2c entre el Gumstix y el Robostix.

Para verificar que este paso fue satisfactorio además de ver los mensajes de verificación luego de ejecutar el comando puede verse que un led rojo ubicado en el Robostix parpadea mientras la comunicación esté establecida.

Vale aclarar que el bootloader se puede cargar sin necesidad del Gumstix, la desventaja de esto es que se necesita hardware adicional

5. Inicialización del funcionamiento del Pez

Una vez realizada la inicialización de la comunicación entre el Gumstix y el Robostix, vaya al directorio que contiene los programas, este se encuentra en una carpeta creada en el directorio root, en el ejemplo de la siguiente figura esta carpeta se llama archivos.



```
fedora-fc6-i386 VMware Player
Aplicaciones Lugares Sistema
Archivo Editar Ver Terminal Solapas Ayuda
# cd root
# cd archivos
# ls
prog_principal encontrar_borde seguir_borde
ir_y_volver hello seguir poligonal
writing.txt hellogumstix
#
```

Figura 5. Programas desarrollados en este proyecto instalados en el Gumstix.

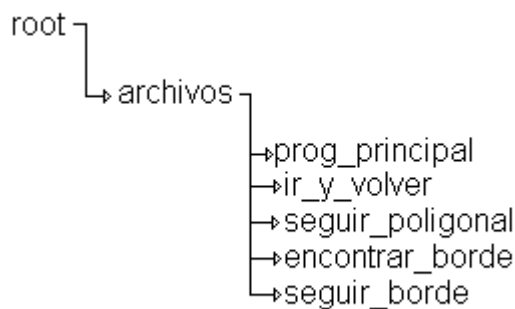
El comando ls muestra todos los archivos que se encuentran en la carpeta es equivalente al comando DIR en MS-DOS

Ejecute el programa principal. Para ejecutar programas en este entorno se debe estar parado en la carpeta de donde existe este archivo y se debe tipear ./nombre_del_archivo, que en el ejemplo sería: ./prog_principal que es el que enciende el pez, luego de esto debería empezar a funcionar.

Para apagarlo, simplemente apriete CTRL+C en la hiperterminal donde este corriendo los programas en el gumstix.

6. Descripción de los programas disponibles en el Gumstix

Dentro del Gumstix se encuentran una serie de programas los cuales se describirán a continuación en orden de jerarquía:



- **root**
Es la carpeta incluida por defecto en el Gumstix donde se corren las aplicaciones.
- **archivos**
Es la carpeta que contiene todos los programas (compilados) escritos en lenguaje C para el funcionamiento del pez.
- **prog_principal**
Este es el programa que se encarga de administrar las tareas en cuanto a sus tiempos y probabilidades de transición entre una tarea y otra.
- **ir_y_volver**
Esta es una de las tareas del pez cuya función es nadar derecho un determinado tiempo o hasta que el mismo perciba un objeto cerca, cuando se cumpla una de las 2 condiciones el pez efectuara un giro de casi 180 grados (muy pronunciado), una vez culminado el giro comenzará a nadar derecho hasta que se cumpla cualquiera de los 2 eventos antes mencionados.

- **seguir_poligonal**
En esta tarea el pez nada derecho hasta que el mismo encuentra un obstáculo, en ese instante da un giro con determinado ángulo (programado) para uno de los 2 lados dependiendo la situación una vez culminado el giro comienza a nadar derecho hasta que se cumpla la condición de acercamiento a un objeto.
- **encontrar_borde**
Aquí el pez nada en una dirección aleatoria hasta que encuentra un borde; una vez conseguido el objetivo culmina la tarea.
- **seguir_borde**
El pez comienza a nadar derecho hasta que el mismo encuentra un borde, una vez que pasa esto el mismo se mantiene nadando cerca del borde a una distancia configurable.

7. Despliegue de información en tiempo real en la PC

Para el despliegue de datos, ejecute el programa `desp_script` en su entorno Linux o Windows, y luego el programa: `despliegue_de_datos_RT`.

El programa `desp_script` copia periódicamente a la PC el archivo de datos generado también periódicamente por el Gumstix con la información del estado actual del pez. Este archivo es un archivo `.dat` capaz de ser leído por nuestra aplicación para el despliegue de datos, que en este caso es el programa `despliegue_de_datos_RT`. Otra tarea que realiza `desp_script`, es concatenar la versión nueva del `.dat`, llevando un historial de todos los estados por los que pasa nuestro robot. Este programa es simplemente un script (de Linux o Windows dependiendo de que entorno usted use, ya que hay dos versiones) que utilizando `scp` realiza lo anterior.

El programa `despliegue_de_datos_RT` es un programa que despliega la información del archivo `.dat` creada por `desp_script`. Posee una simple interfaz gráfica que permite comenzar con el despliegue, pausar el despliegue, borrar el historial de datos, o parar la ejecución del programa. Cuando la ejecución está detenida, este programa también nos brinda la posibilidad de graficar la evolución histórica de nuestro sistema.

Para que el despliegue de datos en tiempo real funcione correctamente, realice siempre las siguientes tareas en el orden detallado a continuación:

- i. Ejecute el programa `desp_script`.
- ii. Ejecute el programa `despliegue_de_datos_RT` y presione borrar.
- iii. Ahora el programa de despliegue de datos estará funcionando correctamente.

El estado con el que empieza el programa de despliegue de datos es en pausa.

Para comenzar a visualizar datos presione **PLAY**, para pausar la visualización **PAUSE** y para parar **STOP** (finaliza la ejecución del programa). Para borrar el archivo que contiene la evolución histórica de los datos presione **BORRAR** (este botón junto con

DESP solo aparecerá cuando la visualización este en pausa, estarán localizados en una nueva ventana del programa que aparecerá). Para desplegar el historial de los datos, presione PAUSE y luego el botón DESP.

A continuación se muestra una imagen de la ventana correspondiente al programa en play.

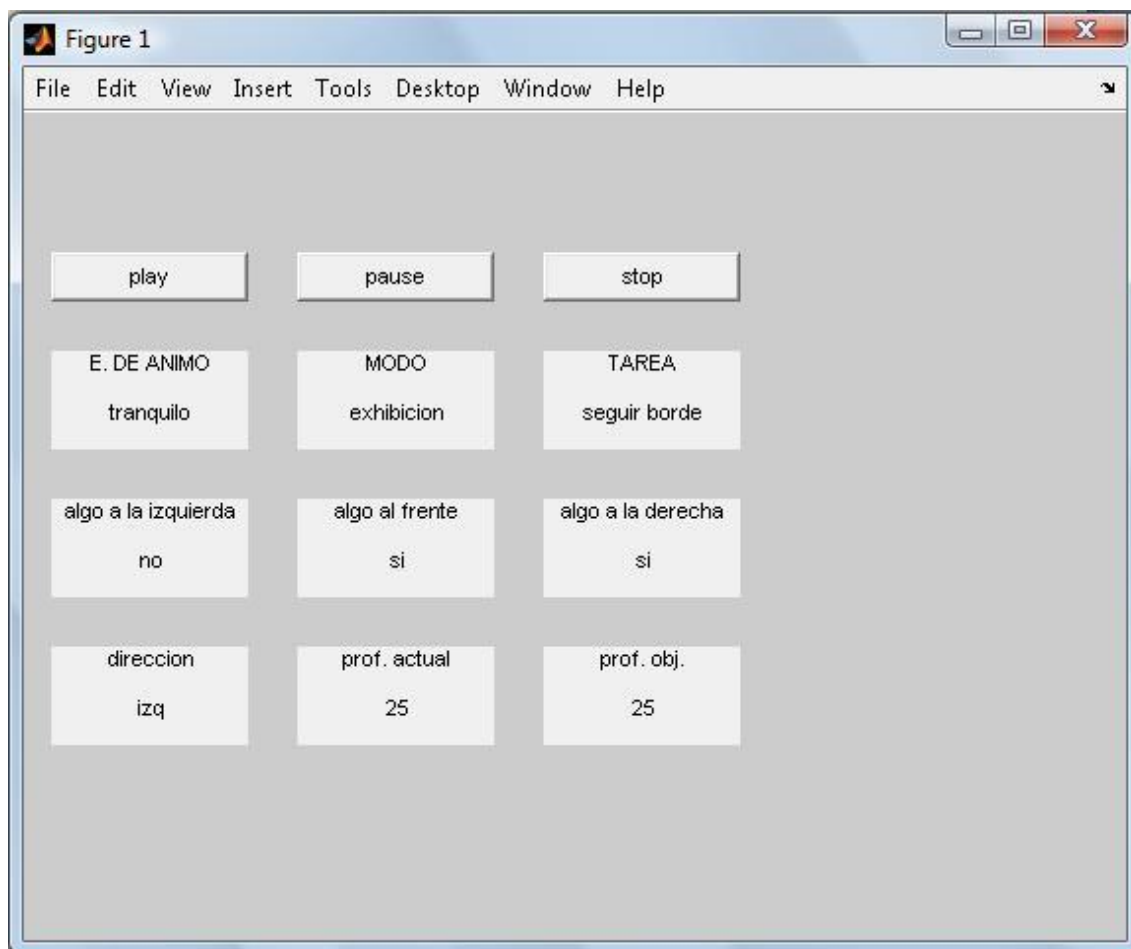


Figura 6. Interfaz gráfica del programa de despliegue de datos.

CAPITULO 3

Implementación final y discusión de pruebas y resultados.

1. Introducción

Entramos aquí a la última etapa de este proyecto, que concierne a los ensayos finales, discusión de pruebas, resultados obtenidos y conclusiones.

Este capítulo se ordena de la siguiente manera. Se comentan brevemente en qué consiste cada uno de los ensayos a realizar, tanto fuera del agua como finalmente con el pez sumergido en ella, detallando qué datos se extraen de los mismos y precauciones a considerar. Inmediatamente se muestran distintos gráficos ilustrativos de los movimientos simples del pez en agua, como también de distintas tareas.

Se atiende luego la aleatoriedad de la decisión de las tareas realizadas, citando un caso real de transición de tareas, que muestra el carácter autónomo del pez, parte objeto de este proyecto.

Finalmente, presentamos una discusión de los resultados obtenidos, conclusiones y miras hacia el futuro de distintas aplicaciones y posibilidades de expansión de este proyecto.

2. Implementación

La implementación final, la cual fue construida por nuestro tutor, consta de tres servomotores (dos para controlar los ángulos de las articulaciones de la cola y el tercero para controlar el nivel de profundidad). También cuenta con un sensor de presión, cinco de IR, y el Gumstix-Wifistix-Robostix.

A continuación se muestra una fotografía del sistema de movimiento de la cola, donde se observan los servos y las articulaciones.

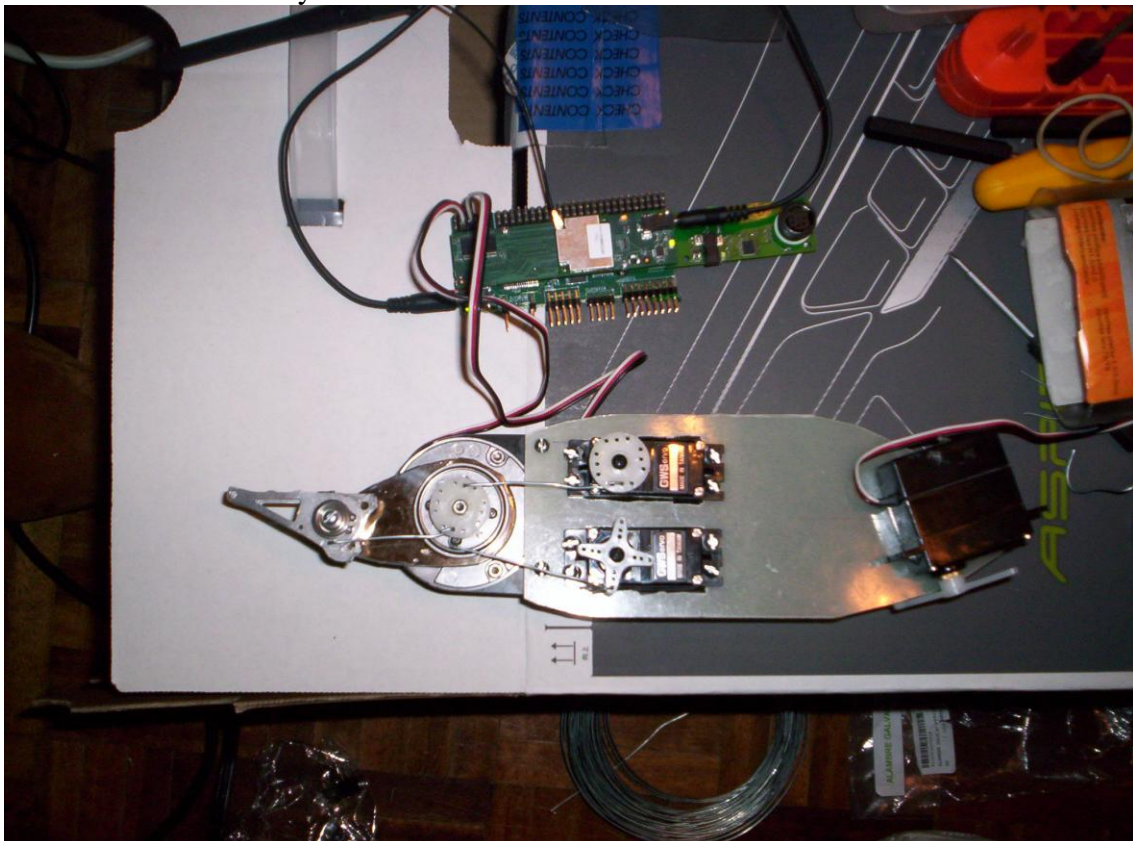


Figura 1. Sistema de movimiento

3. Ensayos

Una vez realizada la construcción mecánica y todos los programas necesarios para comandar remotamente el pez, se procede estudiando la correcta respuesta de los sensores a distintos estímulos.

Dado el tiempo que se requiere para poner en marcha el pez bajo el agua para cada uno de los siguientes experimentos, y las dificultades inherentes a este medio, se prefiere realizar en una primera instancia pruebas simples en aire.

3.1. *Primeras pruebas*

Las primeras pruebas fuera de agua se realizan en un medio controlado y buscan estudiar el correcto funcionamiento de la lógica de los programas anteriormente realizados. Vale observar que el funcionamiento de los sensores fue analizado en etapas previas, habiéndose obtenido exitosas respuestas.

Separemos las primeras pruebas en dos subgrupos: las simples y las compuestas. Las pruebas simples actúan directamente sobre un solo elemento, sea un sensor o bien un servomotor, mientras que el resto de los elementos se mantienen sin estímulos, en un medio controlado. Las compuestas refieren a la actuación sobre más de uno de los elementos, componiéndose entonces de respuestas simples.

Dada la simplicidad de estas pruebas, entendemos un medio controlado como aquél en el que es posible controlar la cantidad de luz incidente como la distancia con un objeto opaco en el caso de los sensores de distancia. A los efectos de estudiar los servomotores es suficiente eléctrica disponible y una mesa rígida.

Las pruebas simples aplicadas a los sensores de distancia exponen a uno de ellos y por turnos, a un objeto opaco (un libro de tapa dura y oscura), mientras que el resto se dispone de forma que el cono de sensibilidad de cada uno de ellos no perciba ningún objeto. Esto nos permite entonces por ejemplo “tapar” el sensor izquierdo, esperando como respuesta un giro hacia la derecha (el objeto a la izquierda simula una pared u obstáculo en general). Por otro lado, al tapar el sensor frontal mientras los demás sensores no actúan, estudiamos la lógica de giros (esperando en algunos casos giro hacia la izquierda y en otros a la derecha, chequeando el buen uso de una función random dentro de los programas). Para el caso de los servomotores estudiamos la aceleración de estos en distintas situaciones, al variar las tareas y en forma cualitativa.

Luego, las pruebas compuestas consisten en la aplicación de estímulos a más de un elemento, como puede ser tapar los sensores frontal e izquierdo; debemos esperar aquí un giro hacia la derecha. De manera análoga se realizan múltiples pruebas tapando algunos de los sensores según distintas situaciones (al variar las tareas).

3.2. *Pruebas en agua*

Hasta el momento nada dijimos del control de profundidad. Tomamos en cuenta las lecturas del sensor de presión para a partir de ellas realizar el control de profundidad

(mediante un controlador PI realizado por software). También, previamente se agregan pesas cuya posición se determina empíricamente según la dirección de la deriva. Las mismas procuran el equilibrio del pez según la dirección vertical. Una vez obtenido este equilibrio se procede analizando la máxima profundidad a la que se sumerge el pez.

Estas pruebas siguen inicialmente un paralelismo con las primeras pruebas mencionadas anteriormente bajo el agua, agregándose múltiples pruebas de la buena realización de distintas tareas como también análisis cuantitativo de giros, radios de curvatura y tiempos de reacción, complementados con diversas gráficas y videos.

Inicialmente por ende se procede colocando al pez en las proximidades de un borde y chequeando si el giro es el apropiado. Verificamos nuevamente entonces que la lógica de los programas es la correcta, aunque ahora bajo agua, algo cuestionable en un principio pues en este medio el nivel de brillo de los objetos es diferente, como también la presencia de partículas que se traducen en ruido en la lectura de los sensores. También analizamos a continuación la velocidad de los servomotores, mediante videos que permitan describir la posición en cada tiempo.

A continuación se estudia el primer movimiento simple, que consiste en el nadado rectilíneo, y se estudia en función de la frecuencia que le imponemos a la cola. Disponemos entonces el pez a distancias razonables de los bordes y estudiamos su evolución a diversas frecuencias. Múltiples conclusiones se pueden obtener a partir de estas observaciones siendo la más importante que la velocidad terminal del nado es directamente proporcional a la frecuencia. Este ensayo nos permite ajustar los parámetros de frecuencia de las distintas tareas.

Luego realizamos un ensayo de radio de giro en función del offset de la cola y la frecuencia de las oscilaciones. La observación principal que concluimos es que a mayor offset se obtiene un ángulo de giro mas pronunciado a frecuencia constante. Esto también nos permite ajustar los intervalos de los desfases angulares a imponer al girar en las distintas tareas en cada uno de los modos.

Posteriormente se realizan ensayos para cada una de las tareas programadas, dedicando la atención tanto a la correcta evolución de las mismas como su prolongación en el tiempo. La cual es satisfactoria.

Finalmente se realiza una prueba general del sistema observando la correspondencia entre la evolución del sistema y lo esperado. Obteniéndose también resultados satisfactorios.

4. Conclusiones

En todo desarrollo de robótica se tienen elementos generales que incluyen el estudio de estímulo respuesta mediante el agregado de sentidos (sensores) al robot, una comunicación remota, y toma de datos y decisiones. La plataforma desarrollada logra satisfactoriamente el carácter autónomo, puesto que la trayectoria del pez a seguir es desconocida a priori y realizada sin interferencia del usuario, y las decisiones del movimiento son por lo tanto tomadas en forma no determinística.

También hemos logrado satisfactoriamente imprimir la conducta y realización de tareas estudiadas en el pez, muchas ideas en relación a esto fueron agregadas en el desarrollo del proyecto.

Es interesante destacar la característica inicial de esta plataforma de pez a futuras aplicaciones, en las que se podrá incluir nuevos sensores y sistemas mas sofisticados de conducta, así como también control de otros parámetros del movimiento para convertir en más realista a esta plataforma (por ejemplo aletas).

Nuestra plataforma finaliza con un sistema visual limitado y un sentido de la profundidad, elementos suficientes para nadar en regiones de geometría desconocida. También cuenta con una interfaz amigable con el usuario, y la posibilidad de ser accedida de forma remota desde una red.

Con respecto a la evolución del proyecto, si bien tuvimos muchos retrasos muchas veces por circunstancias ajenas a nuestro dominio, al final se pudo concluir con los requisitos impuestos al inicio de este proyecto.

Vale la pena recalcar que durante la evolución del proyecto tuvimos que aprender muchas cosas nuevas, como la programación en lenguaje C, el control de motores y las características de varios sensores.

5. Futuras aplicaciones y trabajo a futuro

En cuanto a agregado de electrónica o equipamiento, creemos que se pueden agregar aletas pectorales con motores que las comanden, y programar el control respectivo. También se pueden colocar sensores diversos, por ejemplo un sensor de luminosidad así el pez va hacia la luz, sensores de electro-location, etc. Otro agregado posible aprovechando el potencial del Gumstix es la captura de video que permitiría sistemas de toma de decisiones más complejos.

En cuanto a agregado de conductas, una futura cosa a implementar podría ser la capacidad de aprendizaje automático. También se puede realizar un sistema seguidor con diferentes elementos a seguir, por ejemplo un alimento, una presa, otro pez, la luz, etc. Esto ampliaría la similitud del pez a uno real.

BIBLIOGRAFIA

- [1] Liu J. and Hu H., “Development of Fish-like Swimming Behaviours for an Autonomous Robotic Fish”, IEE Proceedings of the Control 2004. URL: <http://privatewww.essex.ac.uk/~jliua/paper/JindongControl2004-Final.pdf>, [Consulta: 09 de Mayo de 2008]
- [2] Hirata K., “Development of Experimental Fish Robot”, Proceedings of ISME 2000, Tokio Japan, Vol. 2, pp.711-714.
- [3] Leonard J., Bennett A., Smith C. and Feder H. “Autonomous Underwater Vehicle Navigation”, MIT Marine Robotics Laboratory Technical Memorandum 98-1. URL: http://cml.mit.edu/~jleonard/pubs/tech_report981.pdf, [Consulta: 09 de Mayo de 2008]
- [4] Lachat D., “BoxyBot, the fish robot Design and realization”, EPFL - Semester Project, 27 June 2005. URL: <http://birg.epfl.ch/page63000.html> [Consulta: 09 de Mayo de 2008]
- [5] Rother D., “Simulación de Imágenes Eléctricas en Peces Eléctricos de Descarga Débil”, tesis de maestria, Facultad de Ciencias-UDELAR, Abril 2003. URL: http://iibce.edu.uy/neurofisiologiacomparada/dr_thesis.html [Consulta: 09 de Mayo de 2008]
- [6] Budelli R. and Caputti A., “The electric image in weakly electric fish: perception of objects of complex impedance”, The Journal of Experimental Biology (2000); Vol. 203: pp. 481–492.
- [7] MacIver M., Fontaine E., and Burdick J. “Designing Future Underwater Vehicles: Principles and Mechanisms of the Weakly Electric Fish”, IEEE Journal of Oceanic Engineering (2004); Vol. 29, No. 3: pp. 651-659.
- [8] Rother D, Migliaro A, Canetti R, Gomez L, Caputi A, et al., “Electric images of two low resistance objects in weakly electric fish”, Biosystems (2003); Vol. 71: pp. 169–177.
- [9] Freescale Semiconductor, “Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated”, Technical Data, Document Number: MPXV7002, Rev 0, 09/2005.
- [10] Opto-Electronics Devices Division, Sharp Electronics, “Sharp distance sensor general application note”, Feb. 18, 20003.
- [11] Perez D., “Sensores de distancia por ultrasonido”, trabajo del seminario: diseño y construcción de microrobots, Universidad de Alcalá de Henares. URL: <http://www.alcabot.org/seminario2006/Trabajos/DiegoPerezDeDiego.pdf> [Consulta: 09 de mayo de 2008]
- [12] Super Robotica, “SFR04 Sensor de distancias por Ultrasonido”. URL: <http://www.superrobotica.com/S320110.htm> [Consulta: 09 de mayo de 2008]
- [13] Todo Robot, “El Servomotor”, proyecto: Construyendo un Servo Mecanismo, diciembre de 2000. URL: <http://www.todorobot.com.ar/documentos/servomotor.pdf> [Consulta: 09 de mayo de 2008]
- [14] Colaboradores de Wikipedia, "Servomotor", Wikipedia, La enciclopedia libre. URL: <http://es.wikipedia.org/wiki/Servomotor> [Consulta: 09 de mayo de 2008].
- [15] Colaboradores de Wikipedia, "Pulse-width modulation," Wikipedia, The Free Encyclopedia, URL: http://en.wikipedia.org/w/index.php?title=Pulse-width_modulation [Consulta: 09 de Mayo de 2008].
- [16] Carletti, E., “Servos, características básicas”. URL: http://robots-argentina.com.ar/MotorServo_basico.htm [Consulta: 09 de Mayo de 2008].
- [17] Anderson, E., “Submergency: An antonomous Submarine”, Final Paper, University of Florida, Intelligent Machine Design Lab. URL: www.mil.ufl.edu/imdl/papers/IMDL_Report_Fall_98/Eric_Anderson/Submergency.pdf [Consulta: 09 de Mayo de 2008]

- [18] King, J., "The Development of an AUV", Msc. Thesis, University of Waikato, New Zealand, 2002. URL: www.victoria.ac.nz/scps/research/mechatronics/jennifer_king.PDF [Consulta: 09 de Mayo de 2008]
- [19] Gerl, B. "Development of an Autonomous Underwater Vehicle in an Interdisciplinary Context", diploma thesis, Technische Universität München and The University of Western Australia, 2006. URL: robotics.ee.uwa.edu.au/theses/2006-AUV-Gerl.pdf [Consulta: 09 de mayo de 2008].
- [20] Lachat, D., Crespi, A., and Ijspeert, A. "BoxyBot: a swimming and crawling fish robot controlled by a central pattern generator" The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. February 20-22, 2006. pp. 643- 648.
- [21] Tzeranis D., Papadopoulos E., and Triantafyllou, G. "On the Design of an Autonomous Robot Fish". Proc. of the 10th IEEE Mediterranean Conference on Control and Automation, Rhodes, 2003. URL: http://www.cim.mcgill.ca/~egpapado/pdf_ps/MED03.pdf [Consulta: 09 de Mayo de 2008]
- [22] Liu J., Hu H., and Gu D., "A Hybrid Control Architecture for Autonomous Robotic Fish" Proc. 2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Beijing, China, Oct. 9-15, 2006, pp. 312-317.
- [23] Liu J., and Hu, H. "A 3D simulator for autonomous robotic fish", Int. Journal Automation & Computing (2004); No.1: pp. 42-50.
- [24] Sfakiotakis M., et al., "Review of Fish Swimming Modes for Aquatic Locomotion", IEEE Journal of Oceanic Engineering (1999); Vol. 24, No.2: pp. 237-252.
- [25] Saimek S., and Perry Y.L., "Motion Planning and Control of a Swimming Machine", The International Journal of Robotics Research (2004); Vol. 23, No. 1: pp. 27-53.
- [26] Yu J., Wang S. and Tan M., "A simplified propulsive model of bio-mimetic robot fish and its realization", Robotica (2005); Vol. 23, pp. 101–107.
- [27] Dahlquist, G. Bjorck, A. and Anderson N., "Numerical Methods". Prentice-Hall, New Jersey, 1974.
- [28] Alonso, J.A., "Baterías de NiCd y NiMH", El Vuelo Eléctrico: Una guía simple para el diseño, construcción y vuelo de aeromodelos eléctricos. URL: <http://www.icmm.csic.es/jaalonso/velec/baterias/bateria.htm> [Consulta: 09 de mayo de 2008]
- [29] Perales, A., "Las Baterías", Página RC de Alfonso Perales. URL: <http://www.carbi.net/tecnica/newpage1.html> [Consulta: 09 de Mayo de 2008]
- [30] Gumstix, inc., página web de Gumstix. URL: <http://www.gumstix.com> [Consulta: 09 de Mayo de 2008]
- [31] Rodríguez, B. "Notas teóricas del curso Antenas y Propagación".
- [32] "Radar de Subsuelo: Evaluación para aplicaciones en arqueología y patrimonio histórico- artístico", Capítulo 9 - Ensayos en agua.
- [33] Documento de presentación de Teledata Comunicaciones por plan Ceibal. "Cálculos de enlace entre APs y notebook WiFi compatibles".