XXV Jornadas de Automática
Ciudad Real, del 8 al 10 de septiembre de 2004

MODELO DE ARQUITECTURA DE COMUNICACIONES FRAME-SENSOR-ADAPTER (FSA)

José Luis Poza Luján
Universidad Politécnica de Valencia, Camino de Vera s/n, Valencia 46022, jopolu@disca.upv.es

Juan Luis Posadas
Universidad Politécnica de Valencia, Camino de Vera s/n, Valencia 46022, jposadas@disca.upv.es

José Enrique Simó
Universidad Politécnica de Valencia, Camino de Vera s/n, Valencia 46022, jsimo@disca.upv.es

Pascual Pérez
Universidad Politécnica de Valencia, Camino de Vera s/n, Valencia 46022, pperez@disca.upv.es

Raúl Simarro
Universidad Politécnica de Valencia, Camino de Vera s/n, Valencia 46022, rausifer@upvnet.upv.es

Resumen

Para la comunicación entre componentes de un sistema, se debe tener en cuenta que es necesaria una interfaz común. En un sistema, basado en componentes distribuidos, hay dos áreas: los propios componentes y el sistema de comunicación. En lo que el modelo de comunicaciones respecta, es importante que pueda contemplar, de la forma más sencilla, todas las posibles necesidades de los componentes, distinguiendo cuándo los componentes son canales de comunicación o cuando son componentes de procesamiento. Para poder implementar el sistema en la arquitectura de un robot móvil se ha desarrollado un modelo de programación denominado FSA (Frame-Sensor-Adapter). Este artículo proporciona una descripción detallada del modelo de comunicaciones FSA y un ejemplo basado en este modelo.

Palabras Clave: Robótica, Sistemas Distribuidos, Comunicaciones.

1 INTRODUCCION

La gestión de las comunicaciones de un sistema distribuido implica el uso, por parte de diversos componentes, de uno o varios canales de comunicación. Por ejemplo, de la misma forma que un canal TCP puede ser usado para gestionar correo o visualizar páginas Web, un canal serie COM (línea serie, comunicación RS-232, RS-422 o similares) puede ser empleado para comunicar con un sensor láser, un servo que actúe sobre las ruedas de un robot, o componentes similares.

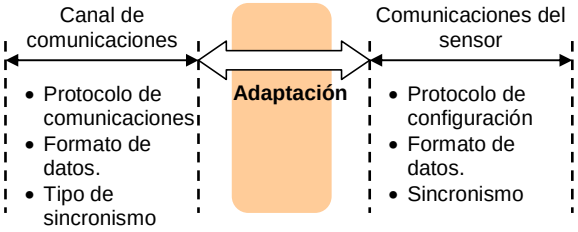

Figura 1: Adaptación de las comunicaciones

La gestión del canal de comunicaciones tiene la dificultad de unir dos sistemas con cierta rigidez, el sistema de comunicaciones con el que se está tratando tiene unas restricciones que se deben tener en cuenta; a su vez, el sistema que debe comunicarse también tendrá una rigidez en lo que a la información se refiera.

Ocultar las restricciones y los aspectos propios del protocolo, al resto del sistema es importante, sobre todo en las comunicaciones en tiempo real que se precisan a nivel de sensores en la navegación reactiva [2]. Pero también lo es adaptar la información del sistema al medio de comunicación (figura 1).

Sin embargo, la comunicación entre los diferentes componentes difiere en algunos aspectos de la comunicación con los canales. Entre los componentes toma mayor relevancia el contenido de la información transmitida, así como la posibilidad de sincronizar dicha información, especialmente cuando se está trabajando con una navegación basada en comportamientos [1]. Finalmente las relaciones entre componentes también suponen uno de los aspectos que, en el caso de un robot móvil, debe cubrir la arquitectura del sistema [3]. En este artículo se plantea la estructuración de los componentes citados de la comunicación.

2 MODELO DE ARQUITECTURA

2.1 CANAL DE COMUNICACIONES

La gestión de las comunicaciones en un sistema, implica un control del canal de comunicaciones y una adaptación de la información, que éste canal transmite, al dispositivo que vaya a hacer uso de dicho canal.

Como se puede ver en la figura 2, la parte de adaptación, y sincronismo se engloba en el ámbito del adaptador. Mientras que los sensores se interconectan entre ellos, de una forma jerárquica. Se considera comunicaciones al conjunto del canal físico y del adaptador. Los sensores reciben la información del canal de comunicaciones, o del sensor inmediatamente superior a él. Mientras que escriben en el canal de comunicaciones por medio del adaptador. Esto hace que la comunicación de una información, por parte del adaptador, sea sencilla. Un efecto de la jerarquía propuesta hace que cada sensor vea como un adaptador al sensor de nivel superior.

Un ejemplo de la jerarquía de sensores se puede ver en el caso de trabajar con el puerto de comunicaciones serie RS232 (figura 3). En este puerto, la mayoría de los sistemas operativos, proporcionan un acceso básico a nivel de byte. Por lo que el sensor más sencillo que se conecta al adaptador de comunicaciones es el sensor que recibe byte a byte la información del adaptador, descargando de éste toda responsabilidad en la generación de estructuras de información de mayor nivel (cadenas y similares). Por ello a este sensor se le denomina “ComByte”. La generación de estructuras de información más complejas será responsabilidad de sensores de niveles superiores.

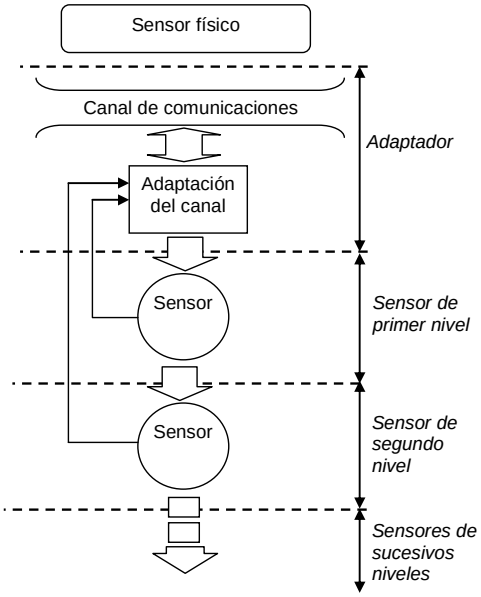

Figura 2: Jerarquía de las comunicaciones.

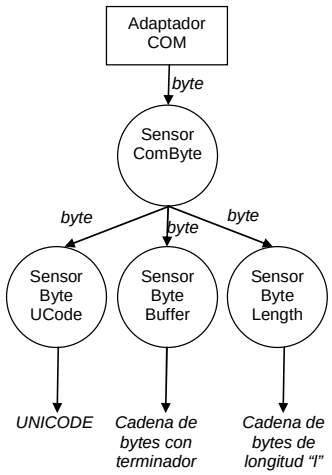

Figura 3: Ejemplo de jerarquía de comunicaciones

El sensor ComByte, puede tener conectados varios sensores a los que enviará un mensaje cada vez que se produzca la recepción con el byte recibido. Los sensores que estén conectados a el pueden procesar dicha información de una forma más coherente y suministrarla en diversos formatos, como caracteres unicote, cadenas basadas en un terminador o cadenas de longitud fija (figura 3), sin necesidad de tener que cambiar de conexión con el puerto serie.

2.2 CONEXIONES BASICAS

Una primera aproximación a la comprensión del modelo FSA consiste en buscar las combinaciones posibles. En la figura 4 se pueden ver las combinaciones posibles.

En la figura 4.a se puede ver la conexión sencilla de un sensor y un adaptador, este modelo es el más sencillo (por ejemplo, la conexión de un sensor de ultrasonidos por un puerto COM). En la figura 4.b se puede ver la conexión de dos sensores con un sólo adaptador (por ejemplo dos sensores de ultrasonidos que comparten un canal). Este caso sólo se puede dar en caso de canales de comunicación compartidos. El siguiente de los casos (figura 4.c) consiste en un sensor conectado a dos adaptadores (por ejemplo un sensor que transmite por un puerto COM, pero comunica la información también por una conexión TCP para, por ejemplo, monitorizar la señal), este caso sólo se puede dar si el sensor tiene capacidad de conectarse a distintos medios de forma simultánea.

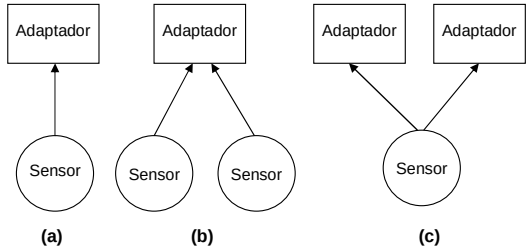

Figura 4. Combinaciones del modelo FSA.

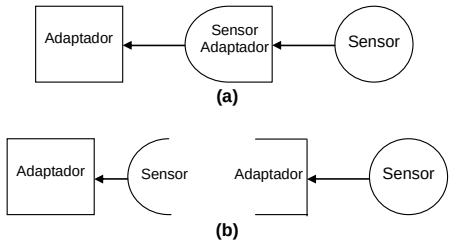

Figura 5. Combinación de modelo mixto FSA.

A partir de los tres modelos anteriores, se pueden hacer combinaciones de la complejidad deseada, siempre que los sensores estén preparados para soportar el sistema de comunicaciones correspondiente.

2.3 EXPANSIÓN DEL MODELO

En ocasiones se hace necesario hacer un procesamiento previo al acceso, por parte del sensor, al medio de comunicación a través del adaptador, o bien un procesamiento posterior en el sentido contrario (Por ejemplo, si se desea comunicar con un adaptador COM orientado al envío y recepción de caracteres, pero el sensor está orientado al envío y recepción de cadenas).

Este procesamiento queda en un lugar indefinido entre el sensor y el adaptador. Si la unidad, o el tipo de información, que el sensor espera son diferentes de la unidad, o tipo de información, que el adaptador procesa, deberá existir un componente intermedio que negocie esta comunicación (figura 5.a). Este tipo de componente es visto desde el sensor como si fuese el adaptador de comunicaciones, sin embargo, desde el adaptador este componente es visto como un sensor, el esquema se puede ver en la figura 5.b. Este modelo más complejo puede extenderse todo lo necesario según los requerimientos o el nivel de procesamiento, previo o posterior, que requiera la información desde que sale de un sensor determinado hasta que llega al adaptador, o en el sentido contrario.

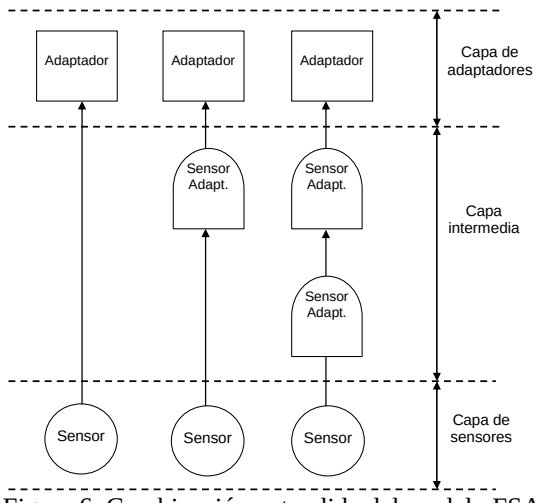

Figura 6. Combinación extendida del modelo FSA.

En la figura 6 se puede apreciar el árbol de combinaciones. Cabe destacar que un sensor puede conectarse a cualquier componente intermedio sin problemas, ya que la visión que tiene del mismo es la de un adaptador. Por medio de combinaciones de estos componentes se puede formar la arquitectura de comunicación-procesamiento de la información más conveniente.

La característica de expansión que proporciona la conexión de sensores entre ellos hace que la escalabilidad del componente esté asegurada. El único problema que puede existir es la aparición de información recurrente, o lo que es lo mismo, primero un modelo como el de la figura 4.b y luego como el de la figura 4.c, teniendo en cuenta que los sensores de la figura 4.b son vistos como adaptadores por el sensor de la figura 4.c.

3 DEFINICIÓN FSA

3.1 ADAPTADOR

La clase base del adaptador es *IAdapter.h*, en esta clase se definen los métodos básicos de la comunicación La clase base del adaptador no tiene propiedades definidas. Sin embargo, para la gestión de las comunicaciones suele ser necesarias, como mínimo, algunas propiedades, que en futuras versiones pueden estandarizarse ya que por medio de la herencia se pueden ir extendiendo patrones más complejos de definición de adaptadores, sensores o clases mixtas.

Generalmente las comunicaciones tienen que ser configuradas en función de los requerimientos del sistema (paridad, sincronismo, velocidad y aspectos similares). Para ello se definen los métodos descritos en la tabla 1.

Connect	Inicia la conexión con el canal físico de comunicaciones. Pero no inicia la comunicación con los sensores.
Disconnect	Desconexión con el canal físico de comunicaciones.
Start	Inicia los sensores conectados al adaptador llamando a los métodos *Start* de los sensores que se hayan registrado.
Stop	Detiene los sensores conectados, llamando a los métodos *Stop* de cada uno de los sensores registrados.
RegisterSensor	Registra un sensor para poder ser avisado cuando las comunicaciones tengan vayan dirigidas a el.
Read	Lee directamente del adaptador, sin esperar a que este envíe un mensaje.
Write	Se emplea para escribir información en el adaptador.

Tabla 1. Métodos del adaptador

OnStart	Función ejecutada cuando se inicializa un adaptador en el que está registrado el sensor. Con el parámetro se indica el adaptador desde donde se ejecuta la función.
OnStop	Función ejecutada cuando finaliza un adaptador en el que está registrado el sensor. Con el parámetro se indica el adaptador desde donde se ejecuta la función.
OnMessage	OnMessage: Función ejecutada cuando un adaptador en el que está registrado el sensor recibe un nuevo valor (buffer de bytes) para un dato al que está conectado el sensor. Con los parámetros se indican el buffer de bytes, su tamaño, el nombre del dato y el adaptador desde donde se ejecuta la función.

Tabla 2. Métodos del sensor.

Hay algunos detalles que se deben apreciar. No existe un método que desregistre a un sensor de un adaptador. Esto se puede gestionar directamente con el método de Stop. Tampoco se encuentran los métodos de conexión y desconexión, sin tener que recurrir a Start y Stop, esto se hace por simplificar la interfaz básica.

3.2 SENSOR

Los sensores son los componentes encargados de enviar o recibir la información a través de un adaptador. Este envío y recepción por parte del sensor, hace que éste deba conocer el adaptador desde el que ha sido iniciado, aunque puede ser que un sensor se conecte a varios adaptadores, pero esa gestión debe ser interna del sensor, o del componente intermedio al que se conecta el sensor; por ello en el método en el que se recibe un mensaje, se debe enviar al sensor la referencia del adaptador que ha generado el mensaje. Los métodos que dan soporte al sensor se pueden ver en la tabla 2.

Estos métodos son los necesarios para una relación síncrona con los adaptadores. La conexión asíncrona se puede obtener por medio de los métodos de *Read* y *Write* de los adaptadores.

3.3 MARCO

El marco se define como el contenedor básico de los sensores y los adaptadores. Es un objeto que proporciona a la aplicación la interfaz necesaria para la gestión conjunta de los adaptadores y los sensores. El marco tiene su fundamento como enriquecedor del acceso a los adaptadores y a los sensores que se hayan definido en la aplicación. Los métodos que dan soporte al funcionamiento del marco se pueden ver en la tabla 3.

Start	Método que invoca a los métodos *Connect* y *Start* de los adaptadores, realizando las comprobaciones en la secuencia adecuada. Los adaptadores serán los que invoquen a los métodos *OnStart* de los sensores que estén conectados a ellos.
Stop	Función que detiene el sistema, invocando a los métodos *Disconnect* y *Stop* de los adaptadores, con las mismas restricciones que los

Tabla 3. Métodos del marco.

El marco se puede ver como un entorno que contiene todas las características peculiares del sistema en el que se han integrado diversos sensores y adaptadores, la funcionalidad que proporciona es la de particularizar el uso de cada adaptador y sensor al sistema. En el ejemplo del siguiente apartado se podrá comprobar mejor el uso que se le da a este componente.

4 EJEMPLO DE APLICACIÓN

Para ensayar el modelo FSA se ha construido una combinación de dos sensores físicos, un láser LMS200 de la empresa SICK [4] y un servo motor de construcción propia. El sistema se ha montado sobre el robot YAIR (figura 7) desarrollado en el propio grupo de investigación. El robot proporciona el entorno adecuado con diversos canales de comunicaciones para emplear (puertos serie, CAN, USB o red TCP) y diversos tipos de sensores (Láser, ultrasonidos o infrarrojos). Cabe destacar la futura necesidad de combinar la información de todos los sensores físicos, para dar soporte a la generación de mapas y a la navegación inteligente del robot.

Figura 7. Robot YAIR con el servo y el láser.

Figura 8. Láser SICK LMS200.

4.1 HARDWARE

4.1.2 Láser

El láser empleado es un modelo LMS200 de SICK que trabaja escaneando el contorno en dos dimensiones. La resolución que proporciona es de cerca de 10 milímetros, la máxima distancia que puede alcanzar es de 30 metros. El entorno escaneado está entre los 100 y 180º con una resolución que varía entre 0,5º y 1º. Funciona en la frecuencia de infrarrojos, por lo que no es visible y precisa de alimentación de 24 V. La información la transmite por medio de un puerto serie RS232 o por medio de un bus RS422. El tiempo de respuesta es de entre 13 y 56 milisegundos y la tasa de transferencia puede variar de 9.6 a 500 Kilo Baudios.

La comunicación del láser con los dispositivos se hace por medio de los telegramas (así los llama el fabricante), que consiste en un protocolo a dos bandas de mensajes orientados a campos. El protocolo se puede ver en la figura 9.

Cabe destacar que el protocolo de comunicaciones es más complejo de lo habitual, puesto que carece de carácter terminador y la longitud está codificada en el propio mensaje. Además el código indicador de inicio de trama puede aparecer como parte del campo data. Si casualmente se da el caso anterior, puede llevar a error la interpretación del mensaje, por lo que la programación de la comunicación es delicada.

STX	ADR	LEN	CMD	Data	CRC

STX: Indicador de inicio de trama (0x20)
ADR: Dirección destino.
LEN: Longitud del telegrama (en bytes)
CMD: Código de orden a realizar
Data: Parámetros de la orden a realizar
CRC: Código de redundancia cíclica.

ACK	STX	ADR	LEN	CMD	Data	STAT	CRC

ACK: Confirmación de la orden realizada
STX: Indicador de inicio de trama (0x20)
ADR: Dirección destino.
LEN: Longitud del telegrama (en bytes)
CMD: Código de orden realizada
Data: Datos obtenidos
STAT: Estado en el que se ha quedado el láser.
CRC: Código de redundancia cíclica.

Figura 9. Protocolo de comunicación del láser.

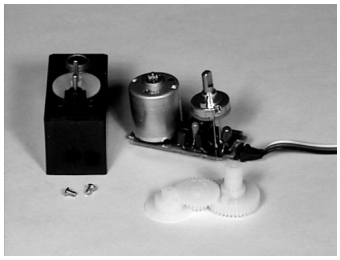

Figura 10. Servo motor.

4.1.2 Servo

La inclinación del láser se ha obtenido por medio de la inclusión de un armazón sobre el que el aparato pueda girar en un ángulo. Para ello ha sido necesario instalar un sencillo servo (figura 10) que facilite el cabeceo del láser.

El servo tiene dos visiones, como sensor y como actuador. Como sensor proporciona la posición en la que se encuentra, en este caso el ángulo de cabeceo del láser. Esta información es transmitida al puerto serie por medio de un microcontrolador ATMEL 89C51CC002. La información se transmite a una velocidad de 2400 bits por segundo de forma continua. Como actuador, recibe solicitudes de giro, para cambiar el ángulo de cabeceo. Las solicitudes se reciben en el puerto serie gestionado por el mismo microcontrolador que proporcionando un control PD dentro del microcontrolador, para asegurar la estabilidad en el ángulo.

El protocolo de comunicaciones empleado está orientado a campo, y consiste en un mensaje que contiene una sencilla cabecera (carácter “#”), para la sincronización, el ángulo codificado en un entero de 4 bytes y un terminador para añadir más seguridad (carácter “/”).

4.2 MODELO FSA

4.2.1. Adaptadores

Para la comunicación con el puerto serie se desarrolló un solo adaptador de comunicaciones, denominado “AdaptComByte”. Este adaptador puede ser configurado con todas las características de los puertos RS232 (puerto, velocidad, sincronización, etc.). Para cada uno de los sensores se creó un objeto del tipo AdaptComByte con los parámetros de puerto COM, velocidad, sincronización correspondientes al sensor físico con el que se comunicarán. Cabe destacar que ambos adaptadores trabajan obteniendo los bytes del puerto serie y escribiendo las secuencias solicitadas, por lo que hay una total independencia del protocolo del sensor físico al que puedan estar conectados, por lo que cumple los objetivos de flexibilidad del modelo FSA.

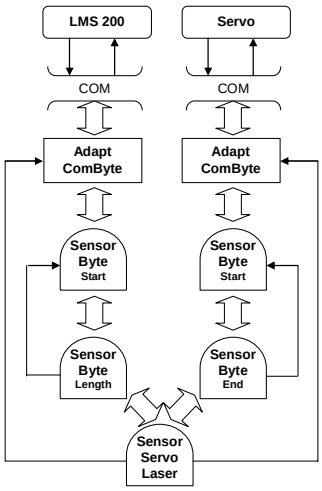

Figura 11. Modelo FSA del Servo - Láser.

4.2.2 Sensores

Para realizar la adaptación del sistema a las concreciones propias del protocolo de cada sensor físico, se desarrollaron varios sensores. El primero de ellos es el SensorByteStart, que hace de filtro. Es un sensor, similar a un flag, que recoge bytes y sólo los pasa si previamente el byte de cabecera se ha dado, en el caso del sensor que se conecta al adaptador del Láser, el byte es 0x20, en el caso del que se conecta al adaptador del servo es 0x23. De esta manera se obtiene la sincronización con las cabeceras. A cada uno de esos sensores se conecta otro que se encarga de montar la trama, en el caso del sensor láser, lo hace leyendo la longitud del mensaje (dada una posición en el número de byte que le van llegando) y el otro lo hace en función del byte del final de trama. Estos dos sensores son los que "avisan" a los sensores a los que están conectados (por medio del método Write) de la finalización de la trama (que envían al sensor conectado a ellos) y de su cambio a posición de escucha de una nueva trama. A los dos sensores anteriores se conecta el sensor ServoLáser que agrupa la información de las tramas que le llegan del servo y del láser y las envía a los sensores (o la aplicación) que esté conectada a él. Este sensor es el que proporciona, también, la posibilidad de enviar órdenes, convirtiéndolas en las tramas que entienden los sensores físicos y escribiéndolas en el adaptador correspondiente.

4.2.3 Marco

El ejemplo presentado es sencillo y, por tanto, sólo se incluye un marco donde se crean los objetos que representarán a los sensores y adaptadores descritos anteriormente. En el caso de disponer de dos sistemas Servo – Láser, bastaría con crear dos marcos, uno para cada sistema, donde los parámetros específicos de los sensores o de los adaptadores se definirían directamente en cada marco.

5 CONCLUSIONES

Se ha desarrollado un modelo de programación de componentes escalable y flexible que facilita el modelado de los sensores y actuadores en un robot móvil. Este modelo permite conectar cada componente a otros que le suministran información procesada y, a su vez, ellos facilitan la información a otros componentes. Este encadenamiento de secuencias de mensajes para transmitir la información, cada vez más elaborada, proporcionan al sistema una visión jerárquica de la información que este tiene.

El modelo se ha probado con un sistema sencillo compuesto de un láser y un servo, empleado para escanear un entorno y proporcionar los datos al resto del sistema. Actualmente el modelo de sensores presentado en este artículo se está empleando para la generación de mapas geométricos empleados en la navegación en robots móviles.

En un futuro se prevé emplear el modelo FSA para modelar todo el robot YAIR, de esta manera la obtención de la información de otras fuentes sensoriales hará posible la generación de mapas topológicos con información más rica del entorno. Se prevé explotar la posibilidad de actuar sobre el robot para transmitirle órdenes concretas que permitan realizar una navegación del mismo por un entorno cerrado.

Agradecimientos

Este proyecto ha sido financiado por el vicerrectorado de Investigación, Desarrollo e Innovación de la Universidad Politécnica de Valencia mediante el proyecto "Comunicaciones Para La Distribución Del Procesamiento De Gráficos En La Tele Operación De Sistemas Autónomos".

Referencias

[1] Arkin, R., (1998). *Behavior-Based Robotics*. MIT Press.

[2] Kopetz, H., (1997) *Real-Time Systems : Design Principles for Distributed Embedded Applications*, pp 29-44.

[3] Pérez, P., Posadas, J.L., Simó, J.E., Benet, G., Blanes, F., (2002). *A Software Framework for Mobile Robot Sensor Fusion and Teleoperation*. Proceedings of the 15th IFAC world Congress. Barcelona (Spain).

[4] Web: SICK Sensor Intelligence (Julio 2004). http://www.sick.com