An Intelligent Sensor Architecture for Mobile Robots

P. Pérez, J.L. Posadas, G. Benet, F. Blanes, J. E. Simó

Computer Engineering Department
Real-time Systems and Robotics Group
Technical University of Valencia.
Apartado 22012. 46080 Valencia (SPAIN)
{pperez, jposadas, gbenet, pblanes, jsimo}@disca.upv.es

Abstract

This paper highlights the characteristics of software and hardware architecture for intelligent sensor and actuators in a distributed system as a mobile robot. The work focuses on reactive architectures and the method to interchange sensor and control data among distributed nodes. The work proposes a hybrid communication protocol between a pure TTP (Time Triggered Protocol) and ETP (Event Triggered Protocol) and reactive architecture for dual devices (sensor/actuator). Finally, a case study and experimental results with message latencies are presented.

1. Introduction

Autonomous mobile robots can be used in a lot of situations, as a museum guide or dangerous work in a nuclear plant with/without human assistance. There are numerous conditions around of robot that make difficult to apply an offline planning as semi-structured environments with continuous changes. To overcome this problem, mobile robots have a set of sensors that allows it to get small pictures of environments. Each single sensor is only capable of viewing one part of the environment through a narrow window of time and space. Besides, sensors measuring a particular property differ in their measuring methods, resolution, measuring range and conversion times. To overcome these limitations of single sensors, methods for the combination of data from different sensors via sensor fusion algorithms have been developed. There are two approaches to organize sensors: a centralized and a distributed paradigm. Traditionally, the first one’s main advantage is simple integration when the number of devices is small. With a single controller, the synchronization among data collected by sensors it is easy to achieve. On the other hand, if more devices are added to this system, the system gets more complexity, dificulting its connection and maintenance. In these cases is more adequate to use a distributed paradigm (divide and conquer). The main problem of this distributed paradigm is the synchronization, (functional and temporal).

Which controller must be charged with the task of ordering a particular device to make its job?, how and when should be distributed the data?. The first question depends of transducer type but the second one transforms the network design in a critical task. In a distributed paradigm, the data semantic of messages must be elected between three main alternatives: as a stream of signal samples, using meta-languages as XML; or using homogeneous data structures that allows an easy application of data fusion algorithms. A good election to implant on a hybrid[1][2] robot architecture (deliberative layer on a reactive backbone) is to use the last approach: an homogeneous data structure to represent a local map consisting of a bundle of characteristics vectors (proximity, lighting, noisy, and so on) [3][4]. This is adequate for a logical connection among sensors and actuators on reactive layer. Also, it is an useful method for sending environment information from reactive to deliberative layers.

Sensor data in a distributed system must be synchronized temporal and spatial in the whole system network. To achieve the synchronization in the present work, we use the temporal firewall concept [5]. It leads to two dif-
ferentiated tasks: the first one is to collect, to process and to fuse data from sensors in some cases; and to get data and to apply control laws to actuators in other cases. The second task is to get data in a synchronized way and to put them into a distributed and synchronized blackboard. This methodology follows the natural trend in intelligent sensors towards delegating more processing capabilities into themselves, as more powerful hardware is available at reasonable cost (size and price).

In this paper, the embedded structure of the intelligent sensor modules of a mobile robot inside its distributed real-time architecture is described. These embedded modules have been designed using the state-of-the-art technologies in embedded systems. Systems-on-a-Chip (SoC), Time-Triggered CAN (TTCAN), and embedded Real-Time Linux, are among the main technologic challenges involved in these intelligent sensors.

1.1. Previous Work

The present work is currently being developed as part of a research project named YAIR[3] project. This one falls within the context of advanced manufacturing systems in industrial processes, more specifically in the framework of intelligent systems to guide mobile robots in dynamic environments, which are usually non-fully specified and running under real-time restrictions. The goal of this project is to define and implement a multi-layer architecture that allows the development of several autonomous mobile agent prototypes with sufficient intelligence, and to develop a prototype of this kind of robot. The prototype must be able to be remotely planned and react to unexpected events. This kind of system requires low and medium perceptual capabilities in order to extract, understand, and manipulate spatial-temporal information about the environment and for the cooperation of those capabilities with internal plans of actions. The hybrid paradigm used in this robot has been widely used to represent mobile robotic systems. In such models, complex mobile robotic applications are defined by combining deliberative goal-oriented planning with reactive sensor driven operations. Agent programs should decompose complex actions into independent behaviours, which tightly couple sensing and acting. The robot first plans how to best decompose a task into subtasks (mission planning) and then, what are suitable behaviours to be accomplished in each subtask. Sensor data gets routed to each behaviour that needs it, but is also available to the planner for the construction of a task-oriented global world model.

Behaviour is defined as a mapping between sensor inputs and a pattern of motor actions, which are used to achieve a task, and this is the fundamental component of intelligence in most robot systems. A behavioural schema is composed of at least one motor schema and at least one perceptual schema, plus local, behaviour specific knowledge about how to coordinate multiple component schemas.

This separation of perception and action leads to the composition of behaviours by the combination of motor and perceptual schemas and suggests a method of implementation.

Behaviours are inherently parallel and distributed, as well as deliberative activities. Consequently, design requires the combination of reactive timing constrained activities with deliberative time consuming components. The temporal requirements and temporal performance of deliberative activities are different from the reactive activities. Furthermore, the behaviours temporal requirements are variable and dependent on the environment and/or application parameters. Reactive tasks have to cope with hard deadlines, while deliberative activities have soft deadlines. When a deliberative task misses a deadline, the system is not exposed to a risk of catastrophic malfunction, because the reactive set of tasks drive the system under control.

The implementation of a architecture, suitable for hybrid systems, has to support the distributed nature of data sensing and behaviours. The communication system will be the basis for constructing the motor and perceptual schemas and must be extensible to the deliberative level of computing. The difference of temporal requirements between reactive and deliberative tasks must be reflected in the communications system design. High real-time performance of communication infrastructure is normally related to predictable computing and a strict frame of communication semantic, well suited to behavioural activities. Other general purpose, non-real-time communication infrastructures have high semantic flexibility, which is necessary for deliberative computing.

YAIR has a communications system with two run-time environments: the first one is based on the CAN[6] bus, a
fieldbus that enables real-time features; the second one is supported by a distributed blackboard system (SC)[4][5].

This paper focuses on the reactive communication level and organization used by sensors and actuators on YAIR robot. The paper is divided into six sections. This section has described the scenario of architecture for intelligent sensors applied to mobile robots. In section II, a proposal of intelligent sensor architecture is described, and the framework to apply it. In the section III a complex sensor of the robot is described as a case study. The section IV presents a few experimental results. Finally, some conclusions about these applications and results are described in section V, and references are listed at the end of the paper.

2. General Description.

The physical architecture of the robot YAIR is depicted in Fig. 1. The use of multiple sensors calls for a meaningful way to combine the information provided by the individual sensors, in a process referred to as sensor integration. Four kind of distance sensors: ultrasonic, infrared, laser and bumper, give the robot the ability to cover a wide range of distances to detect obstacles, build maps (ultrasonic and infrared), avoid imminent collisions (infrared), make close approximations to objects (infrared and bumpers), or detect collisions. It has also an intelligent motion controller module with several odometric sources connected through CAN bus, incremental encoders attached to motors, encoders on casters and an electronic compass.

The more powerful microcontroller the more “intelligent” are the sensory modules. These intelligent sensors or smart transducers (IS/ST) comprise the integration of one or more sensor/actuator elements with a microcontroller and a communication network[7]. They have diagnostic and management services, real-time communication, auto-calibration of sensors, signal conditioning and conversion to standard units or homogeneous structures and so on. Finally, they allow considerable cost reduction in installation and maintenance.

The next section describes this architecture by means of a case study.

3 A case study: an ultrasonic rotary sensor module.

The ultrasonic module of the robot YAIR is a hybrid module. It has piezoelectric transducers that are mounted on a rotary head, resulting in a dual device with sensors and actuators. This characteristic is overcome by the reactive architecture of pattern behaviour described in previous sections. In a distributed system as our robot, several processes could try to access the same resource simultaneously. To solve multiple access problems, there are elements as pattern behaviour processes that carry out a kind of shorted queries to sensor controller. The controller combines a weighted list of these queries to obtain a single operation. The figure 3 depicts a graphical example of this procedure. In this case there are two behaviour that try to obtain information about environment, the #1 behaviour wants a scan of near environment to make a local map for fusing with global geometric maps and the #2 behaviour allows to robot knows object proximity in direction of the actual movements to speed up the robot works.

Fig. 1: YAIR physical architecture

Fig. 2. Graphical example of behaviour fusion
Each circle contains a numerical value that stands for the priority to make a measure on a particular direction, (this figure supposes only eight angular positions of the ultrasonic rotary sensor). Every behaviour assigns individual weights to these directions independently. Moreover, every behaviour has an execution priority: the motivation coefficient. In this case, when the robot moves quickly, it is more important to obtain information about frontal direction.

After behaviour execution, the sensor’s controller carries out a weighted addition and it obtains the real priority to get a measure for each direction. In the example of the figure 3, the final direction to move the sensor’s head to measure is the circle with the maximum value, (0.49). After the measuring process, the motivation coefficient decreases in the second behaviour because the robot knows the obstacles existence. This one will increase again when the data in this direction have poor accuracy, e.g. by time progression. The behaviour tasks put into the distributed blackboard the computed values, and these are collected by the rest of components in the robot connected through the blackboard by means of CAN bus. It is possible to obtain the presence of an obstacle, their distance and shape in some cases, or to put these samples directly into the blackboard to build a class of geometric grid map.

The link among components in robot is the blackboard and the producer and consumer process can be located in the same node or in another remote node. The position of these processes is static during robot execution, except to deliberative agents in JAV A that are capable to change of module, putting itself like data into distributed blackboard to move among nodes. The data structures are identified using the CAN physical identifier and each node has a small piece of this blackboard. Whereas a node produces information, the communication controller gets this data and sends it through bus maintaining the copies of distributes variables up-to-date.

There are two main communication paradigms for the design of distributed real-time communication systems. One of them is the event triggered communication where the temporal control signals are derived primarily from non-time events occurring inside and outside the computer system. The second one, the temporal control signals are solely derived from the progression of time. The CAN and TTP[5] are two examples of these paradigms, respectively. CAN is a field-bus with a message-oriented addressing schema, that is suitable for control applications. A node obtains sensorial information and then makes a message to notify this event to all nodes on network. Each network node filters the message. If it is not relevant for it, the node does not save the message.

Above CAN bus, there are several application layer protocols as DeviceNet, CAL&CANOpen, and so on, these protocols are extensions of basic addressing schema, callback’s management, coherent distribution of identifiers, and network status control and management, but it keeps a event triggered communication essential schema with the same problem: the theoretical jitter variability from block time to maximum latencies calculated in [8] difficulties the regulator design.

To reduce the jitter variability, a proposal for combining EDF in CAN is described in [9]. However, when there are resources and tasks dependencies as is the case of a mobile robot system with distributed intelligent sensor, the global method to assign priority to tasks on multi-thread nodes communicated through a prioritized transmission medium as CAN, it is assumed a lot of pessimistic analysis decisions to simplify the formal treatment [10].

To solve these two problems, jitter variability and tasks dependencies some authors propose to use a TTP paradigm. There are several approximations to this schema in CAN, a pure TTP paradigm proposed by CiA (CAN in Automation), Hong[11] presented a bandwidth allocation scheme applicable to CAN. Mock in [7], proposed an hybrid scheme. This shared channel method can be applied to CAN networks: it consists of splitting the bus access time in temporal slots between synchronization messages as in TTP protocol, but there are slots without
owner. In this case, whatever node on network can use these free shared slots to send a message. In shared slots, the node uses underground CAN medium access techniques to solve the contention on physical medium.

In this protocol, the network medium is shared by real-time and non-real-time data. The real-time data are subdivided into control and event data. Control data are generated from reactive loops which consist of: behaviour processes, controllers and motivation processes. Event data is generated sporadically within a bound of time interval, as alarms. Non real-time data comprise program and data files, and database management information as well. In this protocol, the control data have the greatest priority.

To implement this protocol in YAIR robot, there is a master node that sends a periodic synchronization message (SYNC), due to the fact of that CAN bus addressing schema is message-identification oriented, all messages reach to all nodes on network (messages have not target address). Then the SYNC boots a state machine that reads the transmission messages tables cyclically. Thanks to a dedicated communication processor, the starting times of time slots are defined within a window of few microseconds. In this schema, high priority messages have a time of transmission relative to this SYNC message. Therefore, its transmission is guaranteed in the designing stage. In contrast, low priority messages (as data from slow or complex sensors in the system) have not fixed latencies, but it is still possible to calculate their maximum latencies. The analysis of these ones is comparable with the schedulability test by fixed priorities [8] for messages lower that 8 bytes; and with the more general analysis described in [4] for transmission of fragmented messages with arbitrary length. The main difference is that the shared channel protocol leaves only a bit piece of throughput for Non-real-time messages.

In YAIR robot the SYNC message is transmitted every 10 ms, and each slot has 512 μs (three complete CAN messages), to support two simultaneous transmissions and a single error retransmission. Each node has a table with reserved slots assignation, specified in microseconds relating to SYNC reception. There are entries to shared slots too. When a node wants to transmit non periodic messages, it uses these slots sequentially, verifying that their messages are transmitted.

In the case of ultrasonic module, the sensor gives to the rest of system two different data types: the first is a bundle structure composed by 16 angular positions, (each position contains a normalized value corresponding to the measured distance to objects in the environment). The other data type is a complete local map composed by 16 vectors of 256 samples each, that covers the surrounding objects within 4.5 metres radius around the robot. This matrix is sufficient to reconstruction of nearby environment.

Due to different requirement of resources of these structures, the node sends data using two types of messages. The first structure is transmitted as a periodic message in reserved slots. On the other hand the transmission of a complete map needs 4 Kb. It is not viable the transmission as a periodic message in reserved slots, and for this reason the complete map is transmitted in blocks using some shared slots.

The second problem solved with this approach is the problem of the global scheduling. This approach assumes the principle of “divide and conquer”. It makes a clear distinction between acquisition and processing procedures and communication procedures. The mechanism of temporal firewall isolates each node from the point of view of a global scheduling analysis; it is the property of temporal compositability.

The responsibility of data consistency falls on the communication process in network microcontroller. Messages have time stamp and this process checks the temporal validity of each message. This model of communication allows to study the whole system as composed of isolated nodes, and the bus system and the communication controllers of each node as a single logic entity.

4 Experimental Evaluation.

A set of experiments has been performed using the above described system in real scenarios with the YAIR robot. Due to the hybrid structure of the system (hard real-time communications at reactive level and soft real-time at the deliberative level) in whole work different analysis has been applied for each case. The present paper is focused on the first one, that is related with the CAN bus with shared channel protocol to ensure that it could manage the involved workload with robot sensing and control tasks in the nodes of the distributed system. The test was designed using four nodes sending messages to
evaluate the workload of the low-level communication system, the reactive response. The configuration of communication channel in this test is: SYNC cycle of 10 ms, slot time of 512 us, with a total of 20 slots by basic cycle (The bitrate is programmed to 1MBit/s). The length of real-time messages transmitted is proportional to number of reserved slots assigned, in contrast with non-real-time messages that transmit in fragments distributed in a lot of shared slots.

<table>
<thead>
<tr>
<th>Table 1. Set messages in test workload</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Node</strong></td>
</tr>
<tr>
<td>Infrared</td>
</tr>
<tr>
<td>Ultrasonic</td>
</tr>
<tr>
<td>Main node</td>
</tr>
<tr>
<td>Infrared</td>
</tr>
<tr>
<td>Motors</td>
</tr>
<tr>
<td>Caster</td>
</tr>
</tbody>
</table>


table(*) by basic cycle.

<table>
<thead>
<tr>
<th>Table 2. Experimental results on shared channel protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Node</strong></td>
</tr>
<tr>
<td>Infrared</td>
</tr>
<tr>
<td>Ultrasonic</td>
</tr>
<tr>
<td>Main node</td>
</tr>
<tr>
<td>Infrared</td>
</tr>
<tr>
<td>Motors</td>
</tr>
<tr>
<td>Caster</td>
</tr>
</tbody>
</table>

\(\text{(*) not important due to reserved slot, highest priority.}\)

5 Summary and Conclusions

In this paper, a reactive architecture and a communication protocol have been presented, aimed to be used in intelligent sensors and actuators.

The paper proposes to take the digital signal processing algorithms near to the producer nodes (sensory nodes), increasing the intelligence of sensors and actuators and applying the concept of temporal firewall to isolate the communication mechanisms from control loops. To support this architecture, the work focuses on two directions: the hybrid architecture of the sensor: behaviour, controllers, and motivation processes; and a communication protocol that allows the communication of data between these kind of components. The shared protocol used in this system allows the reduction of the communication jitter to zero, that it is more adequate to control systems than traditional scheme aimed to event-triggered of CAN bus. Besides, it provides a good response to event and non-real-time data.

The main characteristic of protocol is the simplicity, it differs of FTTTP on CAN in the schema of the distribution of shared slots, and it differs from TTCAN in the matrix, in this case there is a single basic cycle only. The use of shared slot is different also. If a node does not use a private slot in 50us it is transformed in a shared slot automatically.

This generic architecture for intelligent sensors has been applied to several sensors. In this case, the complexity of the modules is greater than in single sensor or single actuator modules, as both sensor and actuator parts must work in coordination. This module has been constructed and tested into the robot prototype, under real applications, with satisfactory results.

References
