

Fieldbuses in Real-Time Control of Autonomous Vehicles

Urbano Nunes[†], José Alberto Fonseca[‡], Luís Almeida[‡], Rui Araújo[†] and Rodrigo Maia[†]

ABSTRACT

In this article a survey on fieldbuses is conducted with a special emphasis in real-time control applications for autonomous vehicles. Actual tendencies as well as major characteristics of fieldbuses are described. Applications in high demand systems, either in terms of reliability as in terms of real-time restrictions, are analyzed. Autonomous vehicles are complex systems requiring real-time distributed embedded control composed by multiple acquisition, processing and actuation devices. The interconnection of the distributed intelligent devices is a key factor in the overall performance of the system. To better understanding the requirements in terms of connections in an autonomous vehicle, the main modules of a global navigation system are conceptually described in a systematic way. A special focus in CAN fieldbus is made in order to point out some potentialities, which make it very attractive in autonomous vehicles real-time applications.

KEYWORDS: Fieldbus, Mobile Robotics, Autonomous Vehicles, Autonomous Navigation

1. INTRODUCTION

According to Thomesse [Thomessse 97], the history of Fieldbuses started around 1980 with preliminary experiences and working groups leading to the main fieldbus standards used today. A common meaning for fieldbus is, again according to Thomesse, “a network for connecting field devices as sensors, actuators, and field controllers as PLCs, regulators, drive controllers, and so on”. So, the fieldbus is in fact a communication system, organised as a layered structure from the seven layers OSI model, targeted for specific applications. The first major field of application of fieldbuses has been the industrial automation domain. The main concern there was to simplify the huge amount of cabling required to automate different parts of the factories. In the early eighties different standards still used today emerged. This is the case, for example, of Profibus [Bender 93]

[†] Institute for Systems and Robotics (ISR) – University of Coimbra, 3030-290 Coimbra, Portugal

[‡] Department of Electronics and Telecommunications – University of Aveiro, 3810-193 Aveiro, Portugal

a clear leader in industrial automation in Europe, FIP, currently named WorldFIP [Leterrier 92] which dominates transportation systems such as trains and CAN [Bosch 91].

In what concerns CAN, the original protocol definition was mainly targeted for the use in the automotive industry. However, CAN has evolved in the direction of industrial automation due to the development of application layer protocols such as DeviceNet. This protocol made CAN one of the dominating communications technologies in these applications in the United States. It should also be noticed that this is not the only application protocol available for CAN. One can find also CANOpen, SDS, CanKingdom and others

Another direction in the evolution of fieldbuses in general and CAN in particular was the real-time distributed embedded systems domain. In fact, fieldbuses were no longer viewed as just a simplified interconnection subsystem but they began to be considered a subsystem in which one of the most important roles was to guarantee timeliness properties in the exchange of information between the other parts of the system. This view of the importance of the fieldbus in the overall distributed system led to intense research activity to find suitable solutions to achieve real-time behaviour and, also, to support the safety requirements of the target applications. The automotive industry plays here a clearly important role as a driving force for the research and development.

Besides being one of the first fieldbuses with adequate characteristics for these applications, CAN has also been one of the first fieldbuses whose real-time characteristics were investigated in the context of the automotive industry. A seminal work by Tindell and Burns in 1994 [Tindell 94] determined the worst case response time of a periodic message stream considering that the full set of higher priority message streams were also periodic and known before the system operation. This work triggered intensive research around the use of fieldbuses in real-time applications in general and, in particular, in the use of CAN.

In this paper we will anchor in CAN to discuss the issues of relying in a fieldbus for the control of safety critical complex systems such as vehicles. We will describe briefly the automotive domain which has been traditionally the more advanced application of this kind, and we will extrapolate for the case of mobile robots and autonomous vehicles.

2. FIELDBUSES IN DISTRIBUTED CONTROL SYSTEMS

Embedded systems are often associated with the control of a specific machine or device.

This is the case of the envisaged applications of mobile robots or autonomous vehicles as it is in automotive industries. As these are complex devices they must include several control loops. Each control loop implies the periodic sampling of sensor data as well as an also periodic actuation operation. The sampling rate depends on the dynamic characteristics of the controlled system. Thus, different control loops in a vehicle will have different sampling rates.

Three architectures can be used to perform these control operations. One can be the use of a separate control subsystem for each controlled system. This requires, firstly, a set of specific cabling to interconnect the sensor to the controller and this one to the actuator. However control loops are not isolated. Setpoints must be changed dynamically which means that, secondly, there must be also an interconnection to a higher level system element. This architecture has a high penalty in cabling length and a difficult coordination between subsystems.

The second architectural solution is the traditional centralized control system. A computational device with the adequate processing power executes the control algorithms of every controller. Connections to every sensor and actuator in the system are thus required. This architecture simplifies the coordination problem as it is easy to schedule the operations to achieve real-time performance using well known scheduling theory and techniques. However, it incurs also in a high penalty in cabling. Other disadvantages are the limited scalability and the costs of dependability. In fact, such a complex unit is expensive and thus the installed processing power will not be overestimated. This will limit scalability. In what concerns dependability, this architecture is not very suitable, first because the use of redundant units becomes costly, since the powerful central unit is typically expensive. Also, the handling of backups is not easy and the physical localisation of replicas is a problem, too. It is known that they must be separated to avoid simultaneous destruction in the sequence of a crash. This will add a huge amount of extra cabling and difficult assembly.

The third alternative is a distributed architecture using a common communication medium. In this approach, each control system can be divided in simpler units with different roles: controller nodes, sensor nodes, actuator nodes. The control loop functionalities may then be distributed between 3 or more different system nodes. Besides a significant reduction in cabling, due to the sharing of the communication medium, this distribution has several other advantages. One is the possibility to move the functional blocks to locations close to the subsystem with which

they interact. The sensors and actuators can be equipped with processing and communications capabilities, so that they can execute certain functions locally, e.g. configuration and calibration. Other advantage is the possibility to install anywhere nodes that don't interact with external devices, like the controller node. Also, one major advantage is the possibility to achieve a modular design of the system by dividing the complexity by simpler units in which some of the functionalities are encapsulated.

The use of a shared communication medium means that data can also be shared between nodes. As an example, an independent node, different from the controller, can perform systems identification using the data exchanged between the other nodes without additional traffic in the communication medium. This of course requires capability of the communication system to perform broadcast or at least multicast, which is an intrinsic feature in fieldbuses using a producer-distributor model. This model promotes the coherence of data between the different system nodes, which is also an important property for many distributed applications.

The dispersion of the complexity of the global system among much simpler nodes, when compared to the centralised approach, has also advantages from the dependability point of view. First of all, replication of nodes is cheaper because nodes are simpler and only critical nodes must be replicated. Secondly, replication may often be at the functional level instead of being at the physical level. For example, a recovery from a failure in a controller may be solved using available processing power of a second fully operational controller or other node with adequate capabilities. Even if the full processing power is not available, the system may perform a graceful degradation of its operation by running the control subsystems at lower rates to compensate for the necessity to accommodate the execution of processes from faulty nodes.

Finally, using just a single bus for the interconnection makes much simpler the replication of the communication physical layer.

2.1. Control and real-time requirements

The issue of using a single communication medium to interconnect the overall system has also a significant impact on the controller performance and on the possibility to achieve real-time operation. In Fig. 1 there is an example of two control loops that represent subsystems in a distributed system. There, one can see two controllers A and B, which will generate several

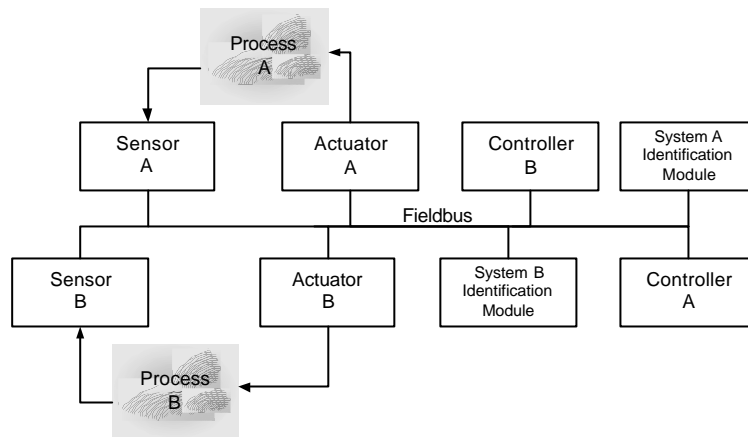


Fig. 1 – Example of two control loops conveyed by fieldbus

periodic data streams that must be conveyed by the fieldbus. In most cases, the periodicity of the data streams A and B are completely unrelated. Furthermore, the system will probably include other nodes, monitoring alarm situations that, when detected, require the transmission of alarm data. The need to transmit all these data using the shared fieldbus will originate conflicts that must be arbitrated by the fieldbus medium access control (MAC).

The sharing of the communication link implies then that the data may not be transmitted in the exact instants in which the application decided to do it. So, in these systems, there must be special care to guarantee that real-time data streams or a sporadic transmission of an alarm are transmitted within a maximum admissible delay, typically called deadline. Also, in periodic data streams, it becomes practically impossible, except when there is a very particular relationship between periods, to avoid variations in the instantaneous values of the period. This variation, known as jitter, can jeopardise the performance of control algorithms. In Fig. 2 this effect of network-induced interference between data streams leading to jitter is illustrated.

In what concerns the fulfilment of deadlines of real-time data, it becomes fundamental to determine the worst case response time of the fieldbus for each individual sporadic datum and for each data stream. This response time measures the interval between the transmission request and the actual transmission instant of the particular datum. To obtain this value it is necessary to obtain a previous knowledge of the data streams set that will use the fieldbus. The full knowledge may, however, be unnecessary if there is the possibility to prioritise critical data streams and if it is possible to define the maximum blocking time that can be imposed by the transmission of lower priority data.

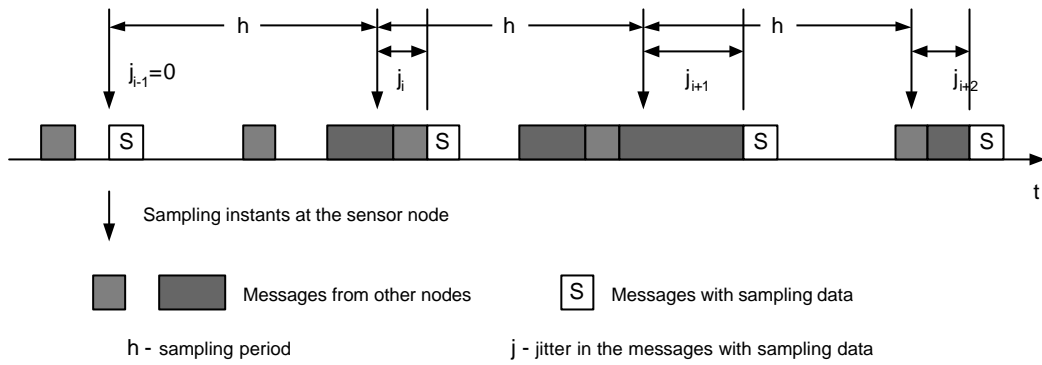


Fig. 2 – Network-induced jitter

The Medium Access Control (MAC) plays then a very important role in determining if a fieldbus is suitable for distributed real-time systems. It can even have a negative impact as it is the case, for example, of the CSMA-CD (Carrier Sense Multiple Access with Collision Detection) MAC used in Ethernet whose non-deterministic behaviour makes difficult its use in these applications. However, considering this example, modifications as the ones presented in [Pedreiras 02], [Jasperneit 01] and [Kweon 00] may in the future overcome this limitation.

On the contrary, the MAC of CAN, usually called CSMA-CA, where the CA means collision avoidance, shows adequate characteristics. For example in the previously referred work by Tindell and Burns [Tindell 94], the timeliness of the transmission through CAN of a set of data streams needed for the control of AGVs (Automatic Guided Vehicles) was verified using worst case response time analysis. This set, specified by the SAE – Society of Automotive Engineers, is often used as a benchmark for fieldbus based distributed control systems. Another study of the transmission of periodic data in CAN was done in [Navet 97] using data from PSA (Peugeot-Citroen group) vehicles.

In what concerns the problem of conveying periodic control data subject to network-induced interferences there has been several issues discussed. In [Hong 95] the problem of losing samples, called vacant sampling is analysed. In [Stothert 98], the effect of jitter in the sampling and in the actuation data is analysed for different controllers. As far as we know, in this last work were firstly presented some illustrative designations for this phenomenon: read-in and read-out jitter. This problem can be addressed from the control point of view by taking into consideration the jitter in the system's model [Mota 00] or from the communications point of view by minimising the network-induced interference with adequate de-phasing of the message streams

[Coutinho 00].

3. AUTONOMOUS VEHICLES

Autonomous vehicles are penetrating on the market and there are high expectations in that the applications of these machines spread all over the world with strong social and economical impacts, namely in manufacturing, oceans and space exploration, medicine, highway and urban transportation. Modern AGVs can navigate in their work area, both indoors and outdoors, without the need of routes defined by physical means. This modern AGVs are actually free-ranging autonomous vehicles that navigate based on dead-reckoning and some technique of absolute re-positioning provided at frequent intervals. Various re-positioning techniques have already been proven in practice such as grid navigation [Van Brussel 88], laser triangulation and GPS (Global Positioning System). Free-ranging AGVs are already operational in outdoors applications such as in driverless people transportation (the Frog Parkshuttle running in Schiphol airport and in Rivium, Holland) and in the transport of heavy steel slabs in steel plants [Makela 01]. This navigation technology used in electric vehicles, equipped with collision avoidance systems, offers a transportation solution that is safe and reliable, flexible, environmentally friendly and cost-effective. It is cost-effective because the operational costs are moderate and the investment cost on infrastructures is reduced since there is no need of physical guidance means.

In this section, a navigation architecture capable of providing an intelligent motion control of autonomous vehicles is conceptually described in a systematic way. The architecture, presented in Fig. 3, is structured in four levels: global motion planning level, local motion planning level, motion tracking level and motion control level.

3.1. Global Motion Planning Level

The global motion planning level is responsible for feeding the local motion planning level with a path to a predefined goal, which may be provided by a Human-Machine Interface (HMI) or a centralized task controller. For this purpose a Knowledge Database (KD) module is also considered, where information regarding the working environment, situation-based restrictions and driving rules are stored. A landmark based re-localization module is also necessary, and is considered at this level since its purpose is to determine the absolute (or global) localisation of the

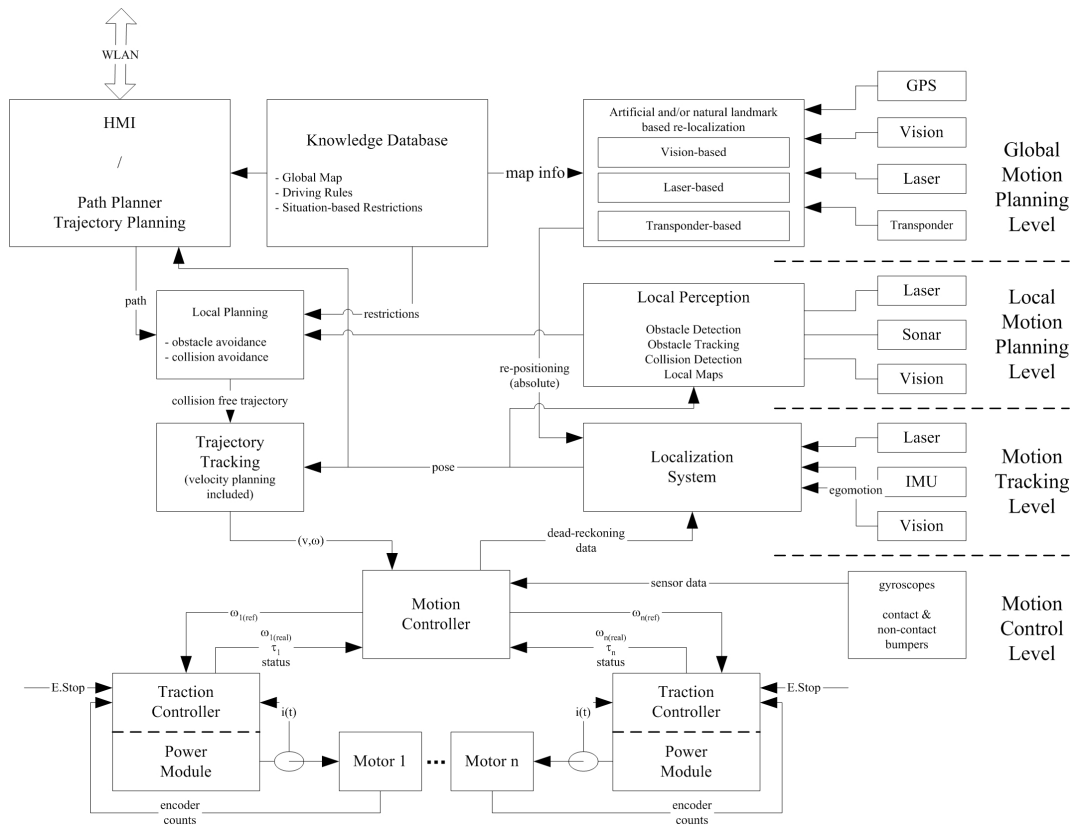


Fig. 3 – Levels and Modules of a Global Navigation System

vehicle.

We are not introducing in this study the global map-building problem. It is assumed that an environment map exists that contains the static structure of the indoor environment, or the relevant landmarks in case of an outdoor environment. However, the vehicle needs to acquire surrounding information to cope with two problems: 1) self-localization; 2) react appropriately to the dynamic environment.

3.1.1. Re-Localization

An absolute localization system requires the existence of a set of references in the environment. These references are marks or devices that are represented in the world model by their locations. The mobile system should be able to gather information about the relative location (distance and/or angles), between the robot and a set of references. This information, when integrated with information of the environment model, yields by geometric analysis a set of restrictions about the location of the robot from which a final estimate is extracted. For example, the trilateration (or triangulation [Betke 97]) consists on the robot's pose estimation, from distance

measures (or angles measures) between the robot and a set of absolute references. References on the environment can be active or passive. In the first case they are called *beacons* which can be devices that emit, or receive, signals such as grids embedded in the floor [Van Brussel 88] (using for instance optical, magnetic or transponder technologies) laser triangulation, satellites [GPS 99], and ultra-sound [Kleeman 92]. The vehicle carries on board the adequate receivers, or emitters, and its navigation system takes into account the geometric distribution of references to calculate the absolute localization.

Passive references, or landmarks, are objects or features, which the robot can perceive and recognize from the sensor data. There are two types of marks: artificial and natural. Artificial landmarks are objects or marks specifically designed to incorporate in the environment for the navigation purpose. Landmarks with specific reflexive proprieties, geometric features of various forms [Feng 92] or bar codes that can be identified by vision sensors or laser scanning, have been used. Natural landmarks are objects or features of the environment whose previous use, if any, did not include the navigation aid of the mobile system. The fundamental idea of beacons and artificial landmarks is to intervene on the environment, through the a-priori placement of references, in order to facilitate or solve the problem of establishing a world model relevant for simplifying and proceeding with the localization process.

Absolute localization methods usually require a mechanism to optimize the correspondence between a local map and a global map of the environment (for example see [Yamauchi 98], [Duckett 98]).

3.2. *Local Motion Planning Level*

The local motion planning level is responsible for providing the motion tracking level with a collision free trajectory in the reference path.

For a dynamic environment the robot may need to deviate from its path or at least it can be forced to change its velocity not following the prescribed value. A local navigation control layer is required to endow the autonomous robot with the capability to react appropriately to the dynamic environment, which may include obstacles moving nearby with unknown trajectories. The perception model at this level is in charge of maintaining a local environment model. This model, essentially composed from sensory-based data, should contain relevant and updated information

that enable the robot to navigate safely avoiding collisions.

The local planning module may have connections with the path planner, KD, local perception module and trajectory-tracking controller. When obstacles appear that can disturb the normal travel of the robot the control loop becomes closed at the local navigation level. The robot can exhibit one of two main classes of behaviours: collision avoidance or obstacle avoidance. In the first group we include the autonomous driving behaviours that contribute to avoid collision keeping the vehicle on its path. Otherwise, the robot may experience obstacle avoidance behaviour if it is allowed to navigate out of its prescribed path.

A collision avoidance module is a crucial technology, for instance, to equip intelligent vehicles running autonomously in outdoors environments following a planned path. In this case appropriate collision avoidance architecture could be based on a perception module in charge of multiple object tracking and object classification, and a decision module that mimics human driver behaviour in different driving situations, like free-flow, following leading vehicle, cut-in, braking and approaching. Presently research is carried out to produce perception modules having as inputs range and visual data, provided by radar, laser and stereovision. The data fusion of these multisensor modalities [Langheim 01] is crucial for extracting relevant and robust information from raw data.

Since the initial works on obstacle avoidance methods based on artificial potential fields [Khatib 86] many methods have been proposed. Potential field methods cannot be applied directly to nonholonomic robots. The robot is treated as a *free-flying object* moving under the influence of an artificial potential field, with obstacles asserting repulsive forces on the robot and the goal asserting an attractive force. The obstacles in the surroundings, as perceived by the robot, typically define the environment model for these methods. This means that the model has information of a snapshot of the environment, as captured and processed from sensor data, composed by the geometry and position of obstacles. No information regarding the obstacles dynamic state is derived. Most obstacle avoidance methods make a similar use of the local information, i.e. object tracking is not performed. Examples of these methods include the Vector Field Histogram (VFH) [Borenstein 91], the Dynamic Window approach (DW) [Fox 97], the Elastic Band [Khatib 96], the Nearness Diagram navigation (ND) [Minguez 00], and other strategies implemented by Fuzzy-Systems [Pires 02].

In the VFH, the motion direction and velocity commands are computed from a histogram description derived from an occupancy grid representation of the local environment. Empirical threshold is applied in the transformation and the VFH does not take into account the width of the robot. This approach has been improved giving rise first to the VFH+ [Ulrich 98] and more recently to the VFH* [Ulrich 00]. This last method incorporates a searching process for the proper action on a spatial tree spanned by a set of candidate directions. In the DW approach is assumed that the robot moves in circular paths, and the search of motion commands is performed in the velocity space (longitudinal and rotational velocities). Kinematics and dynamic constraints are taken into account but the solution (velocity commands) depends on a set of three weighting coefficients heuristically assigned. These coefficients are used to balance the weights of the goal heading, the forward robot velocity and the obstacle clearance. Extensions to the DW approach have been proposed [Brock 99, Arras 02]. In the Arras et al approach a combination of a model stage, which accounts for robot shape and dynamics, with a planning stage is proposed. Paths from the start to the goal position are generated in the planning stage. DW-based approaches rely upon a model of the local obstacles from which the distance to collision along with circular arcs are calculated giving rise to the dynamic window velocity space. The ND follows a situated-activity methodology by using five laws of motion adapted to five situations that describe the relative state of the robot, obstacle distribution, and goal location. In the perception step, the current situation is identified by looking for information regarding the environment, like regions of free space.

A different framework is proposed by the Velocity Obstacle approach (VO) [Fiorini 98], in which potential collisions and the collision-free path of a robot are determined, based on information of the obstacles positions and velocities.

We are pursuing a local navigation architecture embedding a set of behaviours, some of them inspired by human behaviour, and that, in addition to the use of local snapshot maps and obstacle tracking information, also take under consideration a set of driving rules and situation-based restrictions. This set of driving rules and restrictions, if applied to all vehicles sharing the same environment, make the vehicle navigation safer and more efficient, due to the increase of the behaviour predictability of the other vehicles sharing the same space. However the level of difficulty is higher, than usual, since, is mandatory that the robot has the real-time perception capacity of detect, track and classify the objects in its surroundings.

3.3. *Motion Tracking Level*

The motion tracking level determines the velocity commands, to be sent to the motion controller, based on the error between the reference trajectory provided by the upper levels and the real trajectory.

3.3.1. Localization

Navigation systems can be categorized as positioning or dead-reckoning systems. A positioning system determines the vehicle's pose regardless of the traveled path. Dead-reckoning systems keep calculating its pose with respect to a previous determined pose. Concerning their flexibility, navigation systems can also be categorized as rigid or free-range. A free ranging system is not dependent of a rigid track, allowing the vehicle to follow flexible paths. Positioning systems [Lobo 98] provides absolute localization with bounded error but without the required precision, and worst, without the desired frequency compatible with its use in a closed-loop real-time control system. On the other hand the dead-reckoning has accurate short-term measurements but its precision degrades with time due to integration errors. Because absolute positioning and dead-reckoning complement each other, it is common practice to fuse the data provided by both systems in applications calling for continuous high accuracy and reliability. An example is the fusion of dead-reckoning data produced by IMU (Inertial Measurement Unit) with DC-GPS (Differential Carrier GPS) [Farrell 00].

Other methodologies make use of range sensors data [Roumeliotis 02, Lu 97], like laser or cameras, which can also produce estimates about vehicle displacement, by matching 2D range scans.

The time-to-time re-localisation data provided by the upper level that performs global localization through landmark detection is used to limit the growth of the integration error.

3.3.2. Path tracking

A planned path is provided by the global or local planning levels as a list of reference coordinate points and reference velocities. The input variables for the path-tracking are the desired velocity, present vehicle's pose and the next desired vehicle's pose. The trajectory tracking

includes a velocity planning module that calculates the desired velocity as function of the reference velocity defined by upper levels and some constraints such as the curvature radius of the trajectory (this information can be provided by the trajectory planner). From the pose error the linear and angular velocities are computed and provided to the motion control level.

3.4. Motion Control Level

The motion control level is responsible for the robust velocity servo-control.

The motion control architecture here presented is based upon two different modules comprising two different control laws. At a lowest level a traction control module is used to control each wheel velocity. These modules use the encoder information and the torque developed by the motors to perform a feedback control that drives the wheels to their reference velocities. The torque information is important at this level to reduce the wheel slippage.

The upper level (the motion controller in Fig. 3) receives the desired velocity vector from the trajectory tracking module and sends the reference velocities to each traction module. For this purpose it uses inverse kinematics of the vehicle to calculate the reference velocities of both wheels. In complement to this classical approach dynamic models can be here applied in order to improve control robustness to wheel slippage, bumps, and slopes. With the proper dynamic models and information collected by the traction modules about the torque developed by each motor, reliable estimates of the vehicle slippage and stability can be achieved and used to improve the control.

At this level it is also required to perform the dead-reckoning calculations. These calculations often employ distinct types of proprioceptive sensors, e.g. odometers like optical encoder on wheels and gyroscopes [Prassler 01]. By applying appropriate integration of measured quantities, the vehicle displacement can be estimated. Nevertheless the values achieved by these techniques are noise contaminated and the estimates are subject to accumulate drift errors [Fuke 96, Barshan 95].

4. USING FIELDBUSES IN THE CONTROL OF VEHICLES

The control architecture described in the previous section has been partially implemented

by many researchers recurring to a set of heterogeneous subsystems targeted to a specific role. Several examples of these subsystems compose, for instance, the autonomous wheelchair prototypes described in [Tzafestas 01, Pires 02]. These subsystems process distinct critical levels of operations, involving many complex tasks, such as multisensor data acquisition and synchronisation, which have to exchange information among them. For this purpose, it is usual to install heterogeneous communication interfaces such as serial links (RS232), Ethernet, and fieldbuses [Tzafestas 01, Pires 02]. The interaction between the different subsystems becomes then cumbersome to implement, requiring gateways that make it difficult to achieve timeliness, when the data is exchanged between different communication systems. The target applications, autonomous vehicles and specific mobile robots, will then benefit from using a distributed architecture based on a single fieldbus, such as the one described in section 2. In fact, there has been recent examples of using fieldbuses in similar applications [Lisowski 97, Gil 97, Wahl 00, Langheim 01].

Often, the use of a fieldbus is just targeted to simplify the interconnection of subsystems or to promote modularity by encapsulating every function in simple dedicated nodes. However, an integrated view of the distributed embedded control system is required to promote its general dissemination. In spite of the many currently available fieldbus solutions, future dissemination will depend on cost figures, i.e. wide acceptance will require fieldbus technology to be cheap. In this sense, the automotive industry, involving large-scale productions, will definitely contribute to the availability of low-cost and efficient fieldbuses.

When one looks to the automotive industry nowadays, one can realize that most distributed control systems just enhance the performance of conventional mechanical systems (e.g. ABS, assisted steering). However, in the near future, car manufacturers want to remove all mechanical links between driver and actuators, relying on the distributed system only to perform car operation (X-by-wire). The distributed system must then be considered safety-critical and thus, capable of operating in the presence of faults. There are already some prototypes running such as the SKF's Filo X-by-wire system and a Mercedes vehicle presented in the last FlexRay [Belshner 01] seminar [FlexRay 02] in Munich.

In this scenario, the tendency of the fieldbus protocols used in the automotive industries, analyzed later in this section, indicates a clear move towards the time-triggered (TT) paradigm

[Kopetz 91] according to which all communication takes place at pre-determined instants in time. Using this paradigm requires that the transmission instants for every communication are known before hand. Thus, it is well suited to periodic communication for which it is easy to determine a priori all future transmission instants. Conversely, the transmission instants of aperiodic communication cannot be determined a priori and thus several compromises must be made to handle this type of communication in a time-triggered system.

Regarding the control architecture in Fig. 3, we may identify multiple periodic streams of information, namely: 1) motion commands from the local planning level to the trajectory tracking module and from this module to the motion controller; 2) dead-reckoning data from the motion controller to the localization module; 3) actual pose estimates from the localization module to the trajectory-tracking module; 4) laser data from sensor to different perception modules. Also there is a need for supporting aperiodic streams such as alarms, e.g. an unexpected eminent collision situation detected at the local perception module from laser or ultrasound sensors should reach urgently the motion controller.

The use of TT protocols also enables to set an adequate relative phasing between transmitted data streams by controlling the contention at the MAC level. This leads to eliminating, or at least bounding, the network-induced mutual interference that imposes jitter on the transmission of control variables.

Another feature of TT protocols that is particularly relevant to tackle complex applications such as the one considered here, is composability with respect to the temporal behaviour. This means that, upon integration in a larger system, individual subsystems maintain their temporal behaviour as established separately. For example, the motion control subsystem can be tested independently and included in the global system without suffering or causing interference from the remaining subsystems.

However, TT protocols tend to be less effective when dealing with information generated in a sporadic/aperiodic way and requiring fast response, e.g. alarms. That is why these protocols normally include special mechanisms to support event-triggered communication.

Another important issue is the flexibility of the distributed embedded system. This feature can be related to several functionalities such as the ability to support live insertion of nodes, the capacity to support on-line mode changes and download of new application software or

configuration data and the ability to support dynamic communication requirements. In what concerns this last point, event-triggered communication systems are typically considered flexible because they react promptly to communication requests that can be issued at any instant in time. In other words, they react to instantaneous (variable) communication requirements. On the other hand, time-triggered systems are not so flexible because communication takes place at pre-defined instants, only.

Examples of the importance of flexibility in terms of dynamic communication requirements can be found in situations triggered by an operator or by the system autonomously. For instance, the operator, through the HMI interface, can change operational parameters, e.g. control parameters, or even high level behaviours, e.g. path to be followed. Also, depending on the external environment, certain situations may require higher data rates from particular sensors than from others. For example, obstacle detection when there are no obstacles in a long range may require a lower scanning rate than when the vehicle approaches obstacles. Another example is tracking a given path. When the vehicle is moving with a steady low tracking error, the tracking sensors can be sampled at a lower rate. Upon error increase, a higher sampling rate can be re-established for improved control performance.

However, this flexibility can have a strong impact on dependability. One can easily understand that a change in communication requirements can possibly lead to a network overload and consequent timing failures. It is then mandatory to limit the extent of such changes so that the timely behaviour of the network is guaranteed.

Notice that the discussion on dependability is also relevant for the applications discussed in this paper since these systems, autonomous vehicles, will interact strongly with humans, which makes the safety problem a critical one. In this aspect, the modularity obtained with distributed systems facilitates the introduction of redundancy as it was discussed in section 2. Moreover, the time-triggered communication paradigm inherently supports detection of connectivity failures at the receiver because message arrival times are known. On the contrary, in event-triggered systems, e.g. CAN without any upper layer, a fail-silent failure in one node will not be immediately detected by the remaining nodes. To allow remote detection of failures, such systems must use a heartbeat, which is a timed mechanism using coarse synchronisation. An overview of safety critical issues in automotive networks can be found in a special issue of IEEE Micro [Koopman

02], including discussions about TTP [Maier 02], FTT-CAN [Ferreira 02] and FlexRay protocols.

Looking now to the situation in the automotive industry, three fieldbuses can be considered the main candidates to win the adoption war: CAN, TTP/C (Time-Triggered Protocol) and FlexRay [Belshner 01]. CAN, which has been used until now in cars, is not inherently a TT-based protocol. However, it is possible to add upper layers that allow obtaining TT behaviour. This is the case of TT-CAN (Time-Triggered Controller Area Network) [Führer 00], which resulted from a standardization effort under ISO, and FTT-CAN [Almeida 02], developed at the University of Aveiro, Portugal.

The widespread use of one these protocols will depend obviously on commercial interests but also on their efficiency in transmitting various types of data and on their dependability characteristics. In what concerns the first issue, event-triggered data, generated by alarms, and time-triggered data, associated to control loops, must be supported. The handling of dynamic communication requirements also seems to be an issue with growing importance. As it can be seen in the comparison made in [Almeida 02], it is difficult to determine which of these protocols is the best. In fact, when there is a better performance in one aspect in one of the protocols, e.g. TTP/C in what concerns time-triggered traffic and TT-CAN in event-triggered, there will be weaknesses in other aspects. Also, when there is sometimes some advantage in important aspects such as flexibility, as in FTT-CAN, it lacks the industrial backup.

In what concerns dependability issues the FlexRay consortium is claiming a lot of interesting features of their fieldbus, mainly derived from the redundancy of the physical interconnection and the specificity of bus guardians. However, this is already available or can also be obtained in other protocols, such as TTP. Deep discussion on these issues is thus expected, considering the interest in the implementation of X-by-wire solutions.

5. CONCLUSIONS

Fieldbuses will soon play the role of an important and critical component of the distributed embedded computer control system in charge of the autonomous vehicles operation. The current trend towards highly distributed architectures in the control system will continue to increase the responsibility of the fieldbus in determining important system properties, e.g. timeliness, flexibility, scalability, dependability and maintainability. Autonomous vehicles moving in

unknown environments require the system capacity to respond to sudden changes in the operating conditions. Besides other requirements, this calls for accommodating on-line variable data streams, i.e., for a flexible communication.

Automotive industry is moving to a widespread use of X-by wire systems and to rely in the distributed embedded system for the control of vehicles. The results of the technological development and of further research on the communications issue will certainly conduct to a fieldbus protocol that will become widely used in several applications including autonomous vehicles due to availability and cost reasons.

REFERENCES

- [Almeida 02] – L. Almeida, P. Pedreiras and J.A. Fonseca, “The FTT-CAN protocol: Why and How” To appear in *IEEE Transactions on Industrial Electronics*, (2002)
- [Arras 02] – K. Arras, J. Persson, N. Tomatis and R. Siegwart, “Real Time Obstacle Avoidance for Polygonal Robots with a Reduced Dynamic Window”, *Proc. IEEE Int. Conf. on Robotics & Automation*, 3050-3055 (USA, May, 2002)
- [Barshan 95] – B. Barshan and H. Durrant-White “Inertial Navigation Systems for Mobile Robots”, *IEEE Transactions. on Robotics & Automation*, **11(3)**, 328-342 (June 1995)
- [Belshner 01] – R. Belshner, et al., “FlexRay Requirements Specification - Version 1.9.7.”,(2001)
- [Bender 93] – Bender, K. (editor) *PROFIBUS – The Fieldbus for Industrial Automation*. (Prentice Hall, 1993)
- [Betke 97] – M. Betke and L. Gurvits “Mobile Robot Localization Using Landmarks” *IEEE Transactions on Robotics & Automation*, **13(2)**, 251-263, (April 1997)
- [Borenstein 91] – J. Borenstein and Y. Koren, “The Vector Field Histogram (VFH) – Fast Obstacle Avoidance for Mobile Robots” *IEEE Transactions on Robotics & Automation*, **7(3)**, (1991)
- [Bosch 91] – Bosch, “CAN specification version 2.0 - Tech. Report” *Bosch GmbH*, (Stuttgart, Germany, 1991)
- [Brock 99] – O. Brock and O. Khatib, “High-Speed Navigation Using the Global Dynamic Window Approach”, *Proc. IEEE Int. Conf. on Robotics & Automation*, 341-346, (USA, May, 1999)

- [Coutinho 00] – F. Coutinho, et al. “Jitter Minimisation with Genetic Algorithms”, *Proc. 3rd IEEE International Workshop on Factory Communication Systems (WFCS'2000)*, (Porto, Portugal, September 2000)
- [Duckett 98] – T. Duckett and U. Nehmzow “Mobile Robot Self Localization and Measurement of Performance in Middle-Scale Environments”, *Robotics and Autonomous Systems*, **24**, 57-69, (Elsevier, 1998)
- [Farrell 00] – J. Farrell, T. Givardis and M. Barth “Real-Time Differential Carrier GPS-Aided INS” *IEEE Transactions on Control Systems Technology*, **8(4)**, 709-721, (July 2000)
- [Feng 92] – L. Feng, Y. Fainman and Y. Koren “Estimation of the Absolute Position of Mobile Systems by an Optoelectronic Processor”, *IEEE Transactions on Systems, Man, and Cybernetics*, **22(5)**, 953-963, (1992)
- [Ferreira 02] – J. Ferreira, et al., “The FTT-CAN protocol: improving flexibility in safety-critical systems”, *IEEE Micro Special Issue on Critical Embedded Automotive Networks*, (April 2002)
- [Fiorini 98] – P. Fiorini and Z. Shiller “Motion Planning in Dynamic Enviroments using Velocity Obstacles”, *The International Journal of Robotics Research*, **17(7)**, 711-727 (1998)
- [FlexRay 02] – FlexRay Seminar, Munich, Germany, (April 2002)
- [Fox 97] - D. Fox, W. Burgard and S. Thrun “The Dynamic Window Approach to Collision Avoidance” *IEEE Transactions on Robotics & Automation*, **4(1)**(1997)
- [Führer 00] – T. Führer, T. et al “Time-triggered Communication on CAN (Time-triggered CAN-TTCAN)”, *Proc. of the 7th International CAN Conference (ICC'00)*, (2000)
- [Fuke 96] – Y. Fuke and E. Krotkov, “Dead Reckoning for Lunar Rover on Uneven Terrain”, *Proc. IEEE Int. Conf. On Robotics & Automation*, 411-416 (1996)
- [Gil 97] – J. Gil, et al., “A CAN Architecture for an Intelligent Mobile Robot”, *Proc. of IFAC Symposium on Intelligent Components for Control Applications (SICICA'97)*, (Annecy, France, 1997)
- [GPS 99] – GPS. “Special Issue on: GPS, The Global Positioning System”, *Proceedings of the IEEE*, **87(1)**, (January 1999)
- [Hong 95] – S. Hong, “Scheduling Algorithm of Data Sampling Times in the Integrated Communication and Control Systems”, *IEEE Transactions on Control Systems Technology*,

3(2), (June 1995)

- [Jasperneit 01] – J. Jasperneit and P. Neumann, “Switched Ethernet for Factory Communication”, *Proc. of 8th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA2001)*, (Antibes France, October 2001)
- [Khatib 86] – O. Khatib, “Real-Time Obstacle Avoidance for Manipulators and Mobile Robots”, *The International Journal of Robotics Research*, **5(1)**, (1986)
- [Khatib 96] – M. Khatib “Sensor-Based Motion Control for Mobile Robots” *PHD Thesis*, (LAAS-CNRS, December 1996)
- [Kleeman 92] – L. Kleeman “Optimal Estimation of Position and Heading for Mobile Robots Using Ultrasonic Beacons and Dead-reckoning”, *Proc. IEEE Int. Conf. on Robotics & Automation (ICRA'92)*, 2582-2587 (1992)
- [Koopman 02] – P. Koopman “Critical Embedded Automotive Networks”, *IEEE Micro Special issue on Critical Embedded Automotive Networks*, (April 2002)
- [Kopetz 91] – H. Kopetz, “Event-triggered versus Time-triggered Real-Time Systems”, *Lecture Notes in Computer Science*, **563**, 87-101, (1991)
- [Kweon 00] – S-K. Kweon and K. G. Shin, “Achieving Real-Time Communication over Ethernet with Adaptive Traffic Smoothing”, *Proc. of IEEE Real-Time Technology and Applications Symposium*, 90-100, (Washington DC, USA. June 2000)
- [Langheim 01] – J. Langheim, “CARSENSE – New environment sensing for advanced driver assistance systems”, *Proc. IEEE Int. Conf. On Intelligent Transportation Systems* (2001)
- [Leterrier 92] – P. Leterrier, “The FIP Protocol” *WorldFip Europe*, (France, 1992)
- [Lisowski 97] – L. Lisowski and G. Baille, “Specifications of a small electric vehicle: modular and distributed approach”, *Proc. IEEE Int. Conf. On Intelligent Robots and System (IROS'00)*, 919-924 (1997)
- [Lobo 98] – J. Lobo, L. Marques, J. Dias U. Nunes and A. de Almeida, “Sensors for Mobile Robot navigation”, in *Autonomous Robotic Systems*, A. de Almeida and O. Khatib (eds.), , 50-81, (Springer-Verlag 1998)
- [Lu 97] – F. Lu and E. Milos “Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans” *International Journal of Intelligent and Robotic Systems*, **18(3)**, 249-275, (March 97)

- [Maier 02] – R. Maier, et al., “The Time Triggered Architecture – A Consistent Computing Platform”, *IEEE Micro Special issue on Critical Embedded Automotive Networks*, (April 2002)
- [MAkela 01] - H. Makela and T. Von Numer, “Development of a navigation and control system for an autonomous outdoor vehicle in a steel plant”, *Control Engineering Practice*, **9**, 573-583, (2001)
- [Minguez 00] – J. Minguez and L. Montano “Nearness diagram navigation (ND): A new real time collision avoidance approach”, *Proc. IEEE Int. Conf. On Intelligent Robots and System (IROS'00)*, 2094-2100 (Japan, 2000)
- [Mota 00] – A. Mota, P. Fonseca and J.A. Fonseca, “Systems Modelling and Identification in CAN based Distributed Control Systems”, *Proc. 16th IFAC Workshop on Distributed Computer Control Systems (DCCS'2000)*, (Sydney, Australia, 2000)
- [Navet 97] – N. Navet and Y.-Q. Song, “Performance and Fault Tolerance of Real-Time Applications Distributed over CAN (Controller Area Network)” *CiA – CAN in Automation Research Award*, (1997)
- [Pedreiras 02] – P. Pedreiras, L. Almeida and P. Gai, “The FTT-Ethernet protocol: merging flexibility, timeliness and efficiency” *Proc. of the Euromicro Conference on Real-Time Systems*, (Vienna, June 2002)
- [Pires 02] - G. Pires and U. Nunes “A wheelchair steered through voice commands and assisted by a reactive fuzzy-logic controller”, *International Journal of Intelligent and Robotic Systems*, (in press)
- [Prassler 01] – E. Prassler, J. Scholz and P. Fiorini “A Robotic Wheelchair for Crowded Public Enviroments”, *IEEE Robotics & Automation Magazine*, **7(1)**, 38-45 (March 2001)
- [Roumeliotis 02] – S. Roumeliotis and J. Burdick “Stochastic Cloning: A Generalized Framework for Processing Relative State Measurements”, *Proc. IEEE Int. Conf. On Robotics & Automation*, 1788-1795 (USA, May, 2002)
- [Stohtert 98] – A. Stohtert and I. MacLeod, “Effect of Timing Jitter on Distributed Computer Control System Performance” *Proc. 15th IFAC Workshop on Distributed Computer Control Systems (DCCS'98)*, (1998)
- [Thomesse 97] – J.-P. Thomesse, “The Fieldbuses”, invited tutorial, *Proc. of IFAC Symposium on*

- Intelligent Components for Control Applications (SICICA'97), (Annecy, France, 1997)
- [Thomesse 99] – J.P. Thomesse, “Fieldbuses and Interoperability”, *Control Engineering Practice*, **7**, 81-94, (1999)
- [Thrun 98] – S. Thrun. “Bayesian Landmark Learning for Mobile Robot Localization”, *Machine Learning*, **33**, n°1, (1998)
- [Tindell 94] – K. Tindell and A. Burns, A., “Guaranteeing Message Latencies on Control Area Network (CAN)”, *Proc. of 1st International CAN Conference (ICC'94)*, (Mainz, Germany, 1994)
- [Tzafestas 01] – S. Tzafestas (Guest Editor), “Research on Autonomous Robotic Wheelchairs in Europe”, *Reinventing the Wheelchair, Special Issue of IEEE Robotics Automation Magazine*, vol. 7, n. 1, (March 2001)
- [Ulrich 01] – I. Ulrich and J. Borenstein “VFH*: Local Obstacle Avoidance with Look-Ahead Verification”, *Proc. IEEE Int. Conf. on Robotics & Automation (ICRA'00)*, (San Francisco, USA, 2000)
- [Ulrich 98] – I. Ulrich and J. Borenstein, “VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots” *Proc. IEEE Int. Conf. On Robotics & Automation* (1998)
- [Van Brussel 88] – H. Van Brussel, C. Helsdingen and K. Machiels, “FROG: free ranging on grid, new perspectives in automated transport”, *Proc. of the Sixth Int. Conf. on Automated Guided Vehicle Systems*, 223-232, (Brussels, October 1988)
- [Wahl 00] – M. Wahl and G. Dorée , "A CAN Application Layer for an Experimental Real-Time Obstacle Detection Study", *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems (ITSC'00)*, 276-281 (Dearborn, USA, 2000)
- [Yamauchi 98] – B. Yamauchi and R. Beer “Spatial Learning for Navigation in Dynamic Environments”, *IEEE Transactions on Systems, Man, and Cybernetics - Part B Cybernetics*, **26(3)**, 496-505, (1996)