Experimental Platform Car for Automatic Control Applications

S. Bouaziz, R. Reynaud, B. Larraudie, A. Elouardi
Institut d’Electronique Fondamentale - Université Paris Sud - 91405 Orsay cedex
samir.bouaziz@ief.u-psud.fr

Abstract—The PICAR platform is an electrical car including an embedded electronics system. The generic goal is to design an embedded multi sensor platform for automotive application such as collision avoidance. Therefore the system includes classical sensors like video camera and ultrasonic sensors, a PC bi-processors, a CAN network and dedicated software for signal and image processing, data fusion and decision system. This platform allows experimenting customized sensors and specific architectures dedicated to fusion systems and data processing. The target scenarios are collision avoidance system, automatic parking, and lateral control application.

Index Terms— Embedded architecture, CAN networked system, real-time and time-stamping, data fusion platform

I. INTRODUCTION

A great number of vehicles, equipped with sensors, are today available and allows making demonstrations of automatic or assisted car driving applications in scenarios such as automatic parking, lateral control on multi-lanes road [3], cruise control [4, 5]. Our vehicle is a platform whose the finality is an application for low speed control of a vehicle belonging to a fleet of vehicles in town [5].

The most important characteristic of this platform is to allow exhibiting the competencies of the different research axes of the team. The research domain deals particularly with implementing smart retina sensors, developing feature extraction mechanisms in the video images, data compression (image and sensors), real time considerations, with coherent time stamping in a multi-micro-controller system using a multiplexed bus (CAN). In order to promote the onboard hardware architecture and the corresponding principle of co-design, it must be possible to implement different data fusion or control based strategies, under the concept of active vision using model of selective attention.

This platform therefore implements as well low-level mechanism than high level. The philosophy is that the real power will come in the future from interactions between these different mechanisms and not only from high level mechanisms based on a cheaper and cheaper processing load.

First, this paper gives an overview of the PICAR platform and its different hardware parts, then it presents the real-time software implementation and finally we show how to implement the different controls.

II. THE HARDWARE PICAR PLATFORM

The PICAR platform is an electrical car including some embedded electronics systems and sensors. The goal is to experiment either customized sensors, suitable computer architectures dedicated to fusion systems and data processing, and new mechanisms of selective attention, in the case of the scenarios such as collision avoidance system, automatic parking, or application of lateral control on a multi-lane road.

Figures 1 and 2 show the physical smart car and the embedded system. The sensors get data from an internal state (proprioceptive) or an external state (exteroceptive). Internal sensors give us the proprioceptive information relative to the smart car, like speed, acceleration, car direction, brake status...etc. Exteroceptive sensor capabilities give external information around the car. They include cameras, telemeter, and ultrasonic sensors. All external data are used to detect obstacles. Data fusion processing is necessary to certify obstacle positions. Sensor data are multi-processed by PCs to make data fusion, to detect obstacles for tracking or avoiding.

The embedded system is based on a CAN Area Network. CAN network is used to exchange data between sensors and to centralize them in the embedded computer. CAN is a serial bus especially suited for networking smart devices as well as sensors and actuators, within a system or sub-system. This assumes modularity to add or changes some sensors and actuators according to the final application experiment. All sub-systems modules are accessible through the CAN and are subdivided in some parts: actuators, sensors and low level or hardware processing units.

2 http://www.omegas.co.uk/CAN/index.html.
A. Actuators

Three kinds of control are implemented:

- Steering control.
- Speed control.
- Light control, which includes night-light and indicators.

There are four motors; one for each wheel insures the longitudinal propulsion. A motor carries the brake. Steering Motor is used to control the direction (lateral actuator).

The steering and speed controls use a hardware implementation of the PID loop. Each actuator uses a PID integrated circuit (LM629) and is interfaced to a microcontroller chip. The timing constraints loops are upper then 1 ms. The microcontroller assumes data exchanges with the rest of the system by using the CAN bus.

Figure 3 gives a brief hardware organization of the control level of the car. The light actuators allow controlling all the lights of the car by the microcontroller.

There are two kinds of actions: implicit action and explicit action.

- The implicit action is dealt with by the microcontroller without external intervention. For example, the absence of order upstream during a certain time will induce the automatic stop of the vehicle, the brake will be activated and warning is signaled.

- Explicit order mean that the high level processing will send instructions (according to speed and the direction) to be respected.

A non-cohesive temporal instruction will cause the dead halt of the vehicle in automatic mode.

B. Sensors

Two kinds of sensors are onboard: proprioceptive and exteroceptive sensors.

Proprioceptive sensors indicate the inner situation of the smart car. Steering encoder is used to measure the wheel steering. Wheel encoder is used to measure the speed of the car and the longitudinal covered distance. This twice sensors are included in a looped control to adjust position and direction according to the command sent to the steering and speed control.

Exteroceptive sensors are used to capture the external environment around the platform. The objective is to detect fixed and moving objects like tree, others cars and lanes. To do this, we use different CCD camera sensors and also ultrasonic sensors for the nearby field around the car.

C. Hardware processing units

The computation load is distributed in two parts:

- a bi-processors PC embedded computer
- Some buried hardware circuits which support the low level computation loops, like PID controllers, data networked supervision or microcontroller nearby the sensors.

The whole hardware modules are interconnected by using a CAN bus. This architecture has the advantage to be extensible up to the maximum throughput available on this bus. In order to minimize the amount of data to be exchanged between each node, information broadcasting is preferred when it is possible so the whole amount is minimized. The figure 4 gives a good schema of the way the various modules are organized and interconnected.

The PC has the load to plan the trajectory to be followed according to the scenario. In a lane tracking scenario, the PC captures all information necessary to compute and produces a succession of instructions to carry out its the tracking. Mechanisms of data fusion, coming from multiple sensors, will allow to reinforce the
knowledge of the external world around the intelligent car and to reduce the uncertainty of the various sensors.

Fig. 4. Network organization

III. SYSTEM LEVEL

The main features of the onboard computer are the following: a bi processor P3 1 GHz, Ram 1 Go, DD 36 Go SCSI U3. Each processor is operated under a different operating system: Windows 2000 and RTX (a real time kernel). A first demonstration (test of the whole architecture by an open loop control command) is realized. Dedicated software generates a file of records. Each record corresponds to one command to be sent to the micro controller in order to control the steering wheel or the propulsion. It includes a field related to the target actuator, a field identifying the command itself, a field tagging the time at which this command must be applied to the corresponding actuator. This file is preprocessed to assume that the real time constraints can be reached by imposing a minimal delay between two consecutive commands. A simulator verifies then that the sampling of the command is sufficient to make compatible the real trajectory with the target trajectory.

The transmission of the record of the command file is realized by 2 process:
- A Win32 process (under Window 2000) has in charge to read the recorded data and to send them in a shared memory (First In First Out). It must also store the data coming from the micro controller in a file for further processing.
- A real-time task manages the sending of a record stored in the FIFO memory to the target actuator at the precise instant that is indicated in the corresponding field of the record. This task is activated each millisecond. It has also in charge to transmit the information captured by the different sensors, formatted in field, one of them being a time tag based on the microprocessor clock which is synchronized with the common PC clock. The exchange used the same mechanism of flow control with another shared memory.

The figure 5 shows clearly the two process and the corresponding exchange mechanisms. The two tasks operate on two distinct processors. The parallelism is therefore actual and full. This assumes a temporal coherence of the commands and a deterministic functioning of the system. Actuators programming and data dating coming from sensors is assumed to be precise with a timing uncertainty less than 1 millisecond.

Having in mind the prototyping aspect of the platform, a systematic procedure of data logging is implemented. This chain fulfills a data base, which allows replaying offline the different scenarios. All the recorded data, sent commands and sensors data are time stamped with a global time tag. Algorithms can so be checked without having to make a real experiment on the vehicle.

Nevertheless, it is possible to validate some control mechanisms or to analyze sensors data on the onboard computers. Due to the fact that all data are transmitted through the CAN bus, pre recorded data can be sent again on the CAN bus with respect to the time stamps. The different computers believe that data are coming from the sensors nodes. This way to do allows validating the processing algorithms by taking in account the constraints of the onboard computers specifically the processing power.

Fig. 6. Prototyping: offline and online schema
The figure 6 shows two concurrent schemas in the co-design process: first experiment with online exchanges; validation of some part of the experiment through the use of offline emulations.

IV. SCENARIO AND RESULTS

The architecture of the onboard system of the platform has been validated by controlling simple trajectories made by the platform.

The figure 7 illustrates an example of target trajectory that the PICAR platform must execute. The movement is split in two sampled command flows corresponding to the steering wheel and the propulsion.

The architecture of the onboard system of the platform has been validated by controlling simple trajectories made by the platform. The time interval between two commands must be bounded by 10 ms and 500 ms. When no command occurs during the watchdog time, the platform toggles in mode “implicit action” and becomes immobile. The control command loop of the PID chip is operated with a period of 700 µs for each degree of freedom. But the mechanical latencies of the different motors induce that no variation are perceptible below 10 ms.

The figures 8 and 9 show the results of a measurement of the steering angle and the longitudinal position of the platform. By combining the two data, the platform movement is close to the target trajectory.

Other scenarios have been implemented to refine the characterization of each sensor and each actuator. Then a global control command loop has been inserted to adapt dynamically the command by measuring the real position of the platform. The chosen exteroceptif sensors are two cameras and ultrasonic sensors. The mechanisms of multi-sensor data fusion have been validated and are operational.

The further step is the implementation on the platform of the target mechanisms of intentional vision. Scenarios are designed and we have obtained a financial help of our university to work with the teams of two others labs to evaluate the remote control of a vehicle fleet. Vehicles are used by people. When no people use it, a vehicle can be dispatched in some new location by using a remote human control in competition with some local mechanisms of intentional vision.

V. CONCLUSION

This paper deals with the design of the PICAR platform, particularly the hardware organization, the software constraints to take correctly the time in account, and the application experiment. The prototype is under construction for the algorithmic part. Our objective is to implement the low level mechanisms that assume the vehicle to be operational, but also the high-level control strategies through the concept of active vision using model of selective attention and the implementation of the competition between remote control and local supervision issued from active vision.

REFERENCES