Transportation Research Part C 39 (2014) 94–112

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Multilayer distributed intelligent control of an autonomous car

Humberto Martínez-Barberá *, David Herrero-Pérez

Department of Information and Communications Engineering, University of Murcia, 30100 Espinardo, Murcia, Spain

A R T I C L E I N F O

Article history:
Received 6 March 2012
Received in revised form 10 December 2013
Accepted 10 December 2013

Keywords:
Intelligent vehicles
Robotics development frameworks
Mobile robots

A B S T R A C T

This paper shows how the development of an intelligent vehicle application can benefit from using standard mobile robotics elements in general, and a development framework in particular. This framework, ThinkingCap-II, has been successfully used in other robotics applications. It consists of a series of modules and services that have been developed in Java and allows the distribution of these modules over a network. The framework facilitates reusing components and concepts from other developments, which permits increasing the performance of the intelligent vehicle development. This fact is especially useful for small research groups. A two car convoy application has been implemented using this architecture and the development of an autonomous vehicle. Both the ThinkingCap-II and the autonomous vehicle architectures are described in detail. Finally some experiments are presented. Simulated experiments are used to validate the convoy model, testing the activation of the different behaviors in the decision-making process. Real experiments show the actual working of the developed intelligent vehicle application.

1. Introduction

Vehicle automation for different automatic driving related tasks, e.g. Zohdy and Rakha (2012), and assisted driving, e.g. Jiménez and Naranjo (2011) and Chen et al. (2013), has increased during the last years. Nevertheless, in most cases these systems rely on architectures specifically designed for the sensorial system being used (artificial vision, radar, laser, etc.) or are task oriented, such as lane and car following, e.g. Ossen and Hoogendoorn (2011) and Zheng et al. (2013), obstacle avoidance, e.g. Jiménez and Naranjo (2011), and driver supervision, e.g. Di Stasi et al. (2012), Ahlstrom et al. (2013), and Martín de Diego et al. (2013), to name but a few.

There have been several projects and systems showing autonomous driving. Some early developments in autonomous vehicles are CMU NavLab in Thorpe et al. (1988, 1991), OSU vehicles in Ozguner et al. (1997) and Fu et al. (2008), California PATH in Ioannou (1998), ARGO in Broggi et al. (1999), SAVE in Coda et al. (1997), and OTTO in Franke et al. (1995). Among the recent autonomous vehicle developments, we can mention the vehicles participating in the Grand Cooperative Driving Challenge (GCDC), e.g. Martensson et al. (2012), and vehicles participating in DARPA Grand Challenge (DGC), e.g. Ozguner et al. (2007), and DARPA Urban Challenge (DUC), e.g. Campbell et al. (2010), sponsored by Defense Advanced Research Project Agency (DARPA) of US Department of Defense, to cite but a few. Most of them are not general or open. By contrast, in autonomous robotics there are several open frameworks, a compilation and review appears in Oreback and Christensen (2003) and Kramer and Scheutz (2007), that allow transfer and reusability.

The goal of this paper is to put these two together, by showing how to use a general framework (originally developed for autonomous robotics) in an intelligent vehicle application. This approach has been already tried by other groups, e.g. the work by Institut National de Recherche en Informatique et en Automatique (INRIA) in Laugier (1999). A hybrid architecture

* Corresponding author.
E-mail addresses: humberto@um.es (H. Martínez-Barberá), dherrero@um.es (D. Herrero-Pérez).

0968-090X/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.trc.2013.12.006

(hierarchical-reactive) for mobile robots is proposed and is applied to electric vehicle fleets in urban environments. An interesting component of the architecture is a meta-level of skills known as sensor-based maneuvers, which are general templates that encode high-level expert human knowledge and heuristics about how a specific motion task is to be performed. The control architecture has two main components: the mission monitor and the motion controller. Given a mission description, the mission monitor generates a parameterized motion plan, which is a set of generic Sensor-Based Maneuvers (SBM). These SBMs are selected from a SBM library. The goal of the motion controller is to execute the current SBM of the parameterized motion plan in a reactive way. The current SBM is instantiated according to the current execution context (parameters of the SBM are set using the *a priori* knowledge or sensed information available). Another example is the architecture for Intelligent Cruise Control (ICC) by Handmann et al. (1999), which is a flexible and modular architecture for ICC subdivided into three different processing steps: the object-related analysis of sensor data (sensor data is processed and interpreted), the behavior-based scene interpretation (relevant information for the behavior is processed), and the behavior planning (final element that has to evaluate which action should be taken to perform the current task).

The approach proposed in this paper consists of a general framework that allows implementing different cognitive or functional architectures, and, for instance, the two previous architectures can also be implemented. The framework includes a reference architecture that is used to build an intelligent vehicle application, which reuses many components from standard mobile robotics applications, for example electronics components and software modules from the industrial mobile robot shown in Martínez-Barberá and Herrero-Pérez (2010b,a).

The autonomous driving examples mentioned above show that most works take an application oriented or vertical approach, either in the hardware or in the software approach. In addition, many of those investigations have been supported by industrial car makers, whose politics and development agreements do not allow the dissemination of certain privileged information. For these reasons it is difficult to reuse standard mobile robotics techniques and systems for intelligent vehicle applications, especially so for small research groups. The aim of this paper is precisely to present a horizontal approach for intelligent vehicle applications targeted at testing proof-of-concept techniques and applications. The goal is to offer a framework for building an intelligent vehicle application, in which concepts and components from different robotics systems and other intelligent vehicle applications can be ported and reused. Portability and reusability do not usually concern large application oriented projects, but they are key issues for small research groups (or those not directly sponsored by car makers). One typical drawback of such a horizontal approach is performance. While this can be a problem in certain cases, most solutions can be reshaped to use standard architectures, components or techniques.

This paper shows the use of generic hardware and software architectures used in mobile robotics for an intelligent vehicle related application which is independent of the sensori-motor system used and which allows management of multiple vehicles. The application described in the paper borrows from the CHAUFFEUR project, Borodani (2000), the idea of convoying autonomous vehicles, in this case cars. The paper is organized as follows. Section 2 presents and describes some related work. Section 3 describes the characteristics and design criteria of the software architecture which is later used to control the autonomous car. Section 4 describes the hardware architecture of the developed autonomous car. Section 5 presents the instance of the software architecture for the two car convoy application. Some experiments in both simulated and real environments are presented in Section 6, and finally, the conclusions are discussed in Section 7.

2. Related work

Different research groups in universities and private companies of the automotive sector have been working in the development of intelligent vehicles by adding additional equipment to off-the-shelf vehicles. Among these, the different versions of the Carnegie Mellon University's NavLab automated vehicles are quite well known, and NavLab 5 has achieved world wide coverage with the *No Hands Across America* (NHAA) demonstration. This vehicle is equipped with a vision system, a laser range scanner for obstacle avoidance and a positioning system based on GPS and odometers. Its architecture, shown in Jochem et al. (1995), is highly oriented to the vision system.

The Ohio State University's OSU autonomous vehicles have also performed similar demonstrations in Ozguner et al. (1997). These vehicles are equipped with a vision system, a radar, a laser range scanner, and inertial units. Their architecture is a three layer hierarchical architecture, in which the lower layer is in charge of actuator control, the intermediate layer is in charge of the longitudinal and lateral control, and the higher layer is in charge of scenario level control. The interconnection of the sensors and actuators with the different control layers is mainly through RS-232 links.

Another good example of automated vehicles is the California PATH (Partners for Advanced Transit and Highways) project. This is a large research framework, which includes many subprojects that directly or indirectly make use of automated vehicles. Many of the new designs are based on the automated highway system control architecture (PATH AHS Architecture) described in Horowitz and Varaiya (2000). In this architecture, traffic is organized into platoons of closely packed vehicles. The design of this architecture consists of five hierarchical layers: Network, Link, Coordination, Regulation, and Physical. The first two layers are roadside control systems. As for the three layers inside the vehicle, the Coordination layer is a supervisory controller that determines which maneuvers to perform, manages inter-vehicle communications, and coordinates the movement of the vehicle with respect to neighboring cars. The Regulation layer is a continuous-time feedback-based controller that implements and executes the maneuvers. The Physical layer contains hardware, sensors, and low-level controllers.