ControlCIT – A Control Design and Implementation Toolbox for Automatic Vehicle Guidance

Jeroen Ploeg1, Zoltán Papp2, Rob. C. van de Pijpekamp1, Sándor E. Skolnik2, Edwin A.C. van den Eijnden1
1TNO Automotive, 2TNO Applied Physics

TNO Automotive, P.O. Box 756
5700 AT Helmond, The Netherlands
phone: +31 (0)492 566 536
fax: +31 (0)492 566 566
e-mail: ploeg@wt.tno.nl

The main objective of an Automatic Guided Vehicle (AGV) is to autonomously transport goods and/or people. This objective has to be realized under severe constraints imposed by safety and reliability requirements. In order to efficiently develop a control system for the targeted application area, a framework is established which facilitates the design and implementation of control systems for automatic vehicle guidance and for advanced driver assistance systems in general. This framework takes the shape of a toolbox named ControlCIT, comprising hardware and software tools for off-line control design, rapid control prototyping and (semi-)embedded implementation. This paper introduces the concepts behind ControlCIT, describes its components and presents an application to high performance AGV tracking control.

1. INTRODUCTION

The essential objective of an Automatic Guided Vehicle (AGV) is to transport goods and/or people. To this end it has to autonomously, i.e. without a driver, follow a desired but not necessarily a priori determined trajectory. To a certain extend, the same holds for ‘ordinary’ cars which have been equipped with intelligent on-board systems so as to take over certain parts of the driver task. Throughout this paper, the term Automatic Vehicle Guidance (AVG) is used, referring to intelligent road vehicles in general. Focusing on the realization of the main transport objective, the current research aims to provide a development framework which facilitates the design and implementation of control systems for AVG. This ultimately results in a toolbox, named ControlCIT, consisting of soft- and hardware.

Before going into detail, the next section first describes the objectives underlying the development of ControlCIT. Section 3 then focuses on the concepts behind ControlCIT and its main components. Section 4 presents an application to a four wheel steered, four wheel driven AGV. Section 5 contains the main conclusions.

2. CONTROL SYSTEM OBJECTIVES

We aim to achieve the transport objective in a high performance fashion, characterized by the possibility to operate under high speeds and high (lateral) accelerations or by a high level of driver/passenger comfort, so as to fully utilize the capabilities of the specific vehicle. In this area of operation, the dynamic vehicle characteristics such as yaw stability become very important. As a consequence, also the tire behavior plays an important role.

The realization of the required performance is subject to severe constraints with respect to safety and reliability, being two important aspects of what is generally called dependability [1]. Because of the very prominent role of both aspects in practical application of AVG, both the safety and the reliability constraints should rather be regarded as objectives of themselves, albeit with a strong interaction with the performance objective.

Structured according to the above mentioned objectives, a number of specifications can be defined, depending on the actual application. The resulting ‘objective oriented’ specification scheme is depicted in Fig. 1. The figure only indicates the main structure. In practice it is necessary to further detail these objectives by defining several sub-objectives and then formulate quantitative specifications and requirements for a specific application*.

*As a next step, the objective oriented specifications might be mapped to a set of specifications relating to physical parts of the controlled vehicle such as drive line, chassis and control system hard-/software. This results in what can be characterized as an object oriented set of specifications.
With the above specifications in mind, TNO Automotive, in close co-operation with the TNO Applied Physics institute, currently develops a toolbox for design and implementation of control systems for AVG, being characterized by the following properties and requirements:

i. Integral approach of performance, safety and reliability – In order to guarantee a certain performance, safe and reliable vehicle behavior, an integral approach is necessary, resulting in a control system reflecting these requirements in every aspect.

ii. Integral support of all development phases, from control system design through implementation to deployment – In order to significantly shorten the development time and related costs, it is desired to facilitate the entire development process, from modeling and control design on a simulation level up to and including the actual (embedded) implementation.

iii. Generic character enabling flexible application – For AVG in general, the generic character applies to the hard- and software of the toolbox in order to enable an easy adaptation to a specific application. Especially in case of AGV’s, the toolbox should be applicable to a number of vehicle platform types, ranging from mechanically coupled driven and steered wheels to independently driven and steered wheels.

iv. Application of mainstream hard- and software – Application of mainstream hard- and software guarantees a relatively low cost system as well as the possibility to benefit from ongoing world-wide technical developments.

3. SYSTEM CONCEPT

ControlCIT, which is an acronym for Control Computer for Intelligent Transport systems, is characterized by a control design framework, consisting of a library of dynamic vehicle models and model based control strategies, and an implementation framework comprising embedded hard- and software as depicted in Fig. 2.

Both the control design framework and the implementation framework involve safety aspects as part of an integral safety concept. All phases of the design process are supported, amongst which are dynamic modeling, control design, rapid control prototyping/hardware-in-the-loop simulation and finally the (semi-)embedded application.

Practically speaking, this concept takes the shape of a generic toolbox enabling the design and implementation of commercially applicable control systems for AVG. The next few sub-sections will further explain the components shown in Fig. 2.

3.1 Control design framework

The control design framework incorporates all necessary tools in order to develop AVG control strategies by means of computer simulation. To this end, the control design framework includes or will include the following components:

- position estimation methods,
- model based control strategies,
- a software library containing dynamic vehicle models suitable for controller development, including tire, drive line and chassis models as well as sensor models.

The dynamic model library is set-up in MATLAB/Simulink™ and gradually evolves over various research projects. Fig. 3 shows the top level in the library as well as a sub-library containing models which describe an entire vehicle and an AGV. These models are constructed from other library models describing the vehicle subsystems. Note that the dynamic model library is currently redesigned so as to fit within a more extensive dynamic model library called ADVANCE [2].
For AGV's, the main control problem can be characterized as path tracking. Because of the relevance of developing generic path tracking solutions for a wide class of vehicles, this paper concentrates on this issue.

In order to arrive at a generic applicability for this type of applications – requirement iii above – the following control strategy is applied. A lower level controller realizes a so-called Virtual Actuator, i.e. an imaginary actuator in the center of the vehicle. The Virtual Actuator can be a force actuator, a speed actuator, or a combination of both and is a controller in itself. The level above the Virtual Actuator contains a position controller which aims to realize a certain reference trajectory \( s_{\text{ref}}(t) \) of the AGV, for instance expressed in terms of the \( x,y \)-position and the orientation \( \psi \) of the vehicle as a function of time \( t \):

\[
\begin{bmatrix}
x_{\text{ref}}(t) \\
y_{\text{ref}}(t) \\
\psi_{\text{ref}}(t)
\end{bmatrix}^T
\]

To this end, the position controller uses the Virtual Actuator as the means to move the vehicle. Note that this trajectory definition describes the case where the vehicle platform is capable of independently realizing an \( x,y \)-position and orientation \( \psi \). The Virtual Actuator together with the position controller constitute the tracking controller, a simplified block scheme of which is shown in Fig. 4.

The basic notion of a Virtual Actuator combined with a higher level position controller can be applied to a large set of AGV platform configurations with respect to driven and steered wheels. It can therefore be characterized as generic. Moreover, this approach enables the position controller to be truly independent of the type of vehicle, whereas the Virtual Actuator depends on the platform configuration. The latter may incorporate optimization techniques in order to make optimal use of the available actuators, see e.g. [3].

Application of the model library and the control strategy has already proven to have the potential for high performance path tracking for AGV’s, as will be described in the ‘Application test case’ section. As a consequence of requirement i however, safety and reliability also play an important role. Especially the safety aspect links the control design framework to the implementation framework, which is why the latter will be described first.

### 3.2 Implementation framework

The implementation framework establishes the real-time implementation of the designed controller, enabling hardware-in-the-loop tests/rapid control prototyping as well as the embedded application of the control system. The implementation framework covers the following aspects (besides the safety concept, that will be dealt with later in this paper):

- software architecture definition,
- hardware architecture definition,
- real-time communication support,
- explicit reliability measures.

The software architecture is chosen to provide a direct link between control design using non-real-time simulations and real-time testing. To this end, the control design takes place in MATLAB/Simulink\textsuperscript{TM}, which can then be compiled to real-time C-code using the Real-Time Workshop toolbox in MATLAB. Execution takes place under the Real-Time Linux operating system, being widely supported freeware. In order to ensure real-time execution of the control algorithm, the latter is compiled as a (loadable) Linux kernel module.

A Real-Time Linux version of the Simulink runtime environment was also developed, establishing links with TCP/UDP networking, user-mode processes, etc.

The hardware architecture is designed to enable the embedding of safety functionality as well as redundancy, the latter contributing to the reliability of the system. The basic architecture consists of the following components:

**Generic Controller** – The Generic Controller (GC) is the ‘core’ of the control system, implementing the actual controller as well as high level (intelligent) safety related functions. The GC is a PC/104 based system, being a widely adopted standard for embedded controllers. Fig. 5 shows a photograph of the system. Note that the GC is not limited to PC/104, as long as it is a PC-based system capable of running Real-Time Linux. In accordance with its generic character, the GC has two types of interfaces: Ethernet and CAN. Ethernet provides a flexible bus including the possibility for wireless devices, with a relatively high data throughput,
Vehicle Interface – The Vehicle Interface (VI) provides the interfacing between the Generic Controller and the vehicle platform. To this end, the VI performs all kinds of signal type conversions. These signal conversions are performed through another MATLAB/Simulink™ programmable PC/104 system and/or by so-called IO-modules. The first, that is the PC/104 system, performs I/O directly with the aid of peripheral PC/104 hardware, like AD-and DA-converters, quadrature counters and digital I/O. IO-modules are common automotive components providing a means to convert to and from CAN bus signals as well as performing consistency and plausibility checks. The PC/104 part and the CAN IO-modules are complementary in a sense that the latter are very robust while the PC/104 is more flexible and has more computing power. A well considered distribution of the actuator and sensor signals across both types of interfaces is therefore quite important. Note that although the concept of the VI is generic, its specific implementation is not.

Safety Box – The last main component is the ‘low intelligence’ Safety Box, which performs regular plausibility checks of the vehicle status. Also the proper functioning of the Vehicle Interface and the Generic Controller is checked by means of watchdog signals. As described in the ‘Safety Concept’ section, the safety circuitry is the last resort in the incident handling procedure. As a consequence it has to be very reliable which is why it is PLC-, PLD- or relais-based.

3.3 Safety concept

The essence of the safety concept underlying ControlCIT is that a number of possible safety actions are distinguished. In case of AGV tracking control for instance, these can be defined as:

i. Obstacle Avoidance – Attempt to find a free trajectory in the presence of obstacles.

ii. Controlled Emergency Stop – Apply the tracking controller to follow a full-stop trajectory.

iii. Autonomous Emergency Stop – Aim for a full uncontrolled (i.e. feedforward only) stop.

iv. AGV Shut Down – Shut down the entire power supply and use the mechanical fall back braking system.

Each of these actions has a specific level of reliability, intelligence/complexity and ‘restartability’, i.e. the time and effort it takes to restart the AGV again after the incident. The failure modes which can be identified with respect to the AGV and its specific application, are
assigned to one of these safety actions according to the required level of reliability, intelligence and restartability. This concept is visualized in Fig. 7 with a number of example failure modes. Of course it is possible to define additional safety actions such as the possibility to operate under limited speed in case certain sensors fail to operate properly.

Fig. 7 Safety concept (referring to an AGV application)

This safety concept, which can be characterized as a design methodology, structures the initial design of a safety architecture as part of the control system. Moreover, it has the advantage that there is a direct link with a tool for Failure Mode Effects and Criticality Analysis as currently being developed by TNO Automotive [4]. As a result, the safety analysis can be carried out efficiently and effectively. An example of a safety architecture established using this design approach is illustrated in the next section, which describes a first application test case of ControlCIT.

4. APPLICATION TEST CASE

TNO recently established a hardware-in-the-loop test bed for testing intelligent road vehicles, i.e. vehicles equipped with ACC, Lane Keeping, Stop & Go, etc. This test bed, called VEHIL (VEHICLE Hardware In the Loop), essentially implements a virtual traffic scenario comprising the dynamic behavior of an ‘intelligent’ test vehicle within a traffic flow [5], [6]. The test vehicle may use virtual (simulated) sensors to collect information about the surrounding world. In order to achieve a higher fidelity and/or evaluate real sensors, relevant neighboring vehicles are represented by real vehicles so as to provide realistic sensory input. To this end, the vehicle to be tested is placed on a roller bench whereas the surrounding traffic is emulated using Moving Bases. These Moving Bases are in fact high performance AGV’s capable of performing the complex maneuvers corresponding to the motion of other road users relative to the test vehicle. In order to achieve this high maneuverability, the Moving Base has four independent driven and steered wheels, all of which are powered by separate electrical drives. Table 1 below summarizes some important characteristics of the first Moving Base; details can be found in [7].

The Moving Base, a picture of which is shown in Fig. 8, serves as a first test case for the application of part of ControlCIT, being the tracking controller and the safety concept.

Table 1 Moving Base specifications

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle mass</td>
<td>480 kg</td>
</tr>
<tr>
<td>Wheel base x track width</td>
<td>1.4 x 1.4 m</td>
</tr>
<tr>
<td>Maximum driving speed</td>
<td>50 km/hr</td>
</tr>
<tr>
<td>Maximum acceleration</td>
<td>10 m/s²</td>
</tr>
<tr>
<td>Installed power</td>
<td>30 kW</td>
</tr>
<tr>
<td>Acceleration time 0 – 50 km/hr</td>
<td>2.1 s</td>
</tr>
<tr>
<td>Battery pack</td>
<td>NiMH D-cells, 350 V, 75 kg</td>
</tr>
</tbody>
</table>

Fig. 8 The Moving Base application

The implementation of the safety concept resulted in a safety architecture, which is shown in Fig. 9 by means of a detection–processing–action scheme. Note that the ‘detection’ and the ‘processing’ components in this scheme directly relate to the system architecture components shown in Fig. 6 whereas the ‘action’ components directly relate to the safety actions in Fig. 7.

The maneuverability of the Moving Base is illustrated in Fig. 10, showing the measured response to an 8-shaped position reference trajectory according to:

$$s_{ref}(t) = [x_{ref}(t) y_{ref}(t)]^T$$

i.e. with constant orientation $\psi_{ref}(t) = 0$. The position reference track aims at a cruising speed of 5 km/hr.

Fig. 10 Moving Base 8-track response

Deviations of the actual position from the reference track can hardly be distinguished in this plot and appeared to be less than 6 cm. The test was carried out with a path control strategy without the velocity and acceleration feedforward information, but with laser position information for the feedback control loop.
5. CONCLUSIONS

ControlCIT is a generic toolbox for the design and implementation of control systems for automatic vehicle guidance. Consisting of mainstream hard- and software, it aims to provide an integral approach to vehicle performance, safety and reliability. The control and the safety philosophy of ControlCIT have been successfully tested using an in-house developed AGV. Current research is aimed towards the practical evaluation of the hardware component of ControlCIT with respect to performance, reliability, durability, etc., using the same AGV.

Future development is directed towards the application of either ControlCIT as a whole or parts of it to various types of automatic guided vehicles and intelligent road vehicles. During this process we aim to gradually improve and extend this toolbox.

REFERENCES


Fig 9. Safety concept implementation (example)