Paper number TP-0205

Designing a Dual-display HMI System for Connected Vehicles: the AutoNet2030 Approach

P. Pantazopoulos[1*], L. Altomare[2], K. Pagle[1], A. Toffetti[3], P. Lytrivis[1], A. Amditis[1]

1. Institute of Communication and Computer Systems (ICCS), Athens, Greece
Iroon Polytechniou Str. 9, GR-15773, tel. +30-210-7271076, email: ppantaz@iccs.gr
2. Centro Ricerche Fiat S.C.p.A, Trento, Italy
3. Centro Ricerche Fiat S.C.p.A, Orbassano (TO), Italy

Abstract

Connected vehicles, whether manually-driven or of higher automation level, typically rely on HMI applications to provide the driver with maneuvering advices and/or information about road and vehicle events. However, in view of the increased awareness capabilities provided by the cooperative V2X messaging, HMI systems are now required to leverage information beyond the scope of any ego-vehicle perception system and provide it to the driver with the maximum possible clarity. The paper seeks to address this challenge by presenting the design of an innovative dual-display approach to HMI visualizations for connected vehicles, as specified in the context of the AutoNet2030 EU project.

Keywords

ADAS, Human Machine Interaction (HMI), Dashboard display technologies.

Introduction

The last years have seen an increase in the use of wireless V2X communications aiming to improve the safety of the vehicular fleet. In parallel, the introduction of automated driving functions shifts the role of the driver from a manual controller towards a system supervisor. The main cooperative scenario that becomes relevant in both cases suggests the exchange of information with neighbouring vehicles and its processes to notify the driver or provide manoeuvring advices. A key challenge then is to efficiently communicate the increasingly rich information and interact with the driver through carefully-designed HMI applications.

Central in the so-far (non-auditory) proposals is the use of a head-up display (HUD) that provides information through digital graphics close to the driver's natural line of sight [1]. The visual objects can be organized and projected using a display separated into three main components as defined in the HAVEit project [2]. A more concise approach developed in the

InteractIVe project, informs the driver about the position of a potential threat using a bird's-eye view of the vehicle and associated safety-shield icons [3]. To cope with all the available information (to be projected), in the AutoNet2030 project (www.autonet2030.eu), we have carried out a multidisciplinary work (with engineers and cognitive experts). We have designed and implemented an innovative dual-display HMI system that provides the user with information in a distraction-minimizing way regardless the automation level.

To interface with the driver of the manually-driven and cooperative prototype vehicle we employ HUD projections that realize our primary display and (next to it) we mount a touchscreen Android tablet serving as a secondary display. Both of them are implemented through Android applications; the former uses a tablet as the HUD projector to show the most important information and maneuver advice over an appropriate windshield film. Complementary information (text messages/images) as well as interactive buttons are displayed on the second Android that is mounted over the manually-driven vehicle. With this design approach, presented next, we clearly distinguish between critical events and others of only informative nature. On the other hand, under prototypes with automation capabilities, we use a variation of the secondary display design to project a synopsis of useful information to the passengers, over a single Android device.

HMI System Architecture

The automotive setting as considered in the AutoNet2030 project includes a dense V2X communication layer providing connectivity (see Figure 1 left) to the different vehicle platforms (of the realized prototypes) and the corresponding on-board sub-systems[1]. From an on-board standpoint, HMI is one of the four main components (together with the Manuever Controller, the Perception and the Communication Services) that realize the AutoNet2030 functionality (see Figure 1 right).

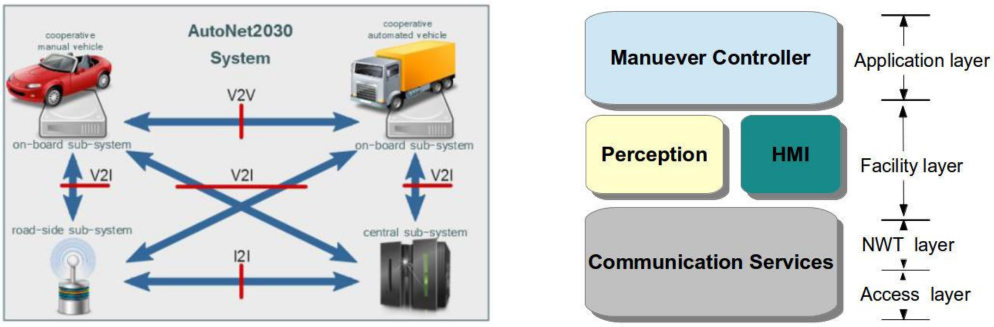

Figure 1 - Left: The AutoNet2030 system and its communication links. Right: Main components of the on-board AutoNet2030 system.

1 The main focus of the AutoNet2030 implementation has been on the vehicle platform functionality; the central sub-system depicted in the rightmost corner of Figure 1 left has not been implemented but can serve as a future extension to support a traffic management system.

The development of an ergonomic HMI can be considered as a natural complement to the AutoNet2030 perception (*i.e.*, precisely, to the Local Dynamic Map concept); pieces of information that the latter stores can be projected to the driver hiding the underlying complexity. In the AutoNet2030 context (and more generally in the connected vehicles paradigm), the role of such an HMI system is twofold: a) to display maneuvering advices to drivers of manual-cooperative vehicles and/or b) to inform the driver/passengers of an automated-cooperative vehicle about upcoming maneuvers and road events. Information originating either from the sensory equipment of the vehicle or from V2X communications has to be presented to the driver and passengers in an intuitive way and format. Furthermore, in the case of manually-driven vehicles any directives for driving maneuvers need to be projected in a way that allows the driver to easily observe the environment, rather than causing any distraction. The AutoNet2030 ergonomic HMI system has been implemented into two modules inline with a carefully-designed architecture that is able to cope with different levels of automation. Those modules are described below.

HMI Support module

The first module is called HMI Support (Figure 2) and aims to collect and organize the bidirectional traffic between interested applications of the internal system and the HMI module. Moreover, it decouples the HMI application logic from any OEM-specific characteristics.

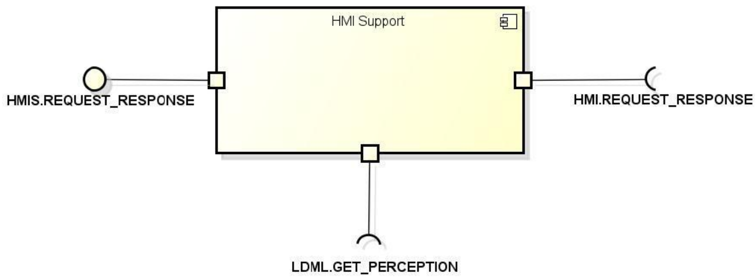

Figure 2 – Design and associated interfaces for the HMI Support module

In terms of the involved interfaces and the accommodated information, the module establishes a communication channel with the Maneuver Controller and the (core module of the) HMI System. The former interface (*HMIS.REQUEST_RESPONSE*) relies on Lightweight Communications and Marshalling (LCM) messages [5] to accommodate velocity profile (current *vs.* target speed), lane-index/lane-change information and convoy status, as estimated by the Maneuver Controller. LCM is a set of libraries and tools for message passing and data marshalling, tailored for systems where high-bandwidth and low latency is critical. It provides a publish/subscribe message passing model and automatic marshalling/unmarshalling code generation with bindings for applications in a variety of programming languages. Accordingly, an LCM listener is implemented in the HMI Support module.

The latter interface (*HMI.REQUEST_RESPONSE*) accommodates an internal communication channel with the HMI Management module which handles the connected devices. Essentially the information sent towards the HMI to drive the projections over those devices is a combination of data coming from the Controller and the Perception module. Information coming from the sensors *i.e.*, Perception module, such as the number and coordinates of tracked surrounding vehicles and the information of the current lane occupied by the vehicle, are retrieved by the HMI Support module (through the *LDML.GET_PERCEPTION* interface from the digital map LDM sub-component); thus, the system is enabled to provide more informed HMI visualizations.

In a nutshell the main sub-functions implemented by HMI Support are:

- Collect the content from requests sent by applications/facilities to the HMI system.
- Receive data from the HMI system, *e.g.*, returned data after a human interaction. The data will be sent to the appropriate application or facility component.
- Retrieve data from the Perception component that are useful for the HMI system internal logic and projection efficiency

HMI Management module

The second module (Figure 3) realizes the core the HMI functionality analyzing the content of incoming requests and providing the driver with the necessary notifications, interaction with the system and/or maneuvering advising. Our design approach allows to support more than one display using a single HMI instance (in the vehicle platform) regardless the current driving mode. HMI includes three sub-modules, namely the HMI Management Services, Scheduler and Device Handler.

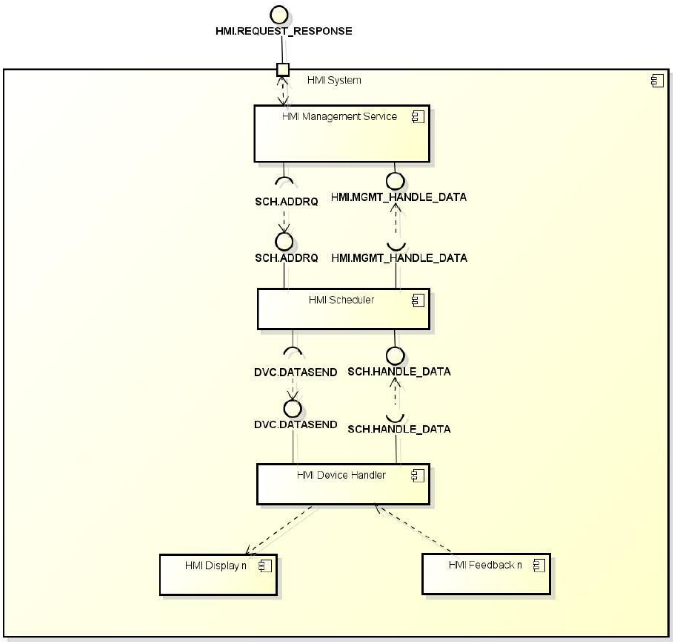

Figure 3 - Design and associated interfaces for the HMI Management module

The first provides the HMI request-response functionalities (to the HMI system) serving as an access point that facilitates the delivery of the information exchanged over the *HMIS.REQUEST_RESPONSE* interface to the second sub-module. The corresponding messages keep (for the sake of efficiency) the same format and carry the same information as they are forwarded from the HMI Support to the HMI Scheduler.

The HMI Scheduler has been designed to carry-out priority-based scheduling of the (incoming) HMI messages. It can therefore support a logic *i.e.,* a way to prioritize the messages with respect to a considered set of their attributes. Then, by using a predefined set of priority values for each message (associated to a considered use case), one only needs simple look-ups to determine the priority of each message. In the considered *cooperative* use cases that involve high-speed vehicles, a simplified strategy has been applied to prioritize the HMI requests. The only required distinction is made for a lane-change directive that needs to be forwarded in higher priority than simple speed adjustment messages. This module, analyzing the content received from various sources (*i.e.,* Maneuver Controller and Perception components) defines specific commands to enable graphical/textual contents that will be projected by the (final) display devices. These commands are provided to final devices, with JSON messages using a TCP Server that manages the connections to the (final) Android devices via a Wi-Fi channel. Essentially the information sent towards the HMI to drive the projections over those devices is a combination and a filtering of data coming from the Maneuver Controller and the Perception module.

The third one *i.e.,* the Device handler, manages the bi-directional WiFi connection with the display-devices in a transparent way for the upper level. The usage of the wireless channel to access the Android device relies on well-established procedures (*i.e.*, through a network SSID applying the appropriate permissions). Then, the component serves two main purposes regardless the number of the final display devices:

- To send to the displays any HMI request and relevant content to be projected according to the designed AutoNet2030 wire-frame (detailed in the next Section).
- To receive feedback/inputs from connected User-Interface devices (e.g., Android).

Dataflow from an HMI standpoint

To obtain an overview of the introduced modules and their interaction, Figure 4 illustrates the data flow from the HMI system perspective, in the Autonet2030 prototypes.

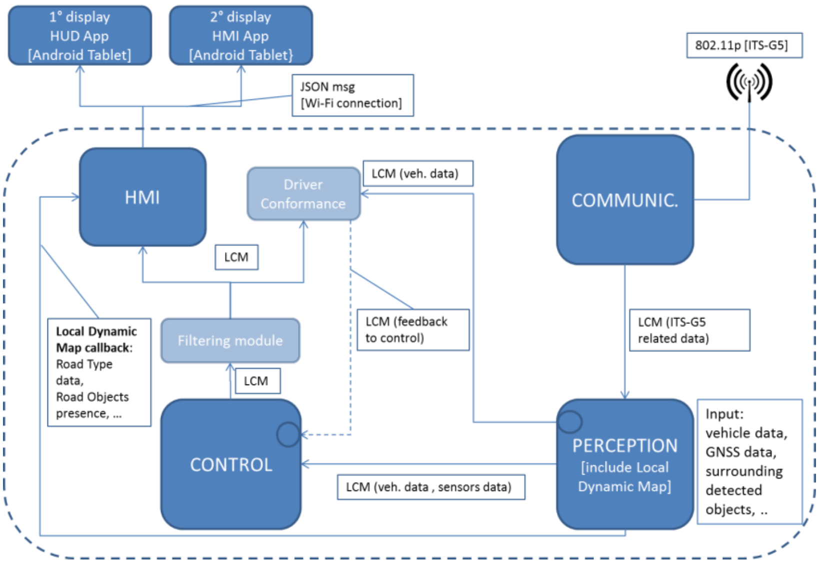

Figure 4 - Modules interaction in the AutoNet2030 architecture

LCM format is used to define the majority of the exchanged messages (as is the case for the MC directives received by the HMI). Since the Maneuver Controller component is primarily intended for vehicle-automation capabilities (*i.e.*, high MC output rate), a Filtering module is present between MC and HMI components. Its aim is to reduce and harmonize the number of directives generated by MC component, and provide an output rate compliant to the human ability to perceive a message. To constantly check the Driver Conformance to directives provided by MC, a dedicated module is running in parallel to the HMI. This module compares the directives from MC with the vehicle dynamic data, to determine whether the driver is following correctly the system's directives.

Wire-frame of displays

The efficient operation of the above software components requires a carefully designed wire-frame for the display devices. Our specification work amounts to the careful break-down of the available displays into a number of sub-areas where display objects or text messages (when possible) are projected to advise/inform the driver in line with any road/vehicle event. Emphasis lies on the reduction of information cluttering and distraction risks. Our aim is to project concise information for both manual and automated driving mode. Figure 5 shows the AutoNet2030 wire-frames for the HUD and secondary device.

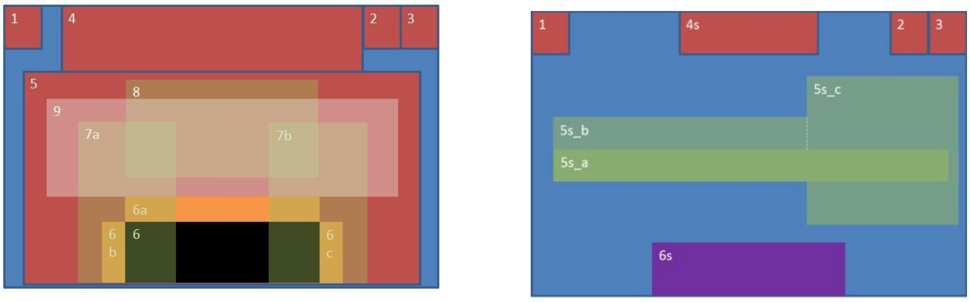

Figure 5 Sub-areas of HUD wire-frame (left) and secondary device (right)

Certain piece of information is projected over the defined wire-frame of the devices in a consistent way so that the driver can quickly understand (it). The effort was to keep some basic information projected in the same way across the screens

HUD

The head-up display is designed to project directives to the driver as well as the main display elements (*i.e.*, road, arrows, vehicle shields etc). The various pieces of information that is projected over the sub-areas of Figure 5 (left) as well as some indicative encoding (*e.g.*, selected colors) are presented in the following table.

Table 1 – Wire-frame specifications for HUD

Sub-area ID	Projected Information
1	Driving mode (M: manual, A:automated)
2	Convoy/Platoon motion (C/P letter)
3	Local Cooperative Area icon
4	Target speed assistant
5	Road with one/two/three lanes
6	Ego vehicle in the color of the current modality (*e.g.*, automatic in blue)
6a	Frontal shield colored inline with the situation criticality
6b/6c	Left/Right lateral shield
7a/7b	Left/Right arrow for left lane change
8	Preceding vehicles
9	Auxiliary sub-area

The central sub-area (no. 5) presents the road and (optionally) visual objects shown from an ego-vehicle point of view. The selection of the road type to project as a background reflects the digitalized map available through LDM (Local Dynamic Map as part of the Perception component) and the map-matching procedure over this digital map. Vehicle-status information (*e.g.*, manual or automated driving mode) and directives/informative messages appear in the top horizontal area. The projection in sub-area 4 bears some comments. It is dedicated for the visualization of *current* vs. *target* speed information, with a graphical solution of a "target speed assistant" widget. The motivation of this approach is twofold: (1) text messages have to be omitted from the HUD for clarity reasons; (2) Typically, the controller directive to the driver consists of a suggested speed and lane to follow so a dedicated area to present this advice is needed. The arrows (of sub-areas 7a, b) will be projected (and hidden) according to the Maneuver Controller decision to allow proper execution of the lane change maneuvers, as defined by the AutoNet2030 system.

Secondary device

A wire-frame configuration of similar design philosophy has been introduced for the Android device which -in this case- maintains an I/O capability *e.g.,* touch screen. The sub-areas of Figure 5 (right) that project different content compared to the HUD (Figure 5 left) are summarized in the following table:

Table 2 – Wire-frame specifications for Secondary device

Sub-area ID	Projected Information
4s	AutoNet2030 Logo
5s_a	Short text explanation (no directive message) of requested maneuver
5s_b	short text explanation (no directive message) of requested maneuver
5s_c	Graphical elements which comes with written feedback (e.g., traffic jam icon)
6s	Interactive area with a virtual central button

The Android HMI application (installed in the secondary display) can cope with different driving modes and operate both in-parallel with the HUD or as a standalone visualizer. The use of the interaction button of sub-area 6s enables the user to switch the driving mode, given that safety conditions are met.

Alternative configuration

This wireframe will be used in vehicles operating under autonomous mode. In this case the expectation is that only an Android device will be present and acting as the Secondary Display (no HUD activated). Note that this Android device can be either an external tablet mounted on the dashboard or a Android-compatible screen of the vehicle’s (entertainment) system. The aim of this alternative configuration is mainly to inform the passengers/driver of AutoNet2030 automated prototypes, about the directives that the system (i.e. Maneuver controller component) has taken and executed during the cooperative maneuvers.

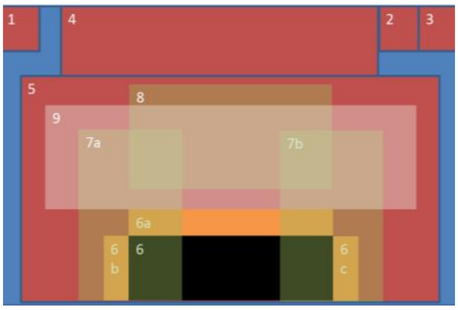

Figure 6 – Sub-areas of the Android device screen under the alternative configuration

The proposed wire-frame for this case is essentially a combination of the previous two (see Figure 6). The central area of the screen (*i.e.*, all sub-areas included in sub-area 5) reproduces the HUD projections but only for informative reasons (rather than providing directives). It can include the target speed assistant (as for the HUD standard configuration) while the sub-area 4 shows short text messages to complement the passenger's awareness about road/vehicle events.

Visual objects and text-messages vocabulary

Following the work of the wire-frame design, we have created a complete set of display objects that help us clearly communicate to the driver any appropriate advices/information as suggested by the Maneuver Controller. In Figure 7 we consider a lane-change scenario and present the corresponding sub-areas that will be 'involved' out of the proposed wire-frame. We also show the two corresponding snapshots (of the HUD and Android visualizer applications) illustrating how the AutoNet2030 display objects are used.

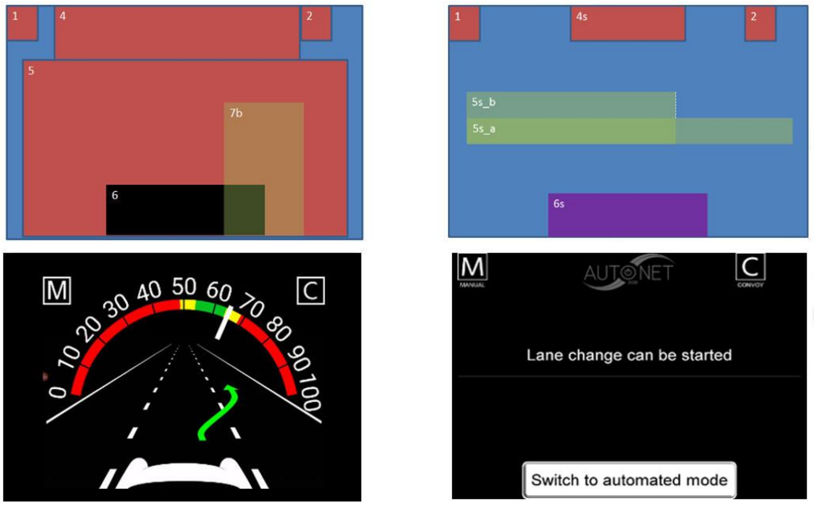

Figure 7 - Sub-areas affected and visual objects displayed in response of a specific directive (HUD: left and Secondary Display: right).

The adopted general strategy is to inform the driver projecting text messages in the secondary display that explain the rationale (*i.e.*, the reason) behind each directive (visualized in the HUD). For the sake of clarity, specification work has been carried out to define a set of rules that clearly prescribe the syntax of the projected text messages and the lexicon that is accordingly in use.

Achieved results and expectations

The AutoNet2030 HMI prototype with the two specified visualizers has been tested both in the lab but most importantly in the test-track where a number of highway use-cases were performed. These use-cases included a mixed fleet of both automated and manually-driven vehicles performing maneuvers that relate to the convoy motion (convoy formation, vehicle-merging into convoy, vehicle-leaving a convoy) and also to the cooperative lane-change (of automated vehicles).

Our first concern was to validate the operation of the software modules (see the HMI System Architecture section) as well as the end-to-end connectivity along the described interfaces. The results of our test showed that the HMI modules successfully facilitate the establishment of a communication channel and reach the final Android devices. At the same time our results provide positive feedback on the applicability of an Android device for HMI purposes *i.e.*, to serve as an HUD projector but also as a secondary device, marking that occasionally the former usage can be affected by the physical luminosity conditions (see Figure 8 left).

Figure 8 - AutoNet2030 HUD (left) and secondary device (right) realistic projections

In our test-track experimentation we validated the capability of the proposed HMI design to provide both sufficient situation awareness (*i.e.*, view of road lanes and surrounding vehicles) and realize fail-safe advised maneuvering (*i.e.*, directives and warning signs). The cooperative driver needs to maintain his current speed "in-line" with the target speed suggested by the AutoNet2030 Maneuver Controller. With the help of our widget, the HMI system has been found to enable the driver to keep the speed deviation from the suggested speed into the suggested interval. Moreover, the driver is notified with a simple yet informative way when a lane change maneuver can start and when is correctly executed, in accordance with the MC estimations and directives.

In parallel the HMI is able to show the current driving lane, in accordance with the map matching procedure running in background inside LDM component of Perception. The corresponding information is rich and is expected to increase in volume as the sensory equipment and V2X communications are improving. In our solution (even if the AutoNet2030 system can achieve a 360^o perception) we maintain the driver's perspective, mainly considering information that summarizes the immediate surroundings of the ego-vehicle. The trade-off between the information availability and the need for efficient filtering to minimize distraction during the cooperative maneuvers is an open question expected to determine future HMI designs.

Conclusions

In light of the advanced perception capabilities and the dense connectivity layer of V2X communications, more reach informative content is increasingly becoming available within the fleet of connected vehicles. Providing (a subset of) those data to the driver/passengers in a concise and ergonomic way is an open ITS problem that has lately attracted increasing attention. In this paper we have presented the relevant AutoNet2030 approach which introduces a distributed (*i.e.*, dual-display) HMI system employing the combination of a head-up display solution with an android device. Regardless the vehicle's automation level we have designed and implemented an intuitive wire-frame used to project directives to the driver of a manual vehicle (through an HUD and android device combination) and informative messages to passengers of automated vehicles (through a single android device). The prototype has been validated during field-testing that realized demanding highway use-cases. The system accurately projected the maneuver-controller directives and informative messages and was well-received by the drivers/users. Given those positive results, the expectation is that the introduced system can serve as the basis for the ITS community to further improve the ergonomic HMI applications and assist towards the proliferation of the connected (and automated) vehicles' fleet.

Acknowledgments

The authors would like to express their gratitude to Antonella Masala from CRF for their valuable help, insightful conversations and comments.

This work was supported by the European Commission under AutoNet2030, a collaborative project part of the Seventh Framework Programme for research, technological development and demonstration (Grant Agreement NO. 610542). The authors would like to thank all partners within the AutoNet2030 consortium for their cooperation and valuable contribution.

References

1. Albert, M. *et al.*, (2015). Automated driving–Assessment of interaction concepts under real driving conditions. *Int'l Conf. on Applied Human Factors & Ergonomics*, (pp. 4211-4218).
2. Schieben, A. *et al.*, (2011). How to interact with a highly automated vehicle. In D. d. Waard *et al.*, *Human Centred Automation* (pp. 251-267). Shaker Publishing.
3. Alessandretti, G. *et al.*, (2014). *InteractIVe EU Project - Final Report.* http://www.interactive-ip.eu/index.dhtml/docs/interactIVe_SP1_20140506v1.2-D19-Final_Report.pdf
4. M. Obst, A. Marjovi, M. Vasic, I. Navarro, A. Martinoli, A Amditis, P. Pantazopoulos, I. Llatser, X. Qian, ``Challenges for Automated Cooperative Driving–The AutoNet2030 Approach", book chapter in Springer *Automated Driving*, Editors: Daniel Watzenig, Martin Horn, September 2016.
5. Huang AS, Olson E, Moore DC. LCM: Lightweight Communications and Marshalling. 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2010. doi:10.1109/iros.2010.5649358