
Technical report, IDE1120, May 2011

Halmstad University Grand Cooperative Driving Challenge 2011 Technical Paper

Technical Report

Kristoffer Lidström, Johan Andersson, Fredrik Bergh, Mattias Bjäde,
Spencer Mak, Katrin Sjöberg



School of Information Science, Computer and Electrical Engineering
Halmstad University



Halmstad University

Grand Cooperative Driving Challenge 2011

Technical Paper

Kristoffer Lidström, Johan Andersson, Fredrik Bergh, Mattias Bjäde, Spencer Mak, Katrin Sjöberg

Halmstad University
Kristian IV:s väg 3, Halmstad, Sweden

Abstract

Traffic congestion is a large and growing problem in many countries due to an ever increasing number of vehicles. Increasing capacity by simply extending the road infrastructure is in many places impossible due to space and cost limitations. Everyday, people spend countless hours in car queues all over the world. The existing road infrastructure must be better utilized to save time and to reduce energy use. By enabling wireless communication between vehicles (V2V) and between vehicles and infrastructure (V2I) the flow of traffic can be better controlled in order to increase not only efficiency but also safety and comfort. Cooperative platooning is one way to increase efficiency by allowing vehicles to form road trains behind a lead vehicle. The Grand Cooperative Driving Challenge (GCDC) is an attempt to move towards a quicker deployment of cooperative platooning. This extended abstract gives an overview of Halmstad University's team and its technical and organizational approach in preparing a vehicle for the competition.

I. INTRODUCTION

The 2011 Grand Cooperative Driving Challenge (GCDC) aims to bring industry and academia together to build vehicles that, by communicating with each other, can achieve safe and efficient platooning behaviour. Each vehicle must be able to communicate with other vehicles and infrastructure using a communication stack based on IEEE 802.11p and CALM FAST. The vehicles will be semi-autonomous in the sense that they are automatically controlled only in the longitudinal direction. By using radio communication information about the state of vehicles further ahead can be used to design controllers that enable safer and more efficient platooning than if only the preceding vehicle is considered.

Several aspects set the GCDC apart from previous cooperative platooning and autonomous vehicle demonstrations. Firstly, the GCDC is inherently multi-vendor and teams are free to implement systems based on different vehicle types, using different sensor systems and control strategies. The multi-vendor aspect drastically affects system design as uncertainty about the behaviour and dynamics of other vehicles must be taken into account. Secondly, the GCDC is a competition where teams are judged not only by their individual behaviour but also by how well their vehicle behaves in a platoon. Still, only a single team can win the competition meaning that a balance must be found between altruistic and individualistic behaviour which adds a further dimension to the system design.

The CoAct project [1] is an umbrella organization for the three Swedish teams participating in GCDC. The different teams are connected to three universities; Halmstad University, Chalmers University of Technology, and Royal Institute of Technology. The CoAct project, jointly organized by the Viktoria institute and SAFER, also includes several industrial partners; Volvo Car Corporation, Volvo Technology, Scania, Denso, Fengco, dSpace and Test Site Sweden. Additional equipment sponsors of the Halmstad team include XSens and PreScan. Within the CoAct project workshops have been organized where technical issues have been discussed as well as real-world testing at closed tracks to ensure interoperability between the CoAct teams. The implementation of the platooning application has been made independently in every team on different hardware platforms. Halmstad University chose to engage undergraduate students in the development of the algorithms and software running on the Halmstad platform both through

a project course and through bachelor theses. Also in the non-technical parts of the competition Halmstad has leveraged student participation, more specifically media sciences students that have produced several movies showcasing the CoAct project.

The remainder of this article is organized as follows. We briefly give an overview of the GCDC scenarios to readers unfamiliar with the event in section II. In section III the system design is described. Section IV describes the development process of the Halmstad team. Finally, section V gives an overview of testing performed to date.

II. THE GRAND COOPERATIVE DRIVING CHALLENGE SCENARIOS

Two main platooning scenarios are considered in the 2011 challenge, urban and highway. The urban scenario has been designed to showcase the advantages of cooperative platooning also in low velocity scenarios as opposed to the traditional highway setting. The acceleration delay of vehicles starting when a traffic light turns green leads to an inefficiencies in intersection throughput. In this scenario, vehicles in a cooperative platoon could be made to start synchronously thus improving the flow of vehicles though the intersection before the light turns red again. In the GCDC urban scenario the participating teams are randomly divided into two platoons. Each platoon is further divided into two sub-platoons, starting at a rear traffic light and front traffic light respectively. As the rear traffic light turns green the rear sub-platoon starts moving and joins the front sub-platoon as smoothly as possible and continues as one platoon when the front traffic light also turns green. Points are awarded to the platoon with the shortest gap length when the last vehicle passes the front traffic light. (For a more detailed discussion of the judging criteria see section III.D).

In the highway scenario both platoons follow a common lead vehicle controlled by the GCDC organizers. The lead vehicle introduces acceleration disturbances and the platoons are judged relative each other on maximum platoon length, stability and safety. By reassigning teams to different platoons for each heat the most cooperative vehicle will be chosen as the winner at the end of the competition weekend.

III. SYSTEM OVERVIEW

The Halmstad platform is partially based on hardware and software used within previous vehicle-to-vehicle (V2V) communication research projects, most notably the European CVIS project [3] (Figure1). The CVIS equipment consists of an automotive grade PC (called e-box) running Linux. The e-box is equipped with three different communication technologies; 2G/3G, IEEE 802.11a/b/g (WiFi), and IEEE 802.11p. 802.11p [2] is the standard developed specifically for communications between vehicles and is used in GCDC. The 802.11p communications is enabled by radiocards from Atheros (AR5413). The antenna pod, also depicted in Figure 1, contains antennas for 2G/3G, 5.9GHz (802.11a/p), 2.45GHz (802.11b/g), as well as GPS.

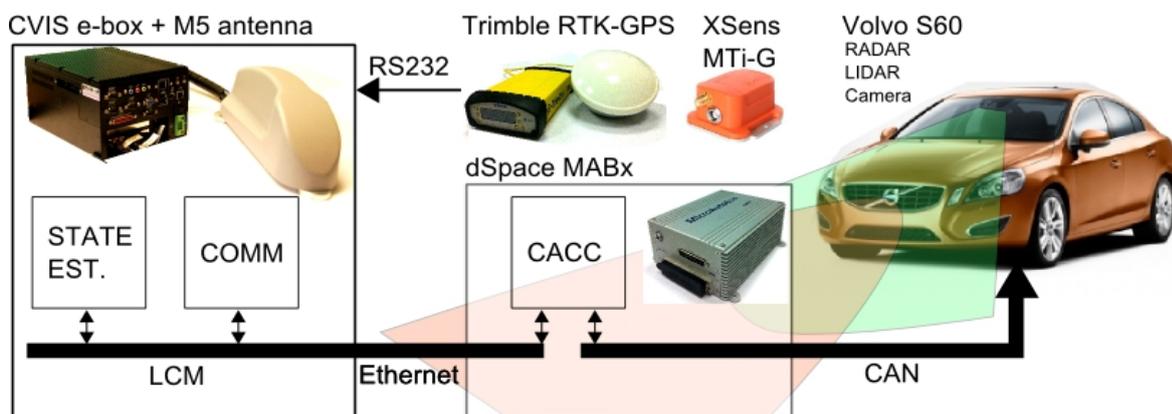


Figure 1. Overview of hardware and software components used.

Processes running on the e-box are state estimation, which fuses data received wirelessly from other vehicles with data observed from in-vehicle sensors such as wheel speed, real-time kinematic (RTK)-GPS and inertial sensors. The COMM module is responsible for receiving and transmitting information between the vehicles according to the GCDC interaction protocol [3].

For inter-process communication (IPC) on the CVIS box the Lightweight Communications and Marshalling (LCM) framework is used [4]. LCM was developed by MIT and successfully used in their vehicle “Talos” for the Darpa Urban Challenge 2007 [5]. The LCM generator tool generates code for message passing between processes for different programming languages such as C/C++, Java, Python, MATLAB and C#. LCM messages are passed between processes in a multicast fashion with the user datagram protocol (UDP) as carrier. The processes are themselves responsible for filtering out relevant messages.

The cooperative adaptive cruise-control (CACC) algorithm has stringent requirements on latency and reliability. Therefore, the CACC is executed on a MicroAutobox (MABx) hardware platform supplied by dSpace, which enables real-time process execution. The controller code for the CACC is developed in MATLAB/Simulink and the functionality verified in the PreScan [6] simulation environment. The Simulink models are then compiled into C code and downloaded to the MABx. The in-vehicle controller area network (CAN) is the interface between the MABx and the sensors and actuators of the vehicle. The CAN bus interface allows for reading out sensor values (e.g., radar) and giving commands to the cruise-control module in the vehicle in order to accelerate/decelerate. The necessary actuators and sensors are already available in a production Volvo S60, thus no mechanical modifications to the vehicle have been necessary.

A. The Vehicle

The Halmstad University team uses a 2010 Volvo S60 (Figure 2) lent to the team by Volvo Car Corporation as the main competition vehicle. The vehicle is equipped with an automatic gearbox and a 205HP diesel engine. Additionally, it has a sensor package consisting of radar, lidar and camera, which are used in the production vehicle to provide adaptive cruise control (ACC), collision warning and collision mitigation functions. The bulk of the hardware used is installed in the trunk of the vehicle where a power distribution box controllable from the front seats provides 12V from the vehicle power supply. A battery back-up additionally provides power to selected systems when the 12V supply is not available, such as when starting the vehicle.



Figure 2. The Halmstad University GCDC 2011 vehicle, a Volvo S60.

To meet GCDC safety requirements, an emergency button is integrated next to the driver's seat. In case of emergency, pushing the button will disable the power supply to the MABx thus disabling the custom CACC controller. There are also functions that disable the system when pressing the brake or gas pedals. Although not required by the GCDC organizers safety measures have also been added during start up, e.g., the system will only be activated when the driver's seatbelt is fastened. A set of flashing lights have been installed on the front and the back of the vehicle to indicate the state of the controller. When the system has been disabled or deactivated, the lights will flash red. Otherwise, if the vehicle is in CACC mode, they flash green. A router provides wired and wireless (WiFi) networking within the vehicle as well as from the outside. Mounted on the vehicle are antennas for RTK-GPS, RTK-GPS radio corrections, WiFi router, CVIS box and reference GPS for the inertial system.

B. Positioning and State Estimation

The GCDC competition rules dictate that vehicles should provide their absolute position with an accuracy "below one meter". In addition, the competition organizers provide a base station for GPS corrections, which led to the use of an RTK-GPS to achieve the necessary absolute positioning accuracy. An RTK-GPS uses carrier phase calculations and corrections received via radio (400MHz) from a fixed base station to calculate receiver position with centimetre accuracy. The Trimble SPS-852 was chosen by the three Swedish teams as RTK-GPS receiver. It is a self-contained system with built-in battery backup and radio modem. The receiver delivers NMEA formatted position messages via serial cable (RS-232) at a maximum rate of 20Hz.

Although the RTK-GPS typically provides high accuracy position information it also has drawbacks that must be addressed. When satellite visibility is low, such as when driving under a freeway overpass or close to tall obstacles, positioning accuracy is quickly degraded. Additionally, since the GPS provides only a point measurement, heading information is very poor at low speeds. To counter these effects GPS data is combined with wheel speed and steering wheel angle measurements from the CAN bus together with data from an inertial sensor using an extended Kalman filter (Figure 3). The filter uses CAN bus measurements in the prediction stage and GPS and inertial measurements in the update stage, for an in-depth overview of a similar filter structure see the work of Rezaei and Sengupta [7]. The inertial sensor used is the XSens MTi-G which is a full attitude and heading reference system (AHRS) that fuses data from built-in accelerometers, gyros, magnetometers, barometer and GPS. In Halmstad University's GCDC vehicle the MTi-G is used to provide reliable heading velocity and acceleration data.

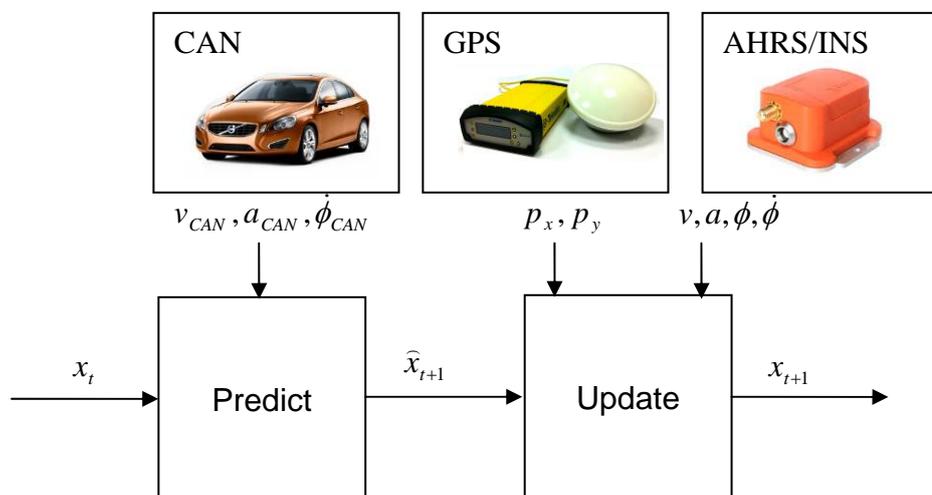


Figure 3. Vehicle state is estimated using data from vehicle CAN, GPS and AHRS sensors.

The state vector \mathbf{x} , (1), consists of location p_x and p_y , velocity v , acceleration a , heading ϕ and yaw rate $\dot{\phi}$:

$$\mathbf{x} = [p_x \quad p_y \quad v \quad a \quad \phi \quad \dot{\phi}]^T \quad (1)$$

The non-linear update stage (2) uses CAN data to predict the state vector at time $t+1$:

$$\hat{\mathbf{x}}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_{CAN}) = \begin{bmatrix} p_{x,t} + \sin(\phi_t)v_{CAN} \Delta t + \frac{\sin(\phi_t)a_{CAN} \Delta t^2}{2} \\ p_{y,t} + \cos(\phi_t)v_{CAN} \Delta t + \frac{\cos(\phi_t)a_{CAN} \Delta t^2}{2} \\ v_{CAN} + a_{CAN} \Delta t \\ a_{CAN} \\ \phi_t + \dot{\phi}_{CAN} \Delta t \\ \dot{\phi}_{CAN} \end{bmatrix} \quad (2)$$

Where $\dot{\phi}_{CAN}$ is the yaw rate, which is derived from CAN steering wheel angle measurements and a turning model for the competition vehicle. If measurement data is available from the GPS and AHRS sensors, an update is also performed. If GPS or AHRS data is not available (such as when the GPS signal is lost) only the prediction stage is performed leading to a dead-reckoning solution until the GPS signal is recovered. To calculate the prediction covariance matrix the Jacobian of the process model f is evaluated around the predicted state at each time point and combined with estimated CAN sensor noise. Figure 4 illustrates a situation during track testing when the position as reported by the GPS (green line) is erroneous due to a building next to the track, in the filtered position however these disturbances are mitigated (red line).



Figure 4. GPS disturbances are handled by incorporating data from the CAN bus and from inertial sensors.

C. Communication Protocols and Implementation

The wireless communication between vehicles in the GCDC is realized through a protocol stack containing the following components; the IEEE 802.11p standard, the IEEE 802.2 logical link control (LLC), the communication access for land mobiles (CALM) FAST protocol [8] and the GCDC interaction protocol. The 802.11p specifies the physical (PHY) layer and the medium access control (MAC) layer, see Figure 5. The data link layer is divided into the MAC layer of 802.11p and the transparent LLC. The CALM FAST covers the network layer up to the session layer of the generic OSI model [9] and the GCDC interaction protocol is situated at the presentation layer. Above the presentation layer the application is found. The communication architecture in GCDC does not use the traditional Internet protocols such as UDP, transmissions control protocol (TCP), or the internet protocol (IP), for other purposes than debugging and remote access to the vehicles. The focus of CALM FAST is as the name suggests a faster way to traverse the protocol stack in order to support road traffic safety applications' requirements on delay.

In Figure 6 the COMM module is depicted (also found in the system overview in Figure 1). The CVIS box runs the open source operating system Ubuntu, a popular Linux distribution. IEEE 802.11p connectivity is provided by Atheros AR5413 radio chipsets using MadWifi [10] drivers. CALM FAST communication is handled by a user-space daemon provided by the GCDC organizers (Figure 6). The internal communication between the COMM module and the state estimator utilizes the inter-process communication framework LCM as described in previous sections.

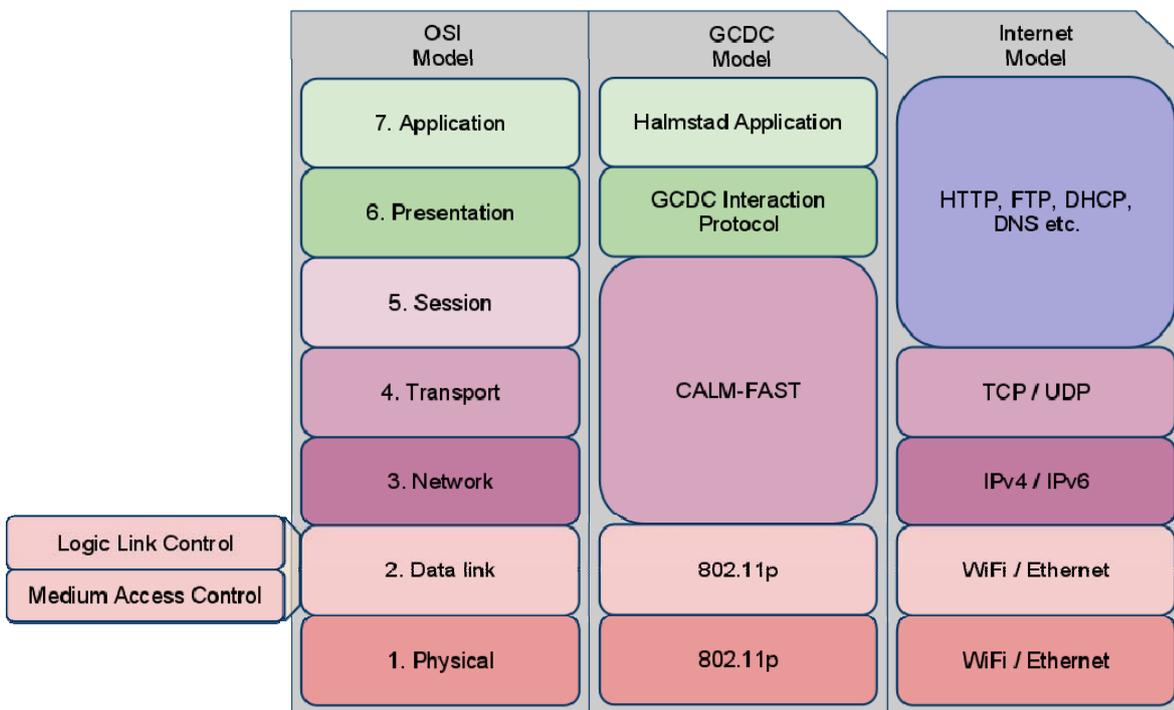


Figure 5. The GCDC protocol stack compared to the generic OSI model and the traditional TCP/IP model used on the Internet.

The application box in Figure 6 includes the GCDC interaction protocol and the actual application written in Java.

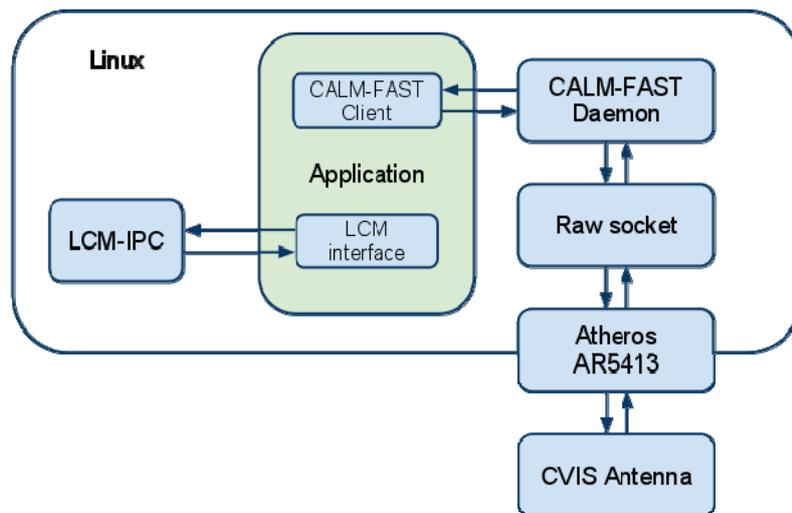


Figure 6. Overview of the COMM module.

The GCDC interaction protocol is specified by an Abstract Syntax Notation One (ASN.1) [11] provided by the GCDC organizers. ASN.1 is a standard notation that is used to describe data structures for representing, transmitting, encoding and decoding data. The ASN.1 notation and the rules which it describes are not tied to any particular computer architecture, operation system or programming language. The ASN.1 specification used in the GCDC covers all the fourteen messages that one vehicle is required to broadcast and receive to enter the challenge. In the specification there is information concerning the formatting of the messages and the data they contain, and also what data each message shall contain. The information in the messages describes the dynamics of the ego vehicle, such as its velocity, acceleration, position as well as static information such as the actual size of the ego vehicle. There are also requirements on when a message shall be sent, e.g. periodic broadcast or single transmission on demand. To extract Java classes from the ASN.1 specification of the GCDC interaction protocol the BinaryNotes [12] framework has been used, see Figure 7. The framework provides an ASN.1-to-Java compiler and a library for binary data encoding/decoding. The generated classes and the encoder/decoder are included in the application.

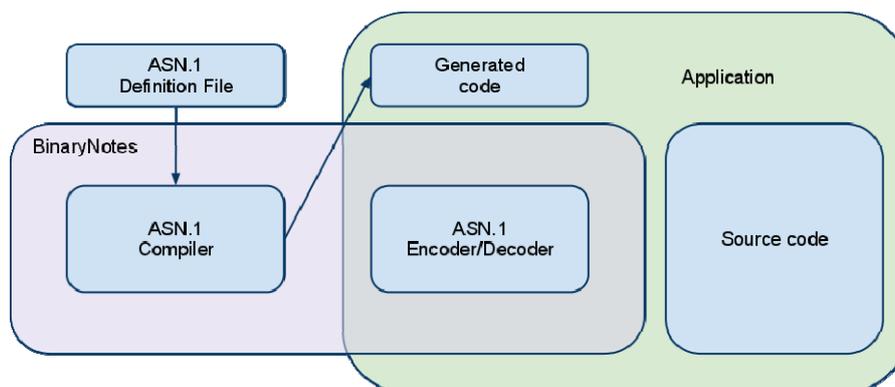


Figure 7. Overview of the application with ASN.1 specification.

The communications application in Figure 7 was created in an iterative process with three releases at Halmstad University. The first release was implemented using UDP/IP protocols. This initial release had limited functionality and was only capable of transmitting/receiving two of the predefined messages for the GCDC. However, this release confirmed the basic functionality of the Atheros card and inter-process communications between COMM module and the state estimator. In the second release the GCDC organizers had changed the communication protocol from UDP/IP to CALM FAST. This required extensive modifications to the overall software for the COMM module. This release resulted in small improvements in functionality, though it was using CALM FAST instead of UDP.

The third release is the most complete release and it supports the required functionality to participate in the challenge. For this release a completely new framework for all applications running on Halmstad University's different hardware platforms was released requiring further modifications to the COMM module.

D. Control System

The Halmstad University's Volvo S60 is equipped with a built-in sensor system and a low level cruise-control regulator that converts a desired acceleration to the required throttle and brake outputs. By combining sensor signals from radar, lidar and camera, the sensor system can detect objects in front of the vehicle and provide target information such as range, range rate and acceleration to the controller. Since the vehicle's CAN database is proprietary the required interfaces to the vehicle's sensors and actuators are provided as pre-compiled Simulink models. Limiting the need for modifying safety critical modules in the production vehicle and it also retains the vehicle's road legal status which greatly simplifies pre-competition testing and transportation.

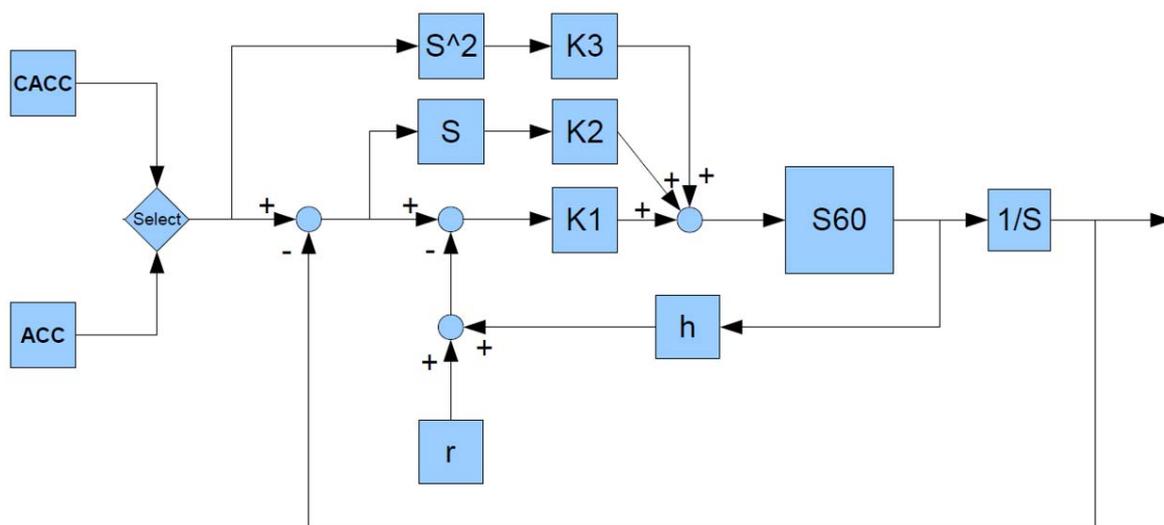


Figure 8. Schematic overview of controller structure.

The control system is hierarchical in nature. At the lowest level the cruise-control module built into the vehicle applies throttle and brakes to reach a requested acceleration. At the level above an ACC algorithm running in the MABx makes sure that the vehicle is correctly positioned in relation to the immediately preceding vehicle. In ACC mode, the controller design is based on a velocity dependent inter-vehicle spacing policy. The desired distance (3) can be described as

$$d(t) = r + h \cdot v(t) \tag{3}$$

where $d(t)$ is the desired distance, r is a velocity independent minimum gap, h is the desired headway time in seconds and $v(t)$ is the host vehicle's speed. The desired distance is used as a feedback signal to keep an optimal position in comparison with the actual distance to the preceding vehicle. In addition to distance control, range rate and acceleration are also used to evaluate the speed and the acceleration of the preceding vehicle to ensure a shorter minimum headway time. As shown in Figure 8, K1, K2 and K3 are three individual controllers for those input signals. The ACC system, which uses only in-vehicle sensors, also serves as an important safety function. In case of a radio communication failure, the safety distance to the preceding vehicle will always be guaranteed.

With a fully functional communication system, more information about the vehicles in the platoon can be provided via radio. This can greatly increase the ability to react on both upstream and downstream disturbances. By analysing this information, a predictive input can be calculated and used in addition to the ACC system, effectively making it a cooperative adaptive cruise control (CACC). What constitutes "good" or optimal platooning behaviour in the competition is defined by the following four GCDc judging criteria (criteria A-C in the highway scenario and criterion D in the urban scenario), which are weighted by the GCDc organizers with as of yet undisclosed weights []:

- A. Minimization of the maximum platoon length $L_{g,\max}$ (4) during one competition heat:

$$L_{g,\max} = \max_{t \in [t_2, t_3]} \left(L_p - \sum_{i=2}^m L_i \right) \quad (4)$$

where L_i is the length of vehicle i and L_p is the length of the platoon from the front bumper of the lead vehicle to the rear bumper of the last vehicle.

- B. Minimization of the difference (5) between the actual platoon length and the safety platoon length over time

$$v_{L_p} = \frac{1}{t_3 - t_2} \int_{t_2}^{t_3} (L_p(t) - L_s(t))^2 dt \quad (5)$$

where the safety platoon length $L_s(t)$ is

$$L_s(t) = \sum_{i=2}^m L_i + (m-1)(r + h \cdot v_{lead}(t)) \quad (6)$$

- C. String stability of the ego vehicle with respect to the platoon leader (when the ego vehicle is the second vehicle in the platoon and when it is the last vehicle in the platoon)

$$\left\| \frac{A_i(j\omega)}{A_1(j\omega)} \right\|_{H_\infty} \leq 1 \quad (7)$$

- D. Minimization of the total platoon gap length at the time the last vehicle passes the red light (t_1) in the urban scenario.

$$L_g(t_1) = L_p(t_1) - \sum_{i=2}^m L_i \quad (8)$$

The above criteria are sometimes in conflict, the safety platoon length might for example be unnecessarily long in situations when the lead vehicle is accelerating and too short when it is decelerating. This requires

a flexible controller design that can be tuned to varying weighting factors. Parameters relating to the above judging criteria are calculated within the CVIS box and sent to the MABx real-time controller via an Ethernet connection. This information is used in the CACC mode as part of the controller's input. Additional information, such as traffic light status and speed limits are also sent from the CVIS box.

To identify the dynamics of the competition vehicle system identification has been performed through step response testing and an estimated model has been created for use in the PreScan simulator. A frequency domain analysis of the control system in continuous time has been performed to define parameters for the controller. A discretized model and the actual discrete controller has then been implemented and tested in PreScan. The speed profile of a preceding vehicle from an actual track test is used in PreScan to verify the simulation models, illustrated in Figure 9.

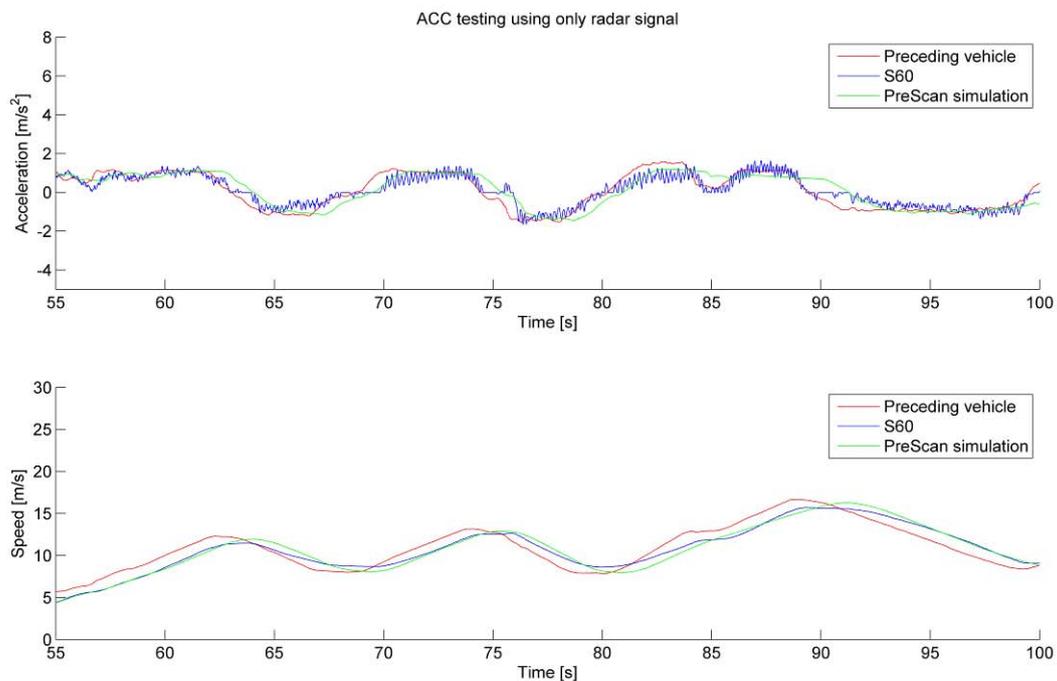


Figure 9. By using real lead vehicle traces in the PreScan simulator, the S60 model can be compared to the actual dynamics of the vehicle.

E. Human Machine Interface

To inform the driver and co-driver about the state of the system, both during the competition and during development, a software-based human machine interface (HMI) has been developed. The HMI, written in Java, runs on the CVIS e-box and allows the user to view status screens for the GPS, AHRS, V2V communications, CAN and state-estimation subsystems. The user can also suspend and restart any individual module for testing purposes. To access the HMI any WiFi-enabled device able to run a virtual network computing (VNC) client can be used. Thus it is possible to connect multiple devices for a shared view of the HMI and to access the HMI from a remote location. Figure 10 shows a dash-mounted iPad being used in Halmstad University's vehicle to access the HMI together with screenshots for some of the subsystems.



Figure 10. Left, iPad used to display the HMI. Top right, AHRS HMI screenshot. Bottom right, state estimator HMI screenshot

IV. PROJECT ORGANIZATION

The development process used by Halmstad University's team is based on an iterative model, with regular releases of the system with gradually increasing functionality. During the fall of 2010 two iterations were made; the first producing a benchtop system consisting of the basic services running against the PreScan simulator, the second producing an in-vehicle reference system running in a non-actuated Volvo S80.

During spring 2011 a third iteration consisting of an actuated and cooperative Volvo S60 was delivered. A timeline of major project milestones is presented in Figure 11.

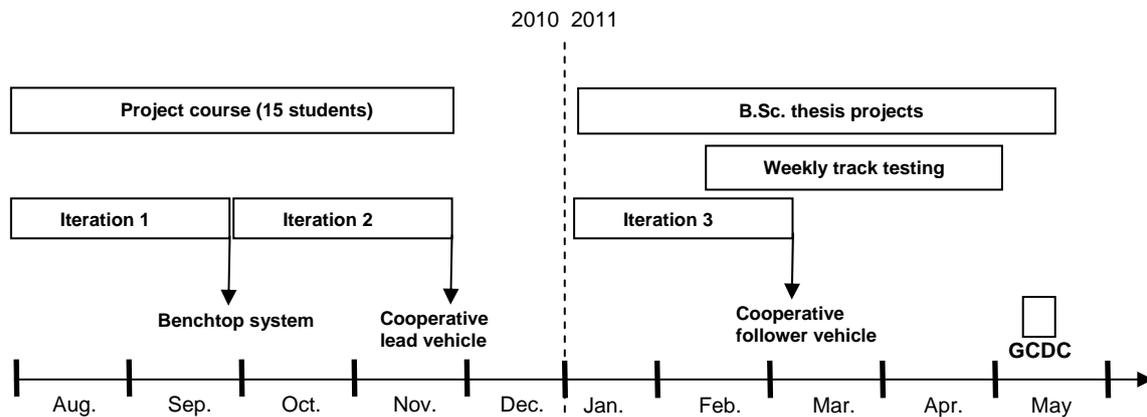


Figure 11. Three major iterations were performed with varying team member constellations

V. SYSTEM TESTING

To test the platooning behaviour independent of other teams a cooperative lead vehicle was developed. The lead vehicle (Volvo S80) contains a CVIS communication- and computation unit which communicates as dictated by the GCDC interaction protocol. However, the lead test vehicle has reduced positioning capabilities as compared to the competition vehicle, relying only on a consumer-grade GPS receiver. Additionally it cannot be automatically controlled. The main reason for using a physical vehicle rather than simulating input to the competition vehicle is to verify exteroceptive sensors such as radar and lidar. In parallel to the development of the physical systems, simulations were also performed using the PreScan simulator. PreScan provides an integrated environment for testing intelligent vehicle systems and vehicle sensors. Vehicles and control algorithms are modelled in MATLAB/Simulink and simulations are then performed in a 3D representation of the world, allowing realistic sensor outputs to be generated.

Since the vehicles are road legal only with the controller disconnected several closed track tests were performed, mainly at the Kristinehed track in Halmstad but also at the Stora Holm and the City Arena tracks in Gothenburg. During track testing two vehicles are typically used to test controller and communication performance. A joint testing workshop was organized among the Swedish teams in the CoAct project at which a total of five cooperative vehicles were present for testing in a competition-like scenario, Figure 11 shows a successful three-car platoon during the joint workshop.



Figure 11. Two actuated Volvo S60 following a non-actuated Volvo S80 lead vehicle during the joint CoAct testing workshop in March 2011.

To verify the range of the Halmstad University's car communication tests with the other Swedish teams have been conducted. The communication settings currently used is 3Mbps rate at 5.89GHz and a transmit power of 17dBm, resulting in a range of around 500 meters with all broadcasting nodes in line of sight

VI. ACKNOWLEDGEMENTS

The authors would like to acknowledge the work of the initial members of the Halmstad University's team, William Andersson, Martin Andreasson, Caroline Bellviken, Mattias Enervall, Albert Hoxha, Anders Johansson, Johan Larsson, Tommy Lidén, Martin Skog, Timmy Svensson and Erik Westerberg as well as the support from professor Ulf Holmberg. We would also like to acknowledge the assistance in testing from the Kristinehed, Stora Holm and City Arena test tracks.

REFERENCES

- [1] CoAct (n.d) [Online]. Available: <http://www.chalmers.se/safer/EN/projects/pre-crash-safety/projects/coact>
- [2] IEEE 802.11p Part 11: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 7: Wireless Access in Vehicular Environment*, July 2010.
- [3] O. Baijer, J. Jongh, A. Neerbos, B. Netten, H. Wedemeijer, "GCDC Interaction Protocol, GCDC-DOC-IP" [Online]. Available: <http://www.gcdc.net>, 2011
- [3] Cooperative Vehicle Infrastructure Systems (n.d) [Online]. Available: <http://www.cvisproject.org>
- [4] A. S. Huang, E. Olson, and D. Moore, "*Lightweight communications and marshalling for low latency interprocess communication*" MIT, Tech. Rep. MIT-CSAIL-TR-2009-041, 2009.
- [5] J. Leonard, D. Barrett, J. How, S. Teller, M. Antone, S. Campbell, A. Epstein, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, T. Jones, O. Koch, Y. Kuwata, K. Mahelona, D. Moore, K. Moyer, E. Olson, S. Peters, C. Sanders, J. Teo, and M. Walter, "Team MIT Urban Challenge Technical Report," MIT Computer Science and Artificial-Intelligence Laboratory, Tech. Rep., 2007
- [6] Hendriks, F.; Tideman, M.; Pelders, R.; Bours, R.; Liu, X.; , "*Development tools for active safety systems: Prescan and VeHIL*" Vehicular Electronics and Safety (ICVES), 2010 IEEE International Conference on , 2010
- [7] Rezaei, S.; Sengupta, R, "*Kalman Filter Based Integration of DGPS and Vehicle Sensors for Localization*" IEEE Transactions on Control Systems Technology, vol. 15, 2007
- [8] ISO 29281:2011, *Intelligent Transport Systems – Communication Access for Land Mobiles (CALM) – Non-IP Networking*, March 2011.
- [9] A. S. Tanenbaum, *Computer networks*, Prentice-Hall, 1996.
- [10] The MadWifi project. [Online]. Available: <http://madwifi-project.org/>
- [11] International Telecommunication Union. "Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation". ITU-T Standard X.680, Aug. 2002.
- [12] BinaryNotes. [Online]. Available: [http:// bnotes.sourceforge.net/](http://bnotes.sourceforge.net/)
- [13] "GCDC 2011 Rules & Technology Document, Final version 2.0". [Online]. Available: <http://www.gcdc.net>