CHRONOS Part 1

2016-02573
External report

Project within Electronics, Software and Communication

Authors: Lars Bjelkeflo, Robert Laxing, Martin Djup, Per Gustafsson, Roland Gustafsson, Niklas Lundin, Viktor Johansson, Vivetha Naterjee, Jochen Pohl, Elad Schiller, Bo Joel Svensson

Date: 2018-09-03

FFI Fordonsstrategisk Forskning och Innovation
VINNOVA Energimyndigheten TRAFIKVERKET FKG VOLVO SCANIA VOLVO

1. Content

Summary 4

Sammanfattning på svenska 5

Background 6

Purpose, research questions and method 7

5.1 Long term purpose 7

5.2 Research questions 7

Objective 8

6.1 Project management, system architecture and dissemination (WP1) 9

6.2 C-ITS use cases for connected and automated vehicles (WP2) 9

6.3 Scenarios and methods for autonomous vehicles - cars (WP3) 10

6.4 Scenarios and methods for autonomous vehicles - trucks (WP4) 11

6.5 Test platform - server part (WP5) 11

6.6 4G Network Characteristics (WP6) 13

6.7 Testplatform object part (WP7) 13

Results and deliverables 14

7.1 Project management, system architecture and dissemination (WP1) 14

7.2 C-ITS use cases for connected and automated vehicles (WP2) 16

7.3 Scenarios and methods for autonomous vehicles - cars (WP3) 17

7.4 Scenarios and methods for autonomous vehicles - trucks (WP4) 19

7.5 Test platform - server part (WP5) 20

7.6 4G Network Characteristics (WP6) 26

7.7 Testplatform object part (WP7) 33

Dissemination and publications 39

Dissemination 39

Publications 40

Conclusions and future research 41

Participating parties and contact persons 41

Annex 1 WP1 architecture team result 43

FFI in short

FFI is a partnership between the Swedish government and automotive industry for joint funding of research, innovation and development concentrating on Climate & Environment and Safety. FFI has R&D activities worth approx. €100 million per year, of which about €40 million is governmental funding.

Currently there are five collaboration programs**:** Electronics, Software and Communication, Energy and Environment, Traffic Safety and Automated Vehicles, Sustainable Production, Efficient and Connected Transport systems.

For more information: www.vinnova.se/ffi

2. Summary

New technology in the area of active safety and autonomous drive requires a parallel development of technology for test and validation. For a test track this technology often means more complex traffic scenarios that have to be orchestrated with high precision and efficiency. Complexity is driven by more advanced active safety features, testing of autonomous vehicles and the increased use of connectivity.

Chronos 1 has investigated future traffic situations for autonomous and connected vehicles with the goal of clarifying new requirements for test tracks like AstaZero. These requirements span from infrastructure support for highway merging to different communication standards. In the communication area AstaZero's 4G network dedicated for test track operation has been benchmarked with fine-tuning proposals as a result.

One main result from Chronos 1 is a prototype test platform targeting more complex traffic scenarios. This platform has demonstrated that through a well defined architecture many different partners components can be integrated to create a powerful test system. The developed platform allows for scenario design in a simulated environment, and includes several alternatives to visually plan, monitor and experience tests ranging from basic two dimensional views to realistic Unreal engine supported visualisations. In the Unreal engine based visualisation the viewpoint can easily be shifted from birds-eye view to single object perspective.

Further, the platform has been integrated towards more than six different test objects developed by five different project partners. Demonstrating basic safety features and mechanisms allowing for repeatable test scenarios orchestration. Also a first prototype of a test supervision algorithm has been developed and tested.

Another main result from the project is ongoing work in ISO where the concepts for monitoring and control of test equipment on test tracks is currently under standardisation.

Further work is already planned in Chronos part 2 (2017-05501).

3. Sammanfattning på svenska

Fordonsindustrin arbetar målmedvetet mot ett säkrare transportsystem. En central komponent i detta arbete är stegvis automatisering av fordon. En annan komponent är intelligent radiokommunikation mellan transportsystemets aktörer (s.k. ITS). Visionen är att framtida autonoma och kommunicerande fordon delar med sig och agerar på information om eget och andra fordon, fotgängare, cyklister, väglag mm. De första synliga resultaten är redan på marknaden eller nära en marknadsintroduktion, i form av exempelvis AEB (automatisk nödbroms) eller platooning (fordonståg) av lastbilar.
I all produktutveckling ingår test, verifiering och validering för att säkra att den nya funktionen/produkten uppfyller förväntningar och ställda krav på kvalitet, egenskaper och funktion. För verifiering och validering av kommande funktioner för automatisering av fordon krävs en förmåga att på ett säkert, repeterbart och effektivt sätt iscensätta komplexa kommunikations- och trafikscenarier som är potentiellt farliga (testscenarier) på provbanor som AstaZero.

Chronos 1 har haft två grova indelningar; kommande testscenarier och utveckling av en första version av en testplattform för komplexa trafikscenarier med möjlighet till inslag av kommunikation (ITS). Under projektet har testscenarier definierats för personbilar riktade mot automatisering på motorväg, för lastbilar med fokus på platooning (lastbilståg) och framtida scenarier där kommunikation är en central komponent. Ur dessa scenarier har framtida krav på testplatformen definierats och kanaliserats mot nya projekt.

Utveckling av testplattformen inleddes med etablering av en miljö för mjukvaruutveckling på AstaZero och definition av en modulär arkitektur. Mål med utvecklingen har varit att styra upp till 10 fysiska testobject och upp till 100 virtuella objekt i ett och samma testscenario. Under projektet kördes lyckade test med 100 virtuella objekt och fyra fysiskt objekt. Fler fysiska objekt är möjligt, men begränsas av objekttillgång. Fler mekanismer för kontroll av objekt har utvecklats och demonstrerats bl a statiska körfiler, triggerpunkter för synkronisering och geografisk begränsning av object. Dessa koncept har matats in i pågående ISO standardisering med mål att nå en öppen standard som möjliggör testscenarior med en blandning av olika leverantörers utrustning.

Testplattformen består av flera olika komponenter; en central server för provövervakning och insamling av loggdata, en simuleringsmiljö för skapande av testscenarior och simulering av objekt, en tre-dimensionell visualisering baserad på Unreal engine, samt ett flertal fysiska objekt. De fysiska objekten var Autolivs körrobot och överkörbara målbärare, en bil styrd av Sentients lösning, en fjärrstyrd bil, och en enhet som rapporterar position och hastighet för placering i rörliga objekt som inte styrs av servern.

En central komponent för genomförande av testscenarier med fysiska objekt, för att garantera personsäkerhet, scenarier med kravställd noggrannhet och insamling av data är kommunikation. Industristandard idag är olika typer av WLAN lösning, men Chronos 1 har använt AstaZeros egna 4G nät. Detta nät är dedikerat för verksamheten på AstaZero och ger en god täckning över hela provområdet. Under projektet genomfördes flera mätningar av 4G nätets egenskaper med mål att identifiera svagheter och möjliga optimeringar.

Planerade milstenar för projektet har uppfyllts avseende förväntade funktioner, en omfattande slutdemonstation med 5 objekt iscensatt i AstaZeros stadsmiljö, och input till framtida forskningsprojekt. Dessutom har flera exjobb, konferenspresentationer och två disputationer skett i samarbete med projektet.

Lärdomar från Chronos 1 i form av utvecklad testplattform och nya krav som mer avancerade mekanismer för att skapa testscenarier kommer förvaltas i framtida projekt, inte minst i Chronos 2 (2017-0551).

4. Background

It is foreseen that our transportation system will change a lot in the future. The major drivers will be efficient multi-modal transportation and increased safety. For these changes two technical enablers are automation and connectivity (V2V and V2X). The automotive industry will be affected in many ways, e.g.:

- Development of new technology, autonomous drive (AD)
- Increased complexity which will push for more efficient development and verification processes and tools

New technology in the area of active safety and autonomous drive requires a parallel development of technology for test and validation. This technology often means more complex traffic scenarios that have to be orchestrated with high precision and efficiency. These scenarios have to represent all types of traffic environments such as cities, rural road and highways. To make the scenarios realistic, and also safe, test objects representing pedestrians, bicyclists, cars etc are needed. These test objects have to be interpreted as real objects by vehicle sensors and at the same time be “soft” to not damage the vehicle under test (VUT) if a collision occurs. It is foreseen that the number of test objects need to increase in the future and that the objects have to be “connected”, i.e. including V2V and V2X connectivity. The long term goal is up to 100 test objects, where the majority will be virtual, handled by a simulation.

5. Purpose, research questions and method

5.1 Long term purpose

The main purpose is to establish a necessary and unique platform for test and research on automated vehicles and the associated communication at AstaZero. This platform will facilitate development of and research on safe and environmentally friendly vehicles which will benefit and strengthen Swedish automotive industry and academia. At a later stage the platform can assume a similar role with respect to homologation and certification of vehicles and components necessary for the future transportation system.

The long term target for usage of the platform is:

1) test and validation of autonomous functionality and vehicles challenged in potentially dangerous and complex test scenarios with up to 100 objects;
    a. support included for simulation, risk assessment and execution,
    b. control, monitoring and logging of test objects, robots, target carriers and vehicles from different suppliers.

2) test and validation of vehicles and components which are communicating utilizing cellular and/or wifi-based technologies,

3) test and validation of test scenarios where C-ITS is a necessary component, such as intersections without traffic lights,

4) creating a base for research and innovation regarding the future transportation system and society,

5) to be a collaborative arena where automotive, datacom, telecom and academia meet.

5.2 Research questions

The following research questions related to the test platform were identified when the project application was prepared.

1. Identification and description of test scenarios for automated vehicles.
2. Assessment of risk and models for “safe way out” for test objects during test execution.
3. Algorithms and protocols for decision management in real-time, during test execution.
4. Safety benefits from usage of C-ITS (information sharing between vehicles and infrastructure) as a complement to data acquired from vehicle sensors..
5. Quantitative requirements on communication systems for testing in terms of data throughput, latency, availability and reliability.

Chronos part 1 will iteratively approach these research questions together with other ongoing and future activities.

6. Objective

Chronos part 1

The project objective is the first step towards a test and research platform as described above, and the objectives are:

- Investigate coming use cases for communication and the transportation system in order to create a holistic understanding of future test needs,
- Define test scenarios and methods for heavy trucks and cars. Focus shall be test scenarios which will provide appropriate requirements for the test and research platform and its associated architecture and interfaces.
- Define use cases for the platform itself, how the daily operation works and also its relation to existing tool chains. This will provide requirements for the platform.
- Develop the first version of the test and research platform based on the rough architecture presented below. The planned deliveries for the first version and the actual results are presented in section Results and Deliveries.
- Promote standardisation of important interfaces, such as scenario description and control of moveable test objects from different suppliers. This will open for a modular and component based test platform, harmonisation of test systems and reproduction of test results between different test tracks.

To minimize risk the platform will be developed iteratively, using small steps which are possible to demonstrate at AstaZero, which means TRL5. The high level steps are defined as milestones in section Results and Deliveries. In figure 5 below a first draft of the functional architecture for platform is included. During the project the architecture naturally matured and was improved.

One important property of the platform is well defined and open interfaces. These facilitate a modular architecture and where applicable open for a multi-vendor situation. This is especially important for object control (the Obj-ctl interface in Figure 5), and for scenario import/export.

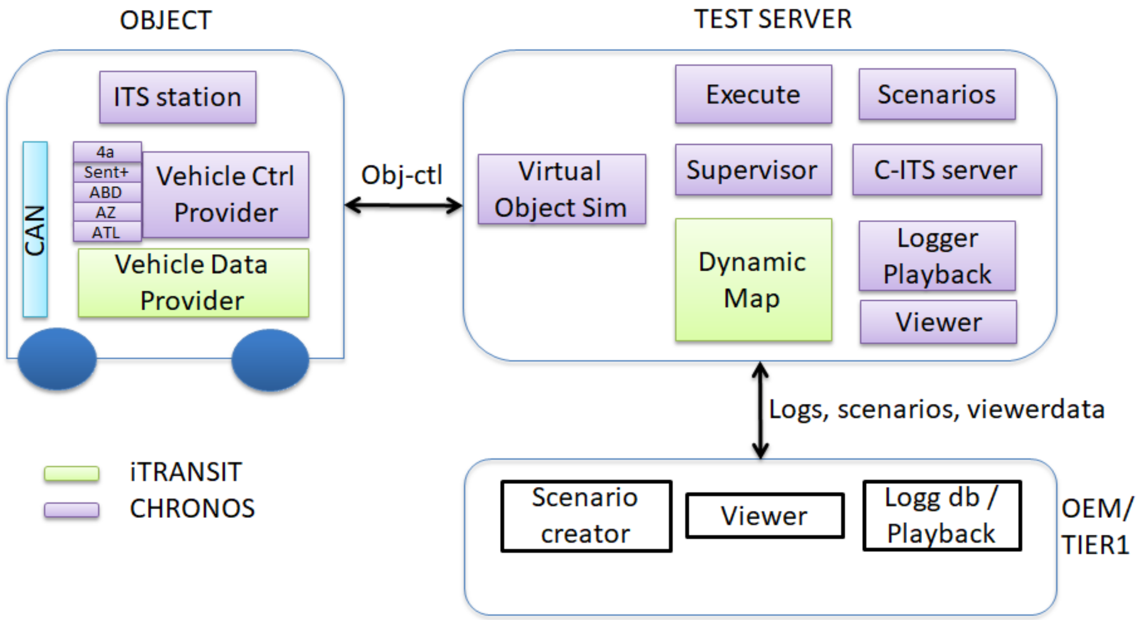


Figure 5 Draft functional architecture, on the system level there are Objects, the cars and targets that are part of a test scenario, and the Test Server. The Test Server controls, monitors and logs test execution, and also interacts with systems on the OEM/TIER1 side.

6.1 Project management, system architecture and dissemination (WP1)

This work package had two main functions; overall planning and reporting, and system architecture. To handle the architecture challenge a dedicated architecture team was established with representatives from HiQ, Chalmers, RISE, Volvo Cars, Ericsson and AstaZero. Individuals involved with the FFI project i-TRANSIT (2015-02330) were included to bring learnings and concepts into Chronos part 1.

This work package also includes integration and demonstrations which were planned as a separate work package in the original application.

6.2 C-ITS use cases for connected and automated vehicles (WP2)

Through workshops with experts from automotive, both commercial and private car area, and telecommunications the goal was to identify and detail use cases relevant a few years from now. The main goal for Chronos 1 was to identify future requirements for the test system and test track, even though the work also serves as a fora for experts from automotive and ICT to meet.

6.3 Scenarios and methods for autonomous vehicles - cars (WP3)

The main objective with this work package was to identify critical scenarios based on real world data which can be used to set requirements on such a steer by server system. This should also suffice to quantify the different dimensions that such a system should be able feature. Developing methods for these scenarios and to quality assess the system is part of this work package as well.

Test Scenarios

Looking at the need for such a steer-by-server system it was identified that such a multi target environment is mostly interesting and needed for Autonomous vehicle verification and validation. Hence already during early phases it was decided to focus on scenarios that are interesting for Highly Automated Driving use cases. Several workshops were conducted both internally at Volvo cars and also together with the project partners to identify the Test Scenarios that are suitable to dimensionalize such a system with the right requirements. This means that the test scenarios were identified such that it is simple enough to achieve within CHRONOS step 1 but also set high level requirements on the ultimate steer-by-server system.

Based on this background 4 Highly Automated Driving car test scenarios were identified and classified in the following categories

- Easy
- Achievable
- Future

Requirements

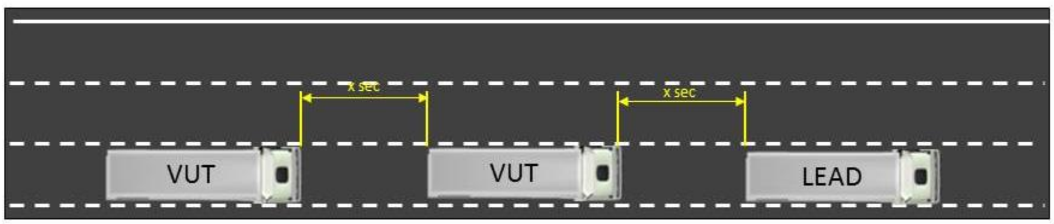
Once the Test Scenarios were decided a need for translating the scenarios into system requirements was identified. To achieve this, workshops were held both internally at VCC and together with the CHRONOS development team. Here the scenarios were broken down into the following system requirements.

- Test System & Server Function Requirements (e.g Drive File, Logging, Synchronization, LDM, GDM)
- Tool Requirements (e.g Logging, Positioning, Communication, Maps)
- Target requirements (e.g Crashability, Type, Speed, Accuracy)

The results are discussed in section 7.3

6.4 Scenarios and methods for autonomous vehicles - trucks (WP4)

This work package has similar objectives as WP3. Trucks and cars mostly operate in the same environment share the same scenarios. The focus have been on developing test methods for supervising and monitoring the AD vehicles to make the verification more efficient and less resource demanding.
The Scenario chosen have been within the Platooning function where two or more trucks are wirelessly connected and the VUT is the following vehicle that is following the lead truck. The platooning function is resource demanding, since a driver is needed in every vehicle. If a robot could supervise the VUTs instead of a real driver, then only one driver is needed to drive the lead truck.



Workshops together with the project partners have been performed as outlined in previous section 6.3, where scenarios, requirements and test system have been discussed. Also requirements related to communication between trucks in platooning case were discussed and included in WP2.

6.5 Test platform - server part (WP5)

Server Core

The server has multiple roles to handle. A primary function is to handle test execution, overall configuration, start, stop, abort, and test logging. All these functions were planned for the first version of the server. Further, to facilitate development it was planned to develop an object simulator, a virtual object that can be used to verify new server support.

Supervisor
A long term goal of Chronos is to be able to perform and repeat test-scenarios safely. A supervisor unit should perform real-time monitoring of the ongoing test and detect if dangerous situations arise. Danger may for example arise due to a communication or positioning problem in one or more of the objects involved, making it divert from its intended trajectory in position or time. When danger is detected by the supervisor, it should safely abort the test-scenario. In Chronos 1 a first step was planned.

Berge-visualizer
The planned result for the visualization engines is to collect all object data from the server, parse it into the engine and visualize the objects and data in a realistic 3D

representation of the AstaZero Testtrack. The engine visualizes real and virtual test objects both in real-time and logged files saved form the server.

The following objects are going to be able to visualize in the first version of Chronos.

- Car
- Truck
- Rigid Truck
- Balloon Vehicle
- Pedestrian
- Elk

Similar to the planned future result for Chronos the visualization engine is going to be able to handle up to 100 test objects in complex test scenarios. The environment build the engine is going to be rendered in real-time which let the user navigate around the test track like a drone or follow specific test objects. Server communication is going to be a UDP connection between the Chronos server and visualization engine. Each object will send following data package which is used to visualize the objects in engine.

<ID>; <ObjectType>; <Timestamp>; <Latitude>; <Longitude>; <Altitude>; <Heading>; <DriveDirection>
Format for all data is stated by the ETSI TS 102 894-2 V1.2.1 (2014-09) documentation.

Simulation Environment

Drive File Creation

The scenarios executed by the Chronos Server are based on Drive Files that contains information about position, velocity etc. at each time step of the objects in the scenario. Having a high resolution (small time steps) in the files introduces the need of a tool that helps to create feasible Drive Files for all objects. The Drive Files also needs to be synchronized between all objects in the scenario. To make the Drive File creation as easy as possible we wanted to use a commercial simulation environment (dSPACE ASM). The simulation environment is able to simulate traffic scenarios with detailed vehicle dynamics. Implementing a logging functionality for the simulation should lead to the possibility to produce realistic Drive Files. Another benefit is also that the simulation environment provides a good GUI for creating traffic scenarios.

Assisting Server Development

In Chronos 1 we also wanted to show that the simulation environment provides several other benefits, such as acting as a development environment for the Chronos Server. This could be achieved by letting the simulation simulate traffic objects and the AstaZero test track.

6.6 4G Network Characteristics (WP6)

The 4G network at AstaZero shall handle the communication between the objects in the test scenario and the Chronos Server. Two important requirements on the 4G network are full radio coverage of the test track and optimal signaling latency between the Chronos Server and the objects on the test track. The objective has been to analyze the network characteristics and data traffic of the existing 4G network and if needed give recommendations to update or re-configure the network to better meet the requirements on an end-2-end test system.

6.7 Testplatform object part (WP7)

Sentient Test Vehicle

During this work package Sentient had the intention to implement a robot for acceleration and braking into their Toyota Auris test vehicle. The robot was the to be connected to the vehicles controller in order to brake and accelerate the vehicle in accordance with the drive files. The power steering was as well to be controlled in accordance with the drive file. In order to be able to operate in a safe manner, an emergency function based on the commands from the server should be implemented. The vehicle should continuously communicate with the server through the agreed protocol.

RISE Self-Driving Vehicle Platform

The RISE Self-Driving Vehicle Platform (SDVP) RC-car will be extended with capability to interface with the Chronos system and act as an object in test-scenarios. The comparably small and lightweight but strong and fast RC-car will become a versatile actor able to take on roles of pedestrians and bikes or as stand-in for full size vehicles while posing smaller danger of seriously damaging other objects or personnel.

Autoliv GPS Robot

The Autoliv GPS robot was continuously updated to be compatible with the communication interfaces developed during chronos. Different functions, such as adaptive sync points, were also prototyped using the robot. The driving robot is vehicle agnostic and was used both in a passenger vehicle and a heavy truck during the project. Some hardware updates was also necessary to cope with the increased work tasks connected to implementing Chronos.

High Speed Platform

The HSP (High Speed Platform), developed by Autoliv under the A-TEAM project, was used in chronos to act as a test object. Its control software was adapted to be compatible with the Chronos communication protocol and new functions, such as adaptive sync points, was prototyped using it. Some hardware updates was also necessary to cope with the increased work tasks connected to implementing Chronos.

Light gate
During the project it was decided that a device was needed to start a test externally while still using chronos, i.e. start test based on a test object not connected to chronos e.g. a vehicle under test driven by a human. Therefore a Light gate trigger was designed and developed. The Light gate communicates using chronos and allows sending an external trigger signal to the Chronos server which in turn sends start message to all connected test objects.

7. Results and deliverables

7.1 Project management, system architecture and dissemination (WP1)

At project start a few changes were made to the project organisation to minimize overhead. WP8 was merged with WP1, and WP5 and WP7 worked closely together.

The architecture team produced a complete high level architecture for the platform which included the level of Chronos 1, a proposed Chronos 2 and future steps in functionality. Please see Annex 1 for details.

7.1.1 Project milestones

The project worked towards and step by step passed new milestones according to the original milestone plan in the project application. All planned milestones were successful.

	Description	Result
M1.1	Robust control of Autoliv robot and platform over 4G in large area (more than 1 km).	OK Q1-2017
M1.2	Architecture in place and tested for test control and communication data	Integrated with arch team
M2.1	Architecture set for test platform. First thoughts about Chronos part 2	OK
M2.2	Autoliv robot and platform drive along whole rural road at AstaZero	OK

M3.1	Monitoring and emergency stop of test vehicle using robot. Keep truck in platoon on the road.	OK
M3.2	Five RC-cars and Sentient car in one test. See link https://youtu.be/hu_4WUCUutA	OK using different objects.
M3.3	Requirement for Chronos part 2 improved.	OK, input to new FFI application
M4.1	Simulation running 100 virtual objects	OK
M5.1	Test run with 5 test objects, an intersection scenario	OK - final demo

7.1.2 Final Test Event

The final test event took place week 8, with preparations during week 6, 2018. The test scenario is depicted in the figure below. The goal of the scenario was to demonstrate the capability of Chronos in terms of number of test objects and scenario execution. The scenario included 7 different objects.

Object	**Type**	**Behavior**
Vehicle Under Test	Not controlled	Depending on test subject driver
Sentient Car	Dynamic	Making left turn in front of HSP, then avoiding pedestrian by steering.
ALV Robot	Dynamic	Driving straight, same direction as VUT
Light Gate	Static	Sending start signal to Chronos when VUT passes
Fengco HIL Rig	Simulation	Approaching intersection from right
ALV HSP	Dynamic	Driving straight, opposite direction of VUT
RISE SDVP	Dynamic	Entering into path of Sentient car and VUT

The scenario setup could be seen as two safety critical traffic events: vehicles making left turn during oncoming traffic and vehicle avoiding pedestrian entering the road. Prior to performing the final test a risk analysis was performed to identify possible risks and plan for countermeasures. 45 risks were identified and divided into three categories: human error, technical problems and causality. Countermeasures were planned and people were assigned to prepare and check that the countermeasures were performed both before and during the test event. The test event was executed successfully with Chronos server being in control of all objects.

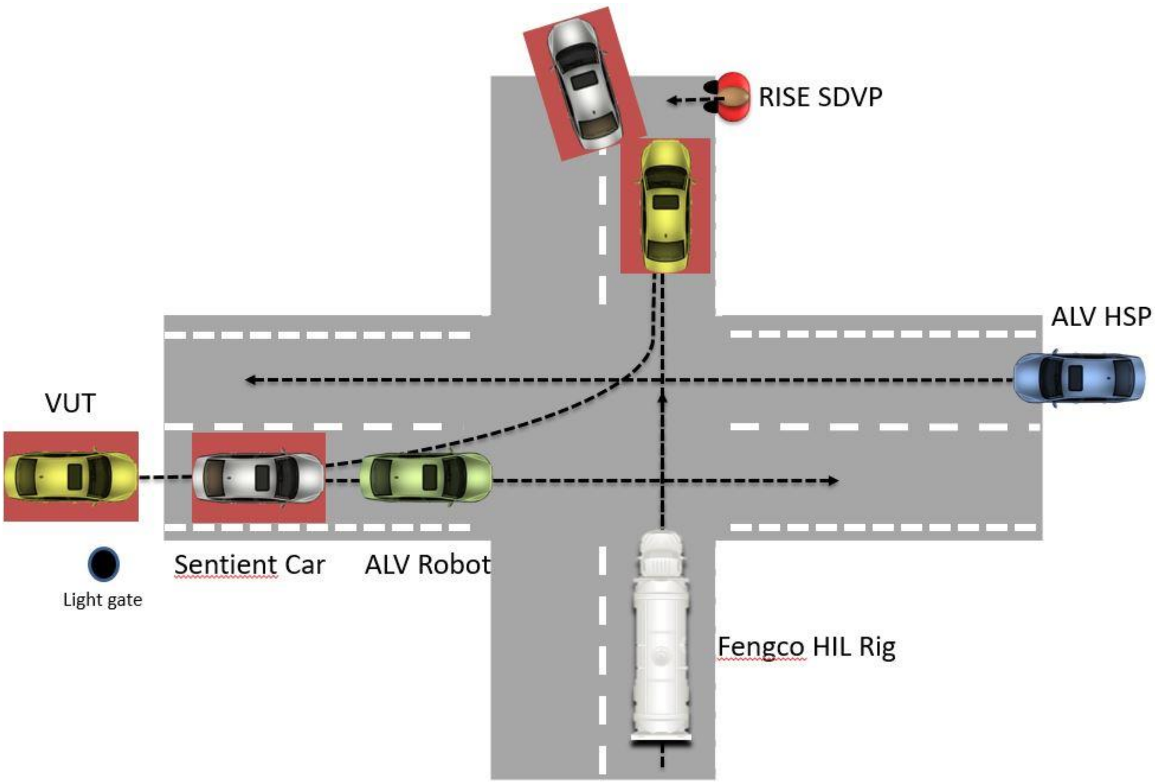


Figure 7.1 schematic view of the final demo setup.

7.2 C-ITS use cases for connected and automated vehicles (WP2)

During Chronos 1 a series of workshops was organized during the fall of 2016 with representatives from Trafikverket, RISE, Ericsson, Chalmers University of Technology, Lund University, Autoliv, AB Volvo, Volvo Cars Corporation and AstaZero with the purpose of capturing future use cases. All workshop output and participant list can be found in [WP2 WS], and a summary can be found in the Annex 2.

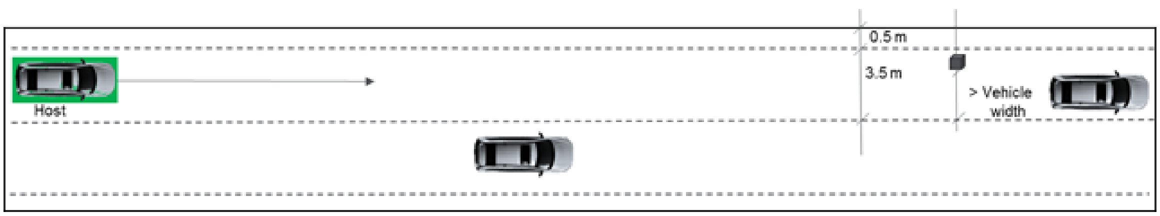
Main output concerning future support required at AstaZero and from the test platform is:

- support is required for Vehicle to Infrastructure information via both 4G/5G and 802.11p for traffic information, such as pedestrians at crossing or merge support into highway, and for traffic lights, etc.
- capability to disturb communication, simulate poor coverage, etc
- new radio technology such as 5G with MIMO and multiple country networks.

7.3 Scenarios and methods for autonomous vehicles - cars (WP3)

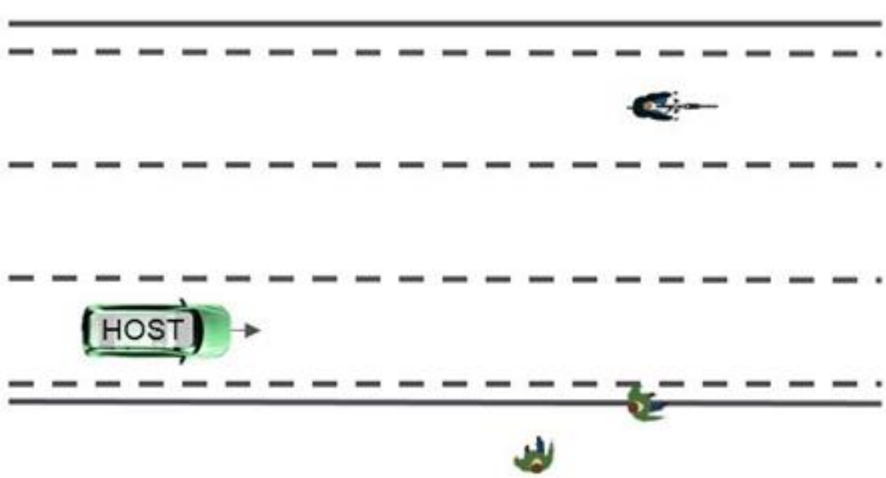
Test Scenarios

Test Scenario 1: *Driving towards a stationary target in a highway traffic environment (Class - Easy)*



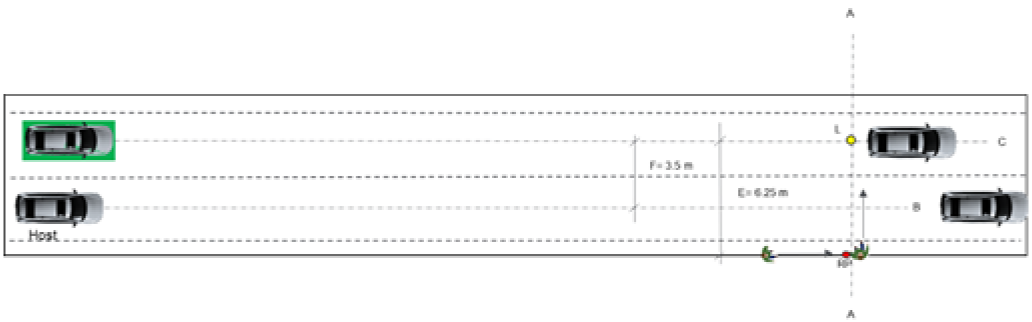
In this scenario a small or large stationary object is partly in host lane. It is sized and positioned in such a way that it is possible for the host to pass within the lane or by using the verge.
There is surrounding traffic influencing the decision making of the HAD car. There is a car in front of the host car that maneuvers for the stationary target quite late which leaves the host with little time to process the stationary target. There is another car in the adjoining lane which restricts the maneuvering of the host car.

Test Scenario 2: *Multiple VRU in a city environment (Class - Easy)*



In this case the host is subjected in a city traffic environment where there are multiple Vulnerable Road users (VRUs) that are dynamic in the test space. The VRU's in the space are bi-cyclists, pedestrian, child pedestrian etc. Unlike the usual pedestrian scenarios that are in EUNCAP these pedestrians and bicyclists can move freely in the space and also multi-directional.

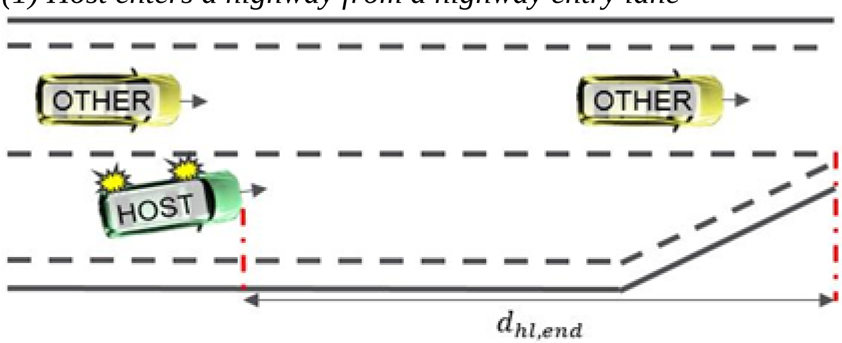
Test Scenario 3: *Driving towards a turning pedestrian in a highway with normal traffic. (Class - Achievable)*



This scenario is similar to Test Scenario 1 with the exception that it is a moving target. The moving target should be a pedestrian which is walking along the road and while the vehicle in front of the host is nearing the pedestrian should start to cross the road.

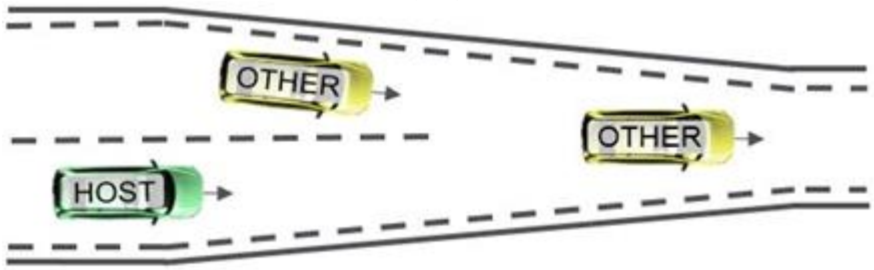
Test Scenario 4: Lane changes (Class: Future)

(1) Host enters a highway from a highway entry lane



In this case the scenario becomes critical when the environment and the traffic together makes the decision making process for the HAD car much complex. As the picture shows the host car is entering a highway while the traffic situation requires the HAD car to find a zip lock pocket and initiate a lane change. The different use case in this scenario would be when the targets in the adjacent lane varies the gap between them. The change in the environment and incorporating it in a test track with map info makes it a difficult dimension for the test control system.

(2) Host merges into a lane merging road.

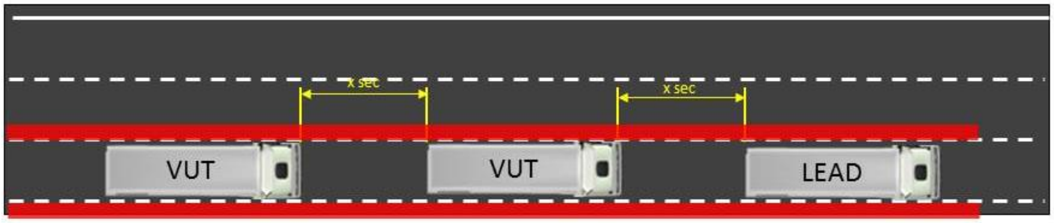


Similar to the above scenario this scenario requires similar decision making capabilities for a HAD car. By varying the position of the target vehicles and by updating the environment in the test system the complexity of the scenario increases.

7.4 Scenarios and methods for autonomous vehicles - trucks (WP4)

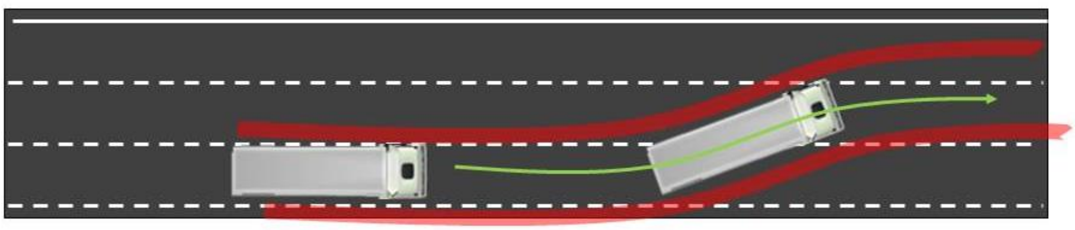
The goal for WP4 has been to investigate how to supervise and monitor the behavior of the AD function in order to make Verification and Validation more efficient and safe. The function chosen for this activity have been the Automated Platooning function of trucks. Platooning is a function where one or more trucks are following a lead truck fully autonomous. The communication between the vehicles is based on V2V communication and the goal is to follow the trajectory of the lead truck as accurate as possible. To be able to verify the platooning function at least two vehicles is needed and there needs to be a safety driver in each vehicle. To make the testing more efficient the idea is to replace safety drivers in all trucks except for the lead truck with driving robots. The driving robot´s task is to monitor the behavior of the function and if the path of the VUT is deviating to much from the lead vehicles trajectory or if the vehicle drives outside the safety margin the robot should take over and steer the truck back to the correct path is possible, otherwise stop the VUT in a safe way.
The test run should be monitored by the supervisor of the Chronos server (more about this in next chapter). If the lead vehicle is controlled by and robot and driven according to a predefined path the supervisor need to monitor that the VUT is following the same path and keeping the correct distance. But if the lead vehicle is controlled by a Manual Driver the challenges of the Supervisor is to monitor that the function in the VUT is behaving correctly since the trajectory of the lead vehicle is dynamic.

7.4.1 Testscenario 1 – Platooning Time Gap (Distance)



Driving straight and keeping the distance. The test scenario aims on testing that the time gap is within limits during acceleration and braking. The lead truck will drive within ego lane all time and the following trucks will drive autonomous with a driving robot supervising the behavior of the function. If the function deviate from the path to much and enters the red area the robot will either steer into the lane or make a safe stop maneuver of the VUT. The limit values of the robot can be recorded and set in the robot.

7.4.2 Testscenario 2– Trajectory following



Keeping the distance and follow the trajectory while changing lane. The test case aims on verifying that the time gap (distance) is within the limit values and that the autonomous following trucks are following the trajectory of the lead truck. If the function in the following truck is deviating too much and the red area is entered the driving robot should take over and steer the truck according to the trajectory of the lead truck or make a safe stop of the VUT.

7.5 Test platform - server part (WP5)

7.5.1 Server Core Overview

In the figure below the high level architecture of the developed server is presented. One main strategy was to use a modular architecture opening for exchange of components to allow parallel development and a future proof solution.

On a high level the components are described below.

System control: the heart of the system and includes all functions which needs to setup and control the test.

Object control: responsible for everything which is specific for the object and its purpose is to be an abstraction layer to hide the details how an object is controlled. It also monitors data from objects.

Supervisor is responsible for monitoring the objects and the complete test. If something is not acting according to plan, or safety issue occurs, the supervision is responsible to tell all the objects to make an emergency stop. Supervisor also includes System control which is the heart of the system and includes all functions which needs to setup and control the test.

Logger adapter is responsible for logging all things which happen in the test system.

Visualization adapter is responsible to format and send data to defined visualizer.

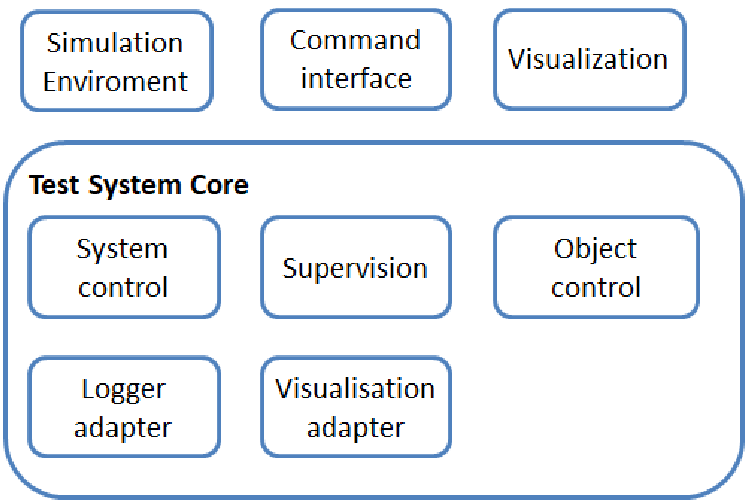


Figure 7.5.1 High level architecture of the developed server.

7.5.2 Supervisor

The purpose of the supervisor is to monitor an ongoing test scenario and to detect problems. A problem is defined as a deviation in position or time of an object from its given trajectory, or if the object has been out of touch with the server for too long. If deviation is severe, the supervisor issues an abort command and ends the test scenario. What is considered a severe deviation is configurable through a supervisor configuration file. The implemented supervisor is a prototype, thus far only used against simulated objects.

While the server is running, the supervisor is listening for a number of the Chronos messages sent between the objects and server. The following list shows the messages that the supervisor responds to in some way:

- **ARM**: This message sets the supervisor in a prepared state. It will start to perform initialisation tasks that require messages from objects, such as an initial alignment of object position to trajectory data.
- **DISARM**: The supervisor is put into an off state and proceeds to reset data structures cluttered by earlier test runs.
- **START**: At this command the supervisor goes into a *started* state where it listens for messages (MON) from objects and correlates these to trajectories. Any unfinished initialisation continues in this state, until finished.
- **STOP**: This takes the supervisor out of the started state and stops correlation of object and trajectory data.
- **MON**: A message from objects containing position, heading and speed amongst other data. This message is used by the supervisor in the correlation of object position against its trajectory.
- **EXIT**: Reception of this message shuts down the supervision process.

The trajectory data that the supervisor uses comes from the same file as the trajectory data that is sent to objects from the server.

The supervisor also sends messages. The most important of the messages sent is abort. Abort is sent when the supervisor detects that an object has deviated to far from the trajectory. When an abort message is sent from the supervisor, the test scenario is terminated and all objects perform their safe exit strategy. The other messages sent are directed at an optionally attached visualiser. These messages provide the visualiser with the currently considered *active* trajectory points and alerts in case of deviation. A visualisation message is also sent in case of abort. The visualiser represents active trajectory points with circles --- the radius of the margin used for maximum deviation --- drawn in green. The green alert circles decay over simulation time. Deviation alerts are illustrated with red dots.

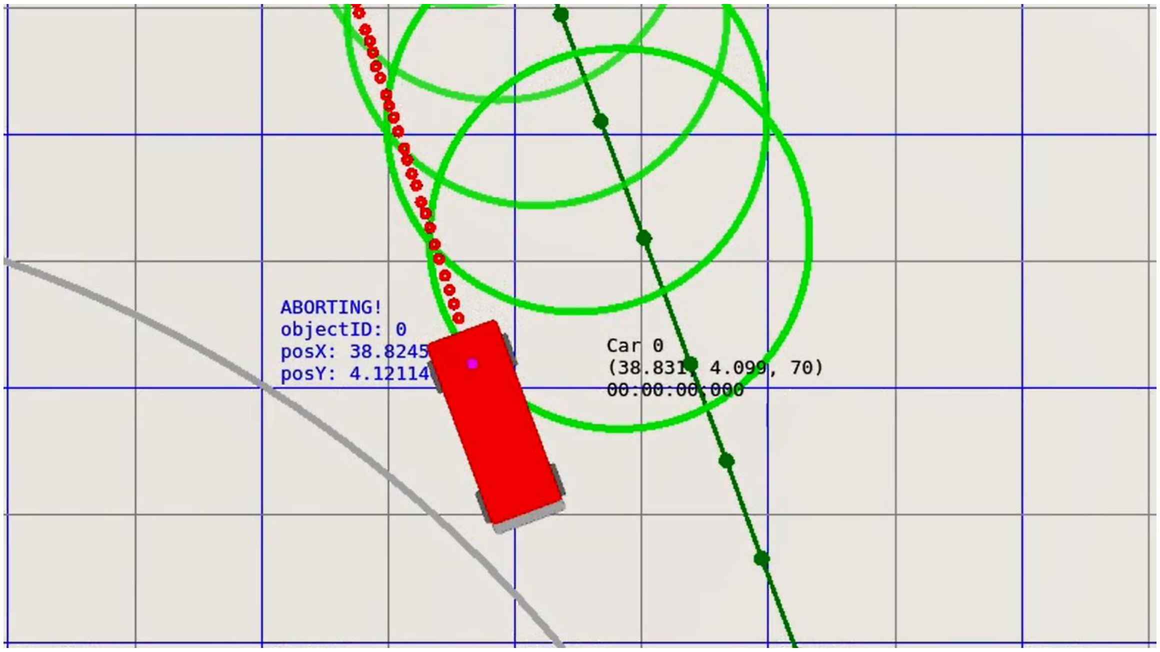


Figure 7.5.2 The visualiser shows the currently considered trajectory point and illustrates the area within which the object must reside (green circle). Alert messages from the supervisor are shown as red points.

The margins used by the supervisor for alerts and to abort are configurable in a configuration file. This allows setting safety margins in position, time and age of last received message from object. Margins for alerts (visualisation) can only be configured for time and position deviation and should be set more narrow than the margins used for abort to serve as early warning of something going awry.

Upon reception of a mon message, the supervisor extracts position and timestamp information from the message and searches forward in the trajectory for the point that is closest in space and time (distance in space and difference in time is treated like a two dimensional vector whose length is the metric used in this search) to where the object is. This point in the trajectory will be considered the active trajectory point. At this time the position and timestamp of the active trajectory point is compared to the values obtained in the mon message.

The implementation of the supervisor is based on the following assumptions:

- When server is armed, all objects will be positioned close to a point on its trajectory.
- Messages (such as arm, start, disarm) follow protocol. For example, an arm message should never be sent while a test scenario is already running.
- The distances between consecutive points along a trajectory are assumed to be smaller than the margins used for aborting a test scenario.

In conclusion, we have developed a proof-of-concept prototype supervisor that works well with simulated objects. The supervisor does however not detect danger, just deviation from trajectory. Future supervisor efforts should be based on a more realistic danger concept that takes into account other objects positions relative to object that is misbehaving. An object's ability to safely catch up with its intended position should also be considered. As the supervisor has only been tested against simulated objects, we have not been able to gauge difficulties specific to real vehicles, AD or controlled by a person (which is also a valid VUT). Future plans also involves more dynamic test scenarios, while the supervisor assumes complete knowledge of the scenario can be obtained through the provided trajectory files.

7.5.3 Berge-Visualizer

The visualization engine resulted in a complete interactive 3D representation of the AstaZero Test Track running and rendered in real-time. As part of the engine a database of the above stated object types where created and represented by a 3D model. See picture below.

The database of objects now consists of 5 objects but the engine is built to be able to easily extended for other and new scenarios. When connected to the server via UDP the stream of data packages spawn and drive the objects in the running scenario, both virtual and real objects. During a test scenario the user can navigate freely around the test track or follow a selected object. The user can monitor all in data based on the information in the data package.

7.5.4 Simulation Environment for Drive File Creation

The implemented logging functionality to the dSPACE ASM-toolchain proved to be a valuable resource to the Chronos project. The GUI and the traffic scheduling in ASM resulted in an easy way to set up a traffic scenario and produce the desired Drive Files. We learned that this was especially useful when a scenario either had a very precise path that an object should follow, or when a scenario included several objects that needed to be synchronized. When Drive Files were requested for an object or scenario, it often resulted in an iterative process with small adjustments to the scenario in each iteration (for example a change in start position or velocity of an object). This allowed the scenarios to be perfected for the desired test. The simulation environment was a good way to handle the iterative process.

Due to the vehicle models in the simulation, the process of creating Drive Files via simulation could verify if a desired scenario was feasible or not. For example, if the desired scenario included an acceleration that a certain object could not achieve, then the vehicle model (if correctly parametrized) would show that the real vehicle will be unable to follow the desired scenario. From this, we learned more about the importance of having correctly parametrized vehicle model for sensitive scenarios.

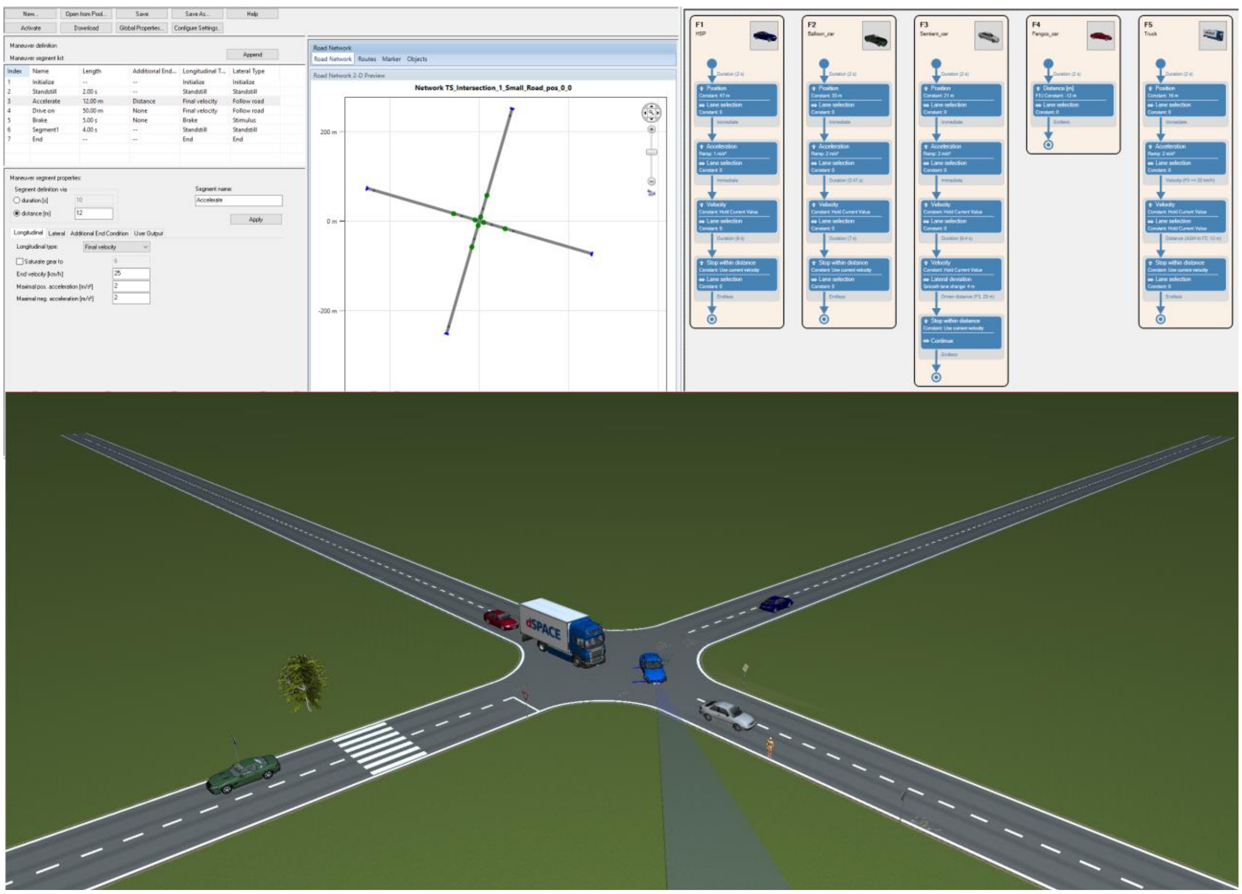


Figure 7.4.3: The GUI for creating a scenario for Drive File-creation together with a simulation of the Final Scenario.

```
Drive File explanation:
TRAJECTORY;<Name>;<Version>;<NbrOfLines>;
LINE;<Time>;<x>;<y>;<z>;<Heading>;<Velocity>;<Acceleration>;<Curvature>;<Mode>;ENDLINE;
LINE;<Time><x>;<y>;<z>;<Heading>;<Velocity>;<Acceleration>;<Curvature>;<Mode>;ENDLINE;
More lines etc…
ENDTRAJECTORY;

Drive File example:
TRAJECTORY;ASTAGarageSquare;0.1;7081;
LINE;0.00;0.000000;0.000000;000.000000;0.000000;0.000000;0.010000;0.000000;0;ENDLINE;
LINE;0.01;0.000000;0.000000;000.000000;0.000000;0.000100;0.020000;0.000000;0;ENDLINE;
LINE;0.02;0.000001;0.000000;000.000000;0.000000;0.000300;0.030000;0.000000;0;ENDLINE;
LINE;0.03;0.000004;0.000000;000.000000;0.000000;0.000600;0.040000;0.000000;0;ENDLINE;
LINE;0.04;0.000010;0.000000;000.000000;0.000000;0.001000;0.050000;0.000000;0;ENDLINE;
LINE;0.05;0.000020;0.000000;000.000000;0.000000;0.001500;0.060000;0.000000;0;ENDLINE;
LINE;0.06;0.000035;0.000000;000.000000;0.000000;0.002100;0.070000;0.000000;0;ENDLINE;
LINE;0.07;0.000056;0.000000;000.000000;0.000000;0.002800;0.080000;0.000000;0;ENDLINE;
LINE;0.08;0.000084;0.000000;000.000000;0.000000;0.003600;0.090000;0.000000;0;ENDLINE;
LINE;0.09;0.000120;0.000000;000.000000;0.000000;0.004500;0.100000;0.000000;0;ENDLINE;
LINE;0.10;0.000165;0.000000;000.000000;0.000000;0.005500;0.110000;0.000000;0;ENDLINE;
LINE;0.11;0.000220;0.000000;000.000000;0.000000;0.006600;0.120000;0.000000;0;ENDLINE;
LINE;0.12;0.000286;0.000000;000.000000;0.000000;0.007800;0.130000;0.000000;0;ENDLINE;
LINE;0.13;0.000364;0.000000;000.000000;0.000000;0.009100;0.140000;0.000000;0;ENDLINE;
LINE;0.14;0.000455;0.000000;000.000000;0.000000;0.010500;0.150000;0.000000;0;ENDLINE;
ENDTRAJECTORY;
```

Figure 7.5.4: Example and explanation of the contents of the Drive Files that was exported from the Simulation Environment.

7.6 4G Network Characteristics (WP6)

7.6.1 General

The 4G network at AstaZero is used for communication between the Chronos Server and the objects in the test scenarios, but it is also used for traffic management at the test track. The focus for the WP6 was to analyze if and how the 4G cellular network affected the end-2-end test system performance.
The two areas of interest were the radio coverage at the test track and the effect of the test network on the test platforms data communication. This communication consists of heartbeat and monitoring signals with a frequency of 100 Hz, which result in a unique traffic model in the 4G network, that in turn puts requirements on low latency and jitter.
Four measurements were performed to find out the characteristics of the 4G network:

1. Coverage measurements at AstaZero test track (see clause 7.6.3.1)

2. Lab measurements related to the Heartbeat/Monitoring traffic model
   This was done to verify test methods and get and reference values of latency and jitter. Not further discussed in this report.
3. Latency measurements at AstaZero (see clause 7.6.3.2)
4. Latency measurements with additional network load and larger payload at AstaZero (see clause 7.6.3.2)

7.6.2 AstaZero 4G Network

The 4G network at AstaZero is an isolated cellular network where AstaZero is acting as an operator. The 4G network is deployed with one radio base station (eNodeB) serving 5 radio cells distributed on 3 radio towers. The 4G network also includes an Evolved Packet Core (EPC) which handles authentication and authorization of the users attaching to the network, mobility, and session control.

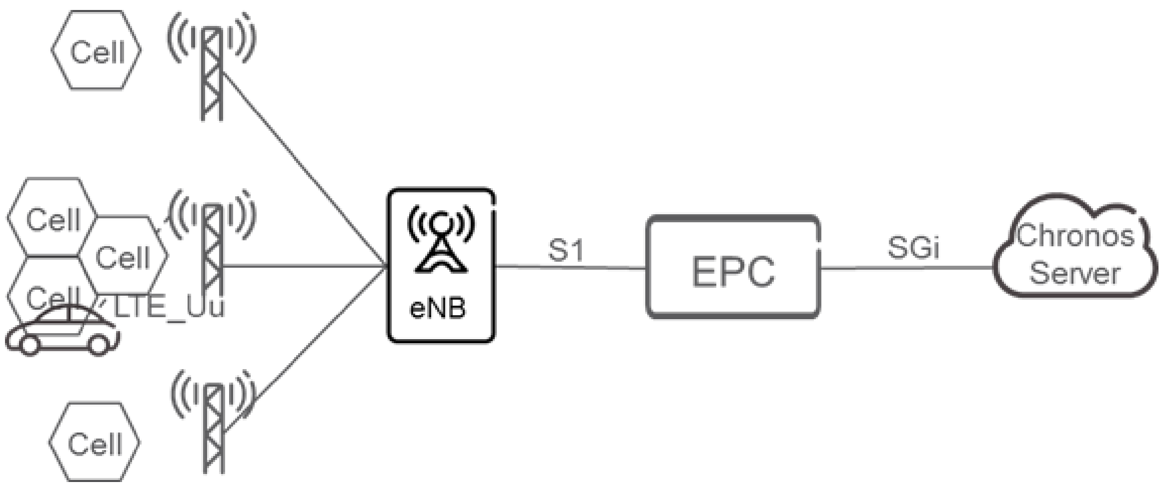


Figure 7.6.2-1: High level 4G architecture at AstaZero

7.6.2.1 Traffic Model

To set the requirements on the cellular network at AstaZero it is important to define the traffic model used between the Chronos server and the objects. Based on the traffic model used and analyses of measurements on the existing system it would be possible to optimize the cellular system for the specific traffic model. In figure 7.6.2-2 the test system signaling is shown for the test preparation and test execution..

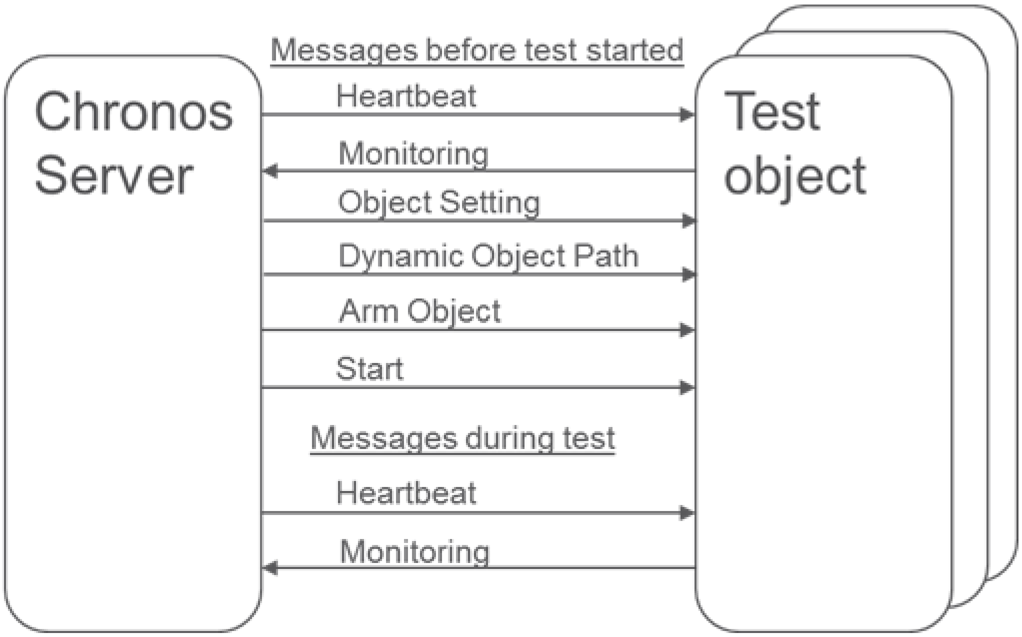


Figure 7.6.2-2: Messages between the Server and Dynamic Object over the cellular network

From a safety perspective the Heartbeat and Monitoring messages are critical, especially while the test is being executed and objects are in motion. These messages are sent with a frequency of 100 Hz and have been the focus of the latency analysis.

7.6.3 Measurements

7.6.3.1 Coverage Measurements

The coverage measurements were done using an Android test tool app installed on a Smartphone that continuously downloads objects from a WEB server in the AstaZero network, see figure 7.6.3-1.

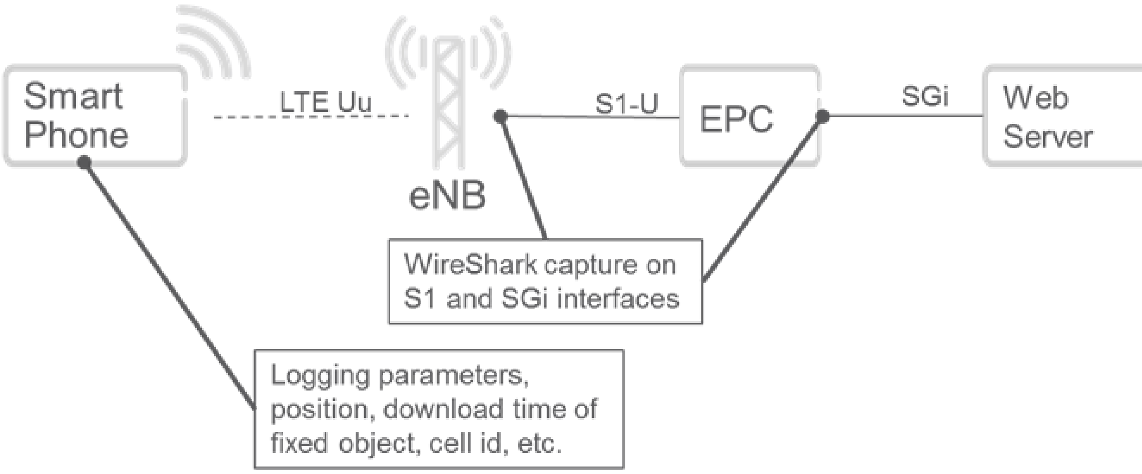


Figure 7.6.3.1: Coverage measurement set-up

The measurements were performed from a car driving around the different areas at the AstaZero track.

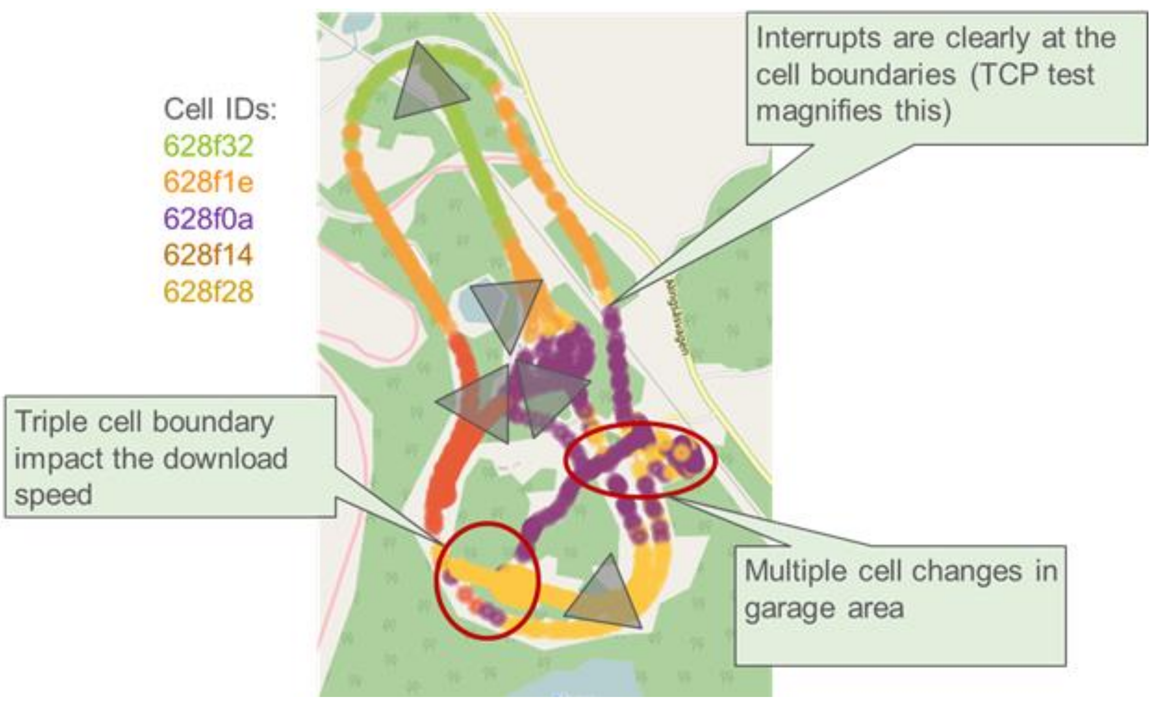


Figure 7.6.3-2: Cell overview & tower positions

In figure 7.6.3-2 the cell tower and cell positions are estimated based on info received from AstaZero combined with measurement observation. All five cells in the AstaZero network were identified as can be seen in the figure. From the figure it can also be seen that the cell changes reduce the number of downloads (fewer dots). It can also be seen that at some places on the track there are multiple consecutive cell changes in a small area, impacting the downloads.

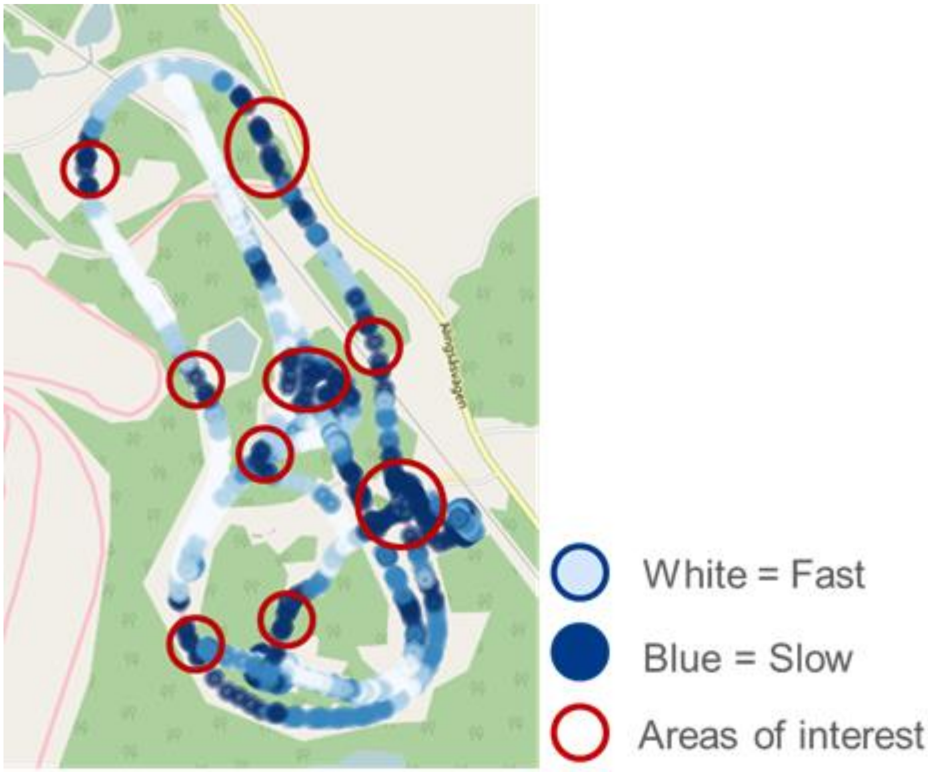


Figure 7.6.3-3: Download speed

In figure 7.6.3-3 download speeds, which vary from 3 – 20 Mbit/sec, are shown on the different parts of the track. Note that this maximum speed can be device specific. As can be seen in figure 7.6.3-3, areas with slow download speed (dark blue) are at cell boundaries, which exception of the high-speed area. The great variations in the download speed in the high-speed area cannot be accredited to cell change alone. An additional contributing factor can be the short distance to the cell tower and the elevation of the antennas not being optimized which may result in poor link quality between or/and high interference from neighboring cells.

7.6.3.2 Heartbeat/Monitoring Measurements

The “Heartbeat tests” were performed using a Raspberry PI connected to a 4G modem sending 100Hz messages (message size 55 bytes) to a server in the AstaZero network which echoed the messages back to the Raspberry PI, see figure 7.6.3-4. The uplink and downlink messages simulate the Heartbeat and Monitoring signals used in the Chronos Test platform.

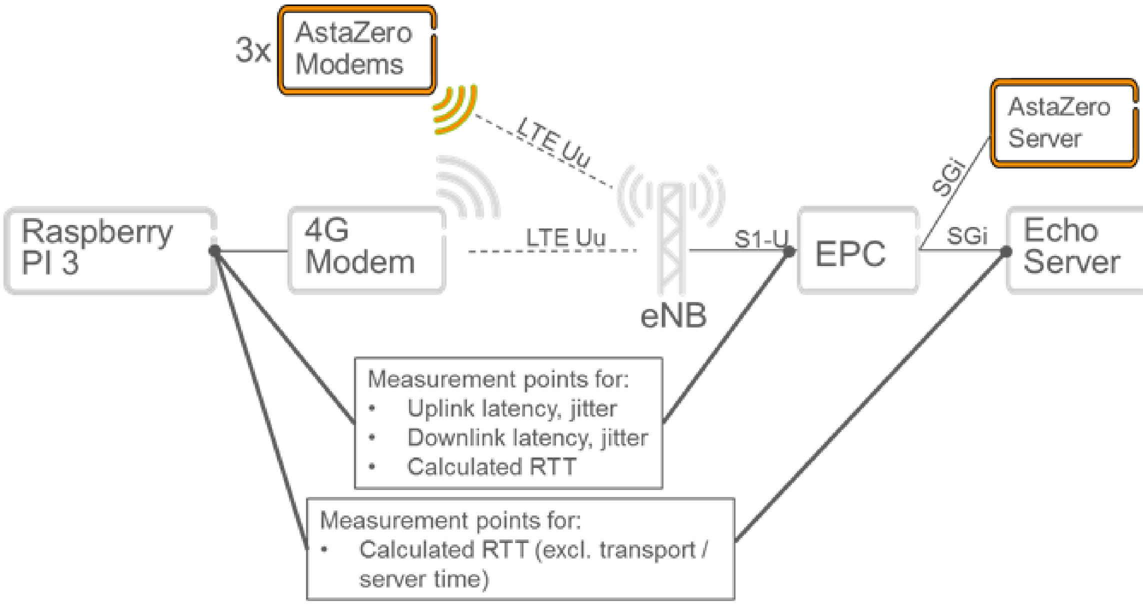


Figure 7.6.3-4: Test set-up for heartbeat measurements

Additional measurements were also performed to see effects of multiple objects with heartbeat/monitoring signaling and the effects of background file transfer. In these tests three additional AstaZero modems were used as also can be seen in figure 7.6.3-4. The latency was measured as the Round Trip Time (RTT) from the message was sent from the Raspberry PI to the "Echo server" and when it was received back in the Raspberry PI.

The measurement results can be summarized as follows:

A. RTT measurements without any additional load
   a. RTT 10 ms to 80 ms and 90th percentile below 25ms
   b. RTT median (50th percentile) 14ms
   c. Jitter -10 ms to 80 ms and 90th percentile below 25 ms
   d. 1 packet lost out of 17783 packets, but this can also be due to start/stop of tests.
B. RTT measurements with larger payload (150 bytes instead of 55 bytes)
   a. Additional 4ms is added to the RTT at the 50th percentile due to the payload being larger than the default initial configured payload.
C. RTT measurements with additional heartbeat load
   a. No additional RTT or jitter compared to result without additional heartbeat load
D. RTT measurements with background file upload/download
   a. 2ms to 4ms RTT degradation at 60th percentile with background download
   b. 4ms to 6ms RTT degradation at 60th percentile with background upload
   c. 90% RTT measurements with background download below 23ms
   d. 90% RTT measurements with background upload below 27ms

Above measurements was performed from the AstaZero office building, the same measurements as in above bullet A. was also performed in other positions along the test track with results as shown in table 7.6.3-1.

measurementSet	varName	samples	min	max	mean	median	perc10	perc90
Track 1	rtt	3059	12.19988	33.60009	17.11550	16.79993	14.40001	19.50002
Track 2	rtt	4166	12.30001	34.49988	16.78752	16.70003	14.40001	19.09995
Track 3	rtt	4631	12.19988	27.80008	16.82937	16.79993	14.49990	19.09995
Track 4	rtt	7647	12.19988	118.00003	17.58381	16.90006	14.49990	20.99991
Track 5	rtt	8526	12.19988	30.20000	16.84737	16.70003	14.40001	19.10019
Track 6	rtt	7092	12.19988	44.49987	16.79304	16.70003	14.40001	19.09995
Track 7	rtt	11939	12.19988	36.09991	16.82722	16.69979	14.40001	19.10019

Table 7.6.3-1: RTT measurements at different positions at the test track.

The following can be summarized from RTT measurements at different positions at the test track:

A. Track location 4 (High speed area) is less good, but still acceptable compared to the rest
B. Track locations 2,3,5,6,7
    a. 96% of measurements below 20ms
    b. 10%ile 14.4ms, median 16.7ms, 90%ile 19ms
C. Track location 4
    a. 85% of measurements below 20ms
    b. Jitter of Track 4 is worst (High speed area)
D. Depending on required latency one can state:
    a. 20ms requirement results in 85% success rate (15% potential too late)
    b. 30ms requirement results in 99.5% success rate (0.5% potential too late)

    (Note this may be higher due to the 3ms passthrough measurement error)

7.6.4 Conclusions and Recommendations

To reduce the number of delayed packets for the Heartbeat/Monitoring signalling a solution is to prioritize and discard buffered/delayed packets using specific handling in the network. Such handling is achieved by assigning a dedicated bearer with **high priority QCI and RLC in non ack mode** in the radio. To reduce uplink latency even further, **1ms uplink pre-scheduling** can also be configured.

To reduce the effects on priority file download (drive files) the recommendation is to assign a dedicated bearer handling this file download with **high priority QCI with RLC in ack mode** (default).

Log file upload should be handled on a bearer with **low priority QCI with RLC in ack mode** (default).

Findings from the above measurements and analyzes can be mapped to commercial use in e.g. cooperative ITS:

- The heartbeat/monitoring can be compared to the CAM/DENM messages specified in ETSI ITS and could be handled in a commercial network same as recommended above.
- The file download (drive file) can be compared to OEM service high priority messaging and updates and could be handled in a commercial network in the same manner as recommended for file download above.
- The file upload (e.g. log files) can e.g. be compared to low priority telematics upload and handled using low priority QCI.

7.7 Testplatform object part (WP7)

7.7.1 Autoliv High Speed Platform

High Speed Platform is the name of the test equipment developed by Autoliv during the FFI A-TEAM project. Its purpose is to carry a 3D vehicle target to represent a real vehicle in a physical test environment. The HSP is 2.5m long, 1.8m wide, approximately 90mm tall and weighs 260 kg; and it can handle being overrun by a heavy truck. The top speed of the HSP is currently 50km/h and maximum acceleration is $1.5 m/s^2$. It has an onboard computer and it can follow a pre-programmed path based on on-board RTK-GPS system.


Figure 7.7.1 depicting the HSP.

During Chronos the software of the HSP has been changed to correspond to the interfaces and protocols defined in the project. Some hardware changes has been made to cope with the software changes. Finally, the powertrain of the HSP has been changed which shows great improvement in reliability of the equipment and ultimately allowed the HSP to be part of the final test event.

7.7.2 Autoliv GPS Robot

The GPS Robot has been used quite extensively in Chronos to evaluate protocols and functions. The developed functions has been evaluated using a Volvo S60 owned by Autoliv but during one test event the robot was mounted in a heavy truck from AB Volvo to test the robot’s ability to prevent run-of-road during testing without test driver intervention. This is one part of one function of the robot which has been developed; i.e. the ability to have the robot monitor the position and speed of the test vehicle and if either deviates from a set threshold the robot takes over the driving. This function could be used to test the behavior of AD vehicles in the near future.

7.7.3 Light Gate

The light gate was used during the final test event to send a start signal to the Chronos server when the vehicle under test passed it. It communicates using the protocol and interfaces defined during the project: it receives DOPM from Chronos and then sends MONR. When the light gate signal gets closed (by a passing reflector) a trig signal is sent to the Chronos server which in turn sends out start signal to all connected test objects.

7.7.4 RISE Self-Driving Vehicle Platform

The RISE Self-Driving Model Vehicle Platform (SDVP) is an open source autonomous vehicle experiment platform developed and maintained at the Research Institutes of Sweden (RISE). One variant of the SDVP system is the RC-Car based self-driving vehicle, there is also support for quadcopters. The self-driving aspect of the RC-Car is provided by an auto-pilot mode that can follow pre-programmed paths accurately outdoors using RTK-GPS.

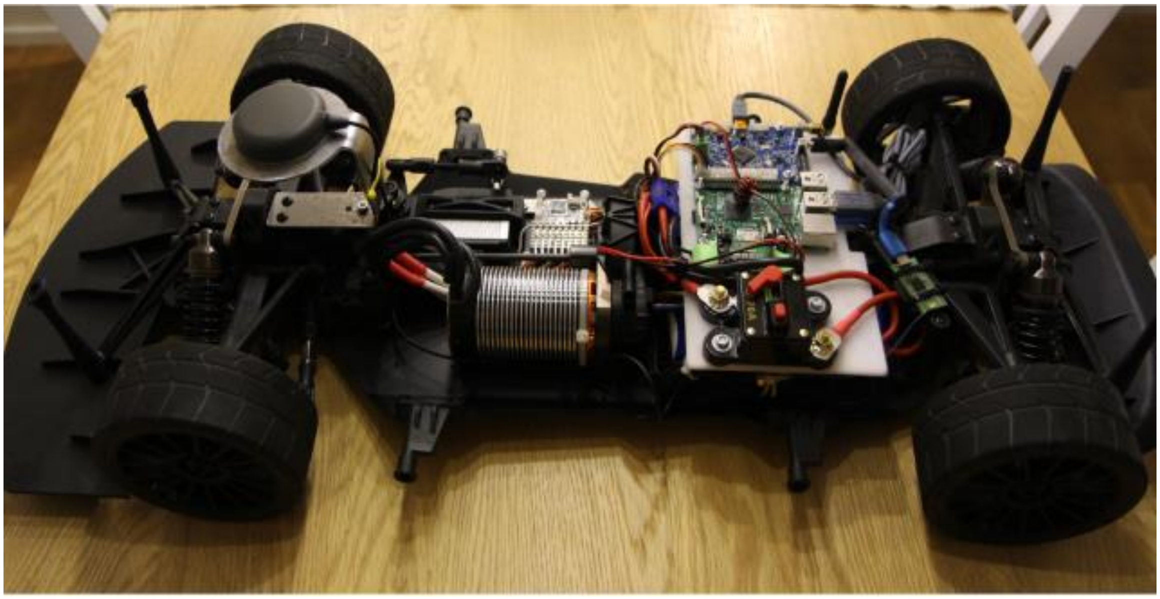


Figure 7.7.2 Placeholder RC-Car

The RC-car has been adapted for use in Chronos as a small and cheap object that can fill many roles, such as:

- Act as a pedestrian, cyclist or other comparable object.
- Stand in for large vehicle in test setup phase.

As the RC-car is small and relatively light it poses less of a danger to seriously damaging full scale objects, infrastructure or personnel involved in or around the test scenario area.

Figure 7.7.3 The RC-car playing the role of a pedestrian.

The RC-car positioning system is based on RTK-GPS but also employs odometry, accelerometer and gyroscope in its positioning calculations. It can travel at speeds of less than 1Km/h and can reach a maximum speed of 80Km/h powered by a 5KW (peak power) brushless dc motor. The hardware designs and software are all fully open source and available for download on github (www.github.com/vedderb).

7.7.5 Simulation Environment as an Object

To assist the development of the Chronos Server, a communication module was implemented to the Simulation Environment that allowed the simulation to communicate with the Chronos Server by the same communication as the real objects on the track. In this use case, the Simulation Environment was running on a dSPACE Scalexio real-time platform. The result was that the Chronos Server could communicate with a virtual object in the simulation and run a virtual scenario. This shows that a Simulation Environment like dSPACE ASM could act as a good development environment for the Chronos Server. Running simulated scenarios have several benefits, such as repeatability, determinism, and not having to book a physical test track for testing functionality of the Chronos Server.

In the final tests of Chronos, the Simulation Environment acted as one of the objects in the scenario. In this scenario, the role of the simulated object was to act as a truck approaching the main events of the scenario, braking before it runs over other

participants. This further showed another benefit of the Simulation Environment - namely that simulated objects can take the role of real objects that have face or cause a dangerous situation in the scenario.
The implementation of the software module in the Chronos project expanded our knowledge about the possibilities and capabilities of the Simulation Environment and the hardware on which it executed.

7.7.6 Sentient Test Vehicle

The plan in this project was to use update the test vehicle in order to be able to a communication protocol in accordance with the ISO draft. For that purpose, the interface towards the vehicle was adapted so that previously available trajectories could be pre-processed in order to be drivable by the vehicle. In addition, the output protocol from the vehicle to the server was implemented, so that the server at all times can establish the position of the test vehicle. In case on a detected fault by the server, in terms of significant deviation or any other fault, such as fault of a third party test participant, an abort signal is sent from the server to the vehicle. This abort signal is connected to an HMI in the vehicle and will be responded to by the safety driver.
The lateral control, i.e. the steering control of the vehicle proved to perform as intended with lateral deviations from the path below 10cm. The trajectories provided by the server commanded the vehicle to be at a certain position at a certain point in time. This proved to be not as easy as it sounds, especially due to restrictions in the longitudinal interface. Early in the project an acceleration and brake robot was implemented. However, it was found that the precision in longitudinal position, i.e. the real position as commanded by the server, against the real position as reported by the vehicle had significant mismatch. The main reason was due to an initial error in speed of the vehicle: As soon as the test scenario starts, the vehicle should move in accordance with the scenario, but gear change and acceleration causes a time delay which exhibits itself as an error in position.
For the server this could look like an erroneous behaviour, while the vehicle only is a little bit “late”. This shows the necessity of a adaptation to more planning of scenarios, through either sync points or adaptive drive files. However, there are a lot of scenarios that can be run that do not require such adaptation. The vehicle used was a Toyota Auris with Automated Manual Transmission. The test vehicle was later changed to a BMW 5 series, with conventional automatic transmission, however, this vehicle was not equipped with the acceleration and brake robot. A lesson learned aspect of this could be that an arbitrary test profile not necessarily can be used on an arbitrary test vehicle.
Using the Sentient test vehicle or a similar test vehicle as the “Vehicle under Test” (VUT) is not a problem as test vehicle then is acting as the “Master” in a scenario, and the longitudinal control performance is of minor importance.
The scenario chosen for the final demo was a left turn with an additional late avoidance of a pedestrian, at a vehicle speed of approx. 25kph. Communication with the server and synchronisation with the pedestrian worked as expected. This was the first time a somewhat more dynamic scenario was used, and the test vehicle behaved as intended. The safety driver in the test vehicle afterwards reported that the drive filed felt slightly too artificial in the test vehicle, and that the test driver would have steered the vehicle in a

slightly different way. For future work, it could be included to look into how drive files can be generated that feel more natural in terms of real driver behavior. A possible solution could simply be to use human drivers to record the drive files with the vehicle that is intended for the test, this was, however, not part of the current project, and the drive files worked as intended.

8. Dissemination and publications

8.1 Dissemination

How are the project results planned to be used and disseminated?	Mark with X	Comment
Increase knowledge in the field	X	
Be passed on to other advanced technological development projects	X	
Be passed on to product development projects	X	
Introduced on the market		
Used in investigations / regulatory / licensing / political decisions		

The video from the final demo can be found here: https://youtu.be/hu_4WUCUutA

Experience and concepts developed in Chronos 1 related to monitoring and control of test equipment on test tracks has been continuously disseminated to an ISO working group (TC22/SC33/WG16) with the goal of an international standard for test equipment control.

Two PhD students were partially supported by Chronos. Iosif Salem and Thomas Petig, they both graduated in the beginning of 2018.

During the project a number of masters thesis and candidate projects were performed at Chalmers:

Albin Pålsson and Markus Smedberg, Chalmers Univ. Tech. 2017, “Evaluation of ROS as a development platform for automated guided vehicles”

David Gardtman and Albin Casparsson, Chalmers Univ. Tech. 2017, “Cooperative Positioning of Vehicles with Consensus”

Henning Phan, Chalmers Univ. Tech. 2016, “Towards Wireless Communication with Bounded Delay”

Herman Fransson and Gustav Ehrenborg, Chalmers Univ. Tech. 2016, “Sensor fusion based indoor positioning with iBeacons”

Nils Andrén, Lars Niklasson, Kevin Petersson Hoogendijk, Filip Slottner Seholm, Alicia Gil Martin,Fanny Sandblom, Chalmers Univ. Tech. 2017, “Predictive Control for Autonomous Articulated Vehicles”

Erik Almblad, Alexander Branzell, Carl Hjerpe, Hawre Aziz, Mattias Eriksson, Robert Krook, Chalmers Univ. Tech. 2017, “Evaluering och utveckling av bildbaserade positioneringssystem för självkörande skalade fordon”

8.2 Publications

The following papers and presentations have been produced:

Martin A. Skoglund, Thomas Petig, Benjamin Vedder, Hans Eriksson, Elad Michael Schiller: Static and dynamic performance evaluation of low-cost RTK GPS receivers. IEEE Intelligent Vehicles Symposium, pages 16-19 2016

Olaf Landsiedel, Thomas Petig, Elad Michael Schiller: DecTDMA: A Decentralized-TDMA with Link Quality Estimation for WSNs. 18th International Symposium Stabilization, Safety, and Security of Distributed Systems, SSS 2016, pages 231-247, 2016.

Shlomi Dolev, Chryssis Georgiou, Ioannis Marcoullis, Elad Michael Schiller: Self-stabilizing Reconfiguration. Networked Systems - 5th International Conference, NETYS, pages 51-68, 2017

Vladimir Savic, Elad Michael Schiller, Marina Papatriantafilou: Distributed Algorithm for Collision Avoidance at Road Intersections in the Presence of Communication Failures. IEEE Intelligent Vehicles Symposium, IV 2017.

Marco Canini, Iosif Salem, Liron Schiff, Elad Michael Schiller, Stefan Schmid: A Self-Organizing Distributed and In-Band SDN Control Plane. ICDCS 2017: 2656-2657

Vladimir Savic, Elad Michael Schiller, Marina Papatriantafilou: Aiding Autonomous Vehicles with Fault-tolerant V2V Communication. CoRRabs/1710.00692 (2017)

Marco Canini, Iosif Salem, Liron Schiff, Elad Michael Schiller, Stefan Schmid: Renaissance: Self-Stabilizing Distributed SDN Control Plane. CoRR abs/1712.07697(2017)

Four internship students developed a path creation tool as part of their education. It was a collaboration between École nationale supérieure d'ingénieurs de Caen, Chalmers University of Technology and Autoliv.

9. Conclusions and future research

Collaboration in the project has worked well and all planned milestones were met. The following points cover the main conclusions and input to future research:

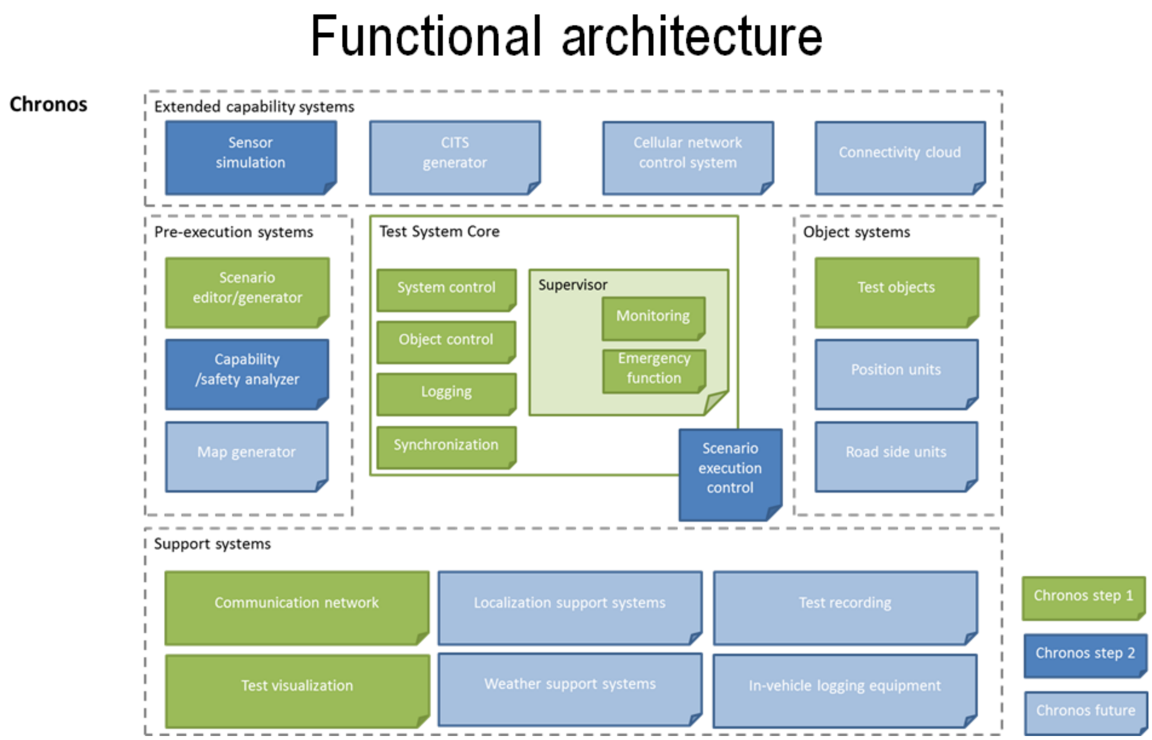
- with a component based architecture and use of well defined interfaces it is possible for many partners to together develop a complex test platform. integration with many types of test objects has been demonstrated, spanning from Sentients cars, to Fengco HIL rig to Autoliv equipment.
- game engine is a powerful tool for visualisation of test scenarios for test planning and result analysis
- design of test scenarios and creation of drive files for tests can be done in a simulator and has potential for future work.
- 4G network at AstaZero can be further enhanced to allow for simultaneous log file download (with lower priority) and test execution (with high priority). Further it is recommended to modify uplink pre-scheduling.
- initially it was planned to re-use ETSIs data formats for V2X communication, e.g. speed, acceleration, position as data formats for communication in the test system. However the accuracy has been proven to be too low. Test track testing requires 1 cm level accuracy for positioning, while ETSI 102 894-2 is limited to approximately 10 cm.
- drive file generation will in the future need many enhancements:
    - speeds and steering behaviour should be more human-like,
    - multiple alternatives at critical points in scenarios to allow for “safe way out”
    - periodic drive file updates, or short-term drive files to allow for dynamic adaption to ongoing test.
- in scenario planning there are two important components which have not been part of Chronos 1 but should be considered
    - awareness of capability envelope for different objects, e.g. pedestrian max speed, car max acceleration etc,
    - awareness of capability envelope of available test equipment when allocation is performed to required objects in the scenario, e.g. some target carrier have lower dynamic abilities than real vehicles, especially in low-my conditions.

10. Participating parties and contact persons

Party	Contact person
AB Volvo	Lars Bjelkeflo
AstaZero	Niklas Lundin/Viktor Johansson

Autoliv	Per Gustafsson
Berge	Jonathan Allvin
Chalmers	Elad Schiller / Erik Ström
Ericsson	Roland Gustafsson
Fengco	Oskar Gellerbrant
HiQ	Anders Bengtsson
RISE	Martin Skoglund
Sentient+	Jochen Pohl
Volvo Cars	Vivetha Natterjee

Annex 1 WP1 architecture team result



- Pre-execution
  - Scenario editor/generator: Tool to edit scenario to be used during test and to generate necessary information for test system core.
  - Capability/safety analyzer: Tool to check and verify capability and/or safety for defined scenario, track and available objects.
  - Map generator: Tool to generate real or fictive maps to be used in objects or vehicle under test.
- Test System Core
  - System control: The heart of the system and includes all functions which needs to be able to control the test.
  - Object control: Responsible to control and communication with each object which is included in the scenario.
  - Logging: Responsible to log all things which occur in the test system as well as monitor data from objects.
  - Synchronization: Responsible for making sure that all system elements have coherent and synchronized view about the system state.
  - Supervisor: Responsible for monitoring the complete test incl objects. If safety issue occur the supervision is also responsible to perform emergency action.
  - Scenario execution control: Responsible to read in a high level scenario and control the test system accordingly.
- Extended capability systems
  - Sensor simulation: Simulation of sensors inside vehicle under test to be able to perform virtual objects injection into vehicle under test.
  - CITS generator: System to generate CITS messages for track objects or simulate traffic according to a defined traffic model.
  - Cellular network control system: System to control the cellular network used by the vehicle under test to for example simulate a roaming.
  - Connectivity cloud: System to enable testing where vehicle under test get or generate information to a connectivity application e.g. Slippery road warning.
- Track objects
  - Test objects: Controllable entities in a test. Objects can be real vehicles and/or target carries to be able to carry crashable dummies.
  - Position units: Device to send position for non-controllable vehicles or other objects to the test system.
  - Road side units: Device located on the roadside that provides support to passing vehicles, either visual and/or via wireless communication.
- Support systems
  - Communication network: Network to support necessary communication for test system and vehicle under test. Examples can be 3G,4G or Wi-Fi.
  - Test visualization: Systems to visualize test before, during and after running a scenario. Can be done in 2d or 3d but also with VR and AR.
  - Localization support systems: System to provide necessary information to test objects to perform localization (e.g. RTK data) or to provide exact position.
  - Weather support systems: System can be to collect weather input or to actually create a wanted lightning and precipitation in a indoor facility.
  - Test recording: External systems to record the test for example a camera synchronized with the test system.
  - In-vehicle logging equipment: Equipment to store all needed information created inside the vehicle.

Annex 2 WP2 output

Typ of communication	Objects involved	Environment	Scenario
MIMO based 5G	Vehicles with antennas	City environment with changeable walls	Test of beam forming in a multipath situation. EMC-testing
4G, 5G or 802.11p	Vehicles capable of sending CAM* messages	Intersection with connected traffic light	1 Red light violation (stop violating vehicle automatically) 2 Priority for emergency vehicles / public transport 3 Green light optimisation (GLOSA*), adapt speed top avoid stop. 4 Simultaneous start at green light for efficiency
4G, 5G or 802.11p	Construction machines, trucks	Closed off area	Autonomous goods handling based on on-board sensors and communication.
4G/5G (via base station)	cars and trucks	Highway	Vehicle to vehicle applications with real-time requirements, i.e. emergency electronic brake light, traffic jam ahead. Change output power, s/n ratio etc
4G/5G (via base station)	cars and trucks	any road	Cloud to cloud information exchange between different OEMs vehicles.

802.11p or 4G-V2X	pedestrians, bikes, cars and trucks	City environment with realistic walls	VRU position announcement via communication, other vehicles avoid if conflict occurs.
802.11p or 4G-V2X	trucks and cars	highway	Merge assistance via communication Highway entrance assisted via communication
802.11p and 4G/5G	any vehicles	large intersection, road or roundabout	Many vehicles, attempts to disrupt communication with radio and microwave transmitters.
	any vehicles	bridge over other road	Vertical positioning while passing bridge with GPS interference.
Relevant standards for transportation	any	Realistic city environment	Poor radio coverage. Congestion. Cloud to cloud, and cloud to vehicle.
4G/5G	trucks and cars	Highway	Many vehicles, multiple data flows. Priority of traffic related communication vs infotainment.s
any	any	any	V2V active safety V2I cloud traffic conditions, roadwork V2I/V2V traffic mgmt V2V low speed control, parking

*CAM Cooperative Awareness Messages
*GLOST Green Light Optimal Speed Advisory