UNIVERSITY OF GOTHENBURG

# Synchronising vehicles during Active Safety Systems testing

## A Design Science Study

*Bachelor of Science Thesis in Software Engineering and Management*

JAMES EAGLE

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
Göteborg, Sweden, June 2016

**Synchronising vehicles during Active Safety Systems testing**

JAMES EAGLE



Supervisor: Alessia Knauss
Examiner: Christian Berger

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden June 2016

# Synchronising vehicles during Active Safety Systems testing

James Eagle
University of Gothenburg
Department of Computer
Science and Engineering
guseagja@student.gu.se

***Abstract*—Proving-ground testing of Active Safety Systems is typically based on scenarios in which relative positioning and speed of all vehicles on the track is strictly defined. Current testing methods are not designed for human driven or autonomous Vehicles Under Test as they rely on the predictability of path-following driving robots to precisely control these parameters. As Active Safety Systems are both increasingly integrated with vehicle autonomy and are further augmenting human driving, these testing methods become less useful. This paper presents an approach for preserving most of the control of current methods while allowing for some level of uncertainty generated by an autonomous Vehicle Under Test. Utilising the master/slave architectural pattern, slaved mock-up vehicles, or Test Targets, are synchronised with the Vehicle Under Test via a centralised Master Server. Using a design science methodology, this approach is described and then evaluated against the existing best practice. It represents a first move towards integrating autonomous vehicles into existing Active Safety Systems testing.**

***Index Terms*—Synchronisation, Active safety systems testing, Master slave architecture, Autonomous driving**

## I. Introduction

Active Safety Systems (ActiveSS), also known as Advanced Driver Assistance Systems (ADAS), are an increasingly important part of general vehicle safety. ActiveSS, such as collision warning or Autonomous Emergency Braking (AEB), are typically thought of as assisting human drivers in potentially dangerous situations by avoiding or mitigating the implications of an accident. Human reflexes and awareness are augmented by quick, digital decision making and sensor arrays.

As automotive automation technology advances, ActiveSS are also supplementing or are integrated into autonomous driving systems. Failure in any component of a vehicle can be fatal and the increasing complexity and autonomy of these systems requires a similarly sophisticated approach to quality and performance testing. This can mean simulation testing, Hardware and/or Human in the Loop testing or physical testing, either on a controlled test track at a proving-ground or in the field in real world conditions. The improvement of controlled track testing is the focus of this paper [23].

Proving-ground testing of ActiveSS typically involves one or more Vehicle/s Under Test (VUT for singular and plural) – the vehicles on which ActiveSS are being tested are installed. There may also be a variety of Test Targets also known as Test Targets – moving or stationary mock-ups of pedestrians, obstacles or other vehicles that are designed to trigger sensor arrays in the same ways as their real world counterparts. They are designed to limit or eliminate the possibility of damage or injury in the case of a collision. A test scenario will define the behaviour and positioning of Test Targets and take into account (or define) the expected behaviour of the VUT.

This multi-agent environment must be able to fulfil exacting test criteria and produce repeatable results, not always an easy task in a physical environment. Usually the scenario is designed to meet a safety awards standard such as the European New Car Assessment Programme (EuroNCAP).

### *A. Problem Domain and Motivation*

Currently, most proving-ground test environments use agents, both VUT and Test Target, that are synchronised by relying on internal real-time clocks and a simultaneous start. They follow predictable, predefined paths in order to ensure that the testing parameters are replicated consistently. The VUT is often controlled by a robot driving rig that fulfils this behaviour requirement.

As the level of ActiveSS driver augmentation as well as Autonomous Driving increases, however, test scenarios will need to find a way to synchronise all agents even with a less predictable VUT. A recent study by A. Knauss et al. on ActiveSS testing trends identified remote control and synchronisation in a multi-agent test scenario to be an important next step [16].

An autonomous VUT (human or computer driven) will not necessarily behave exactly in the way that the test scenario prescribes. The existing environments are not well adapted to handling unpredictability. There needs to be a way of handling control of the Test Targets that preserves the test conditions and repeatability in synch with the VUT. Synchronisation of communication and generated data is also important for analysing the test results.

Synchronisation (or coordination) of automated systems is something that has been studied for some time in the robotics field but there is little material on solving the specific problems of handling autonomous VUTs [11]. Such a system must consistently meet difficult test conditions in an unstructured environment. Synchronisation for ActiveSS testing on a proving-ground track requires a compromise between the fully controlled environment vs. agent freedom-to-adapt dichotomy.

### B. Research Goal and Research Questions

In this paper we investigate how to enable existing ActiveSS testing processes to support Autonomous VUTs, considering industry standard test scenarios and the synchronisation of multiple, controlled Test Targets. Hence, the research question for this paper is formulated as follows:

RQ - In Active Safety Systems Testing, how can path-following Test Targets be synchronised with an autonomous Vehicle Under Test to meet strict, repeatable test criteria?

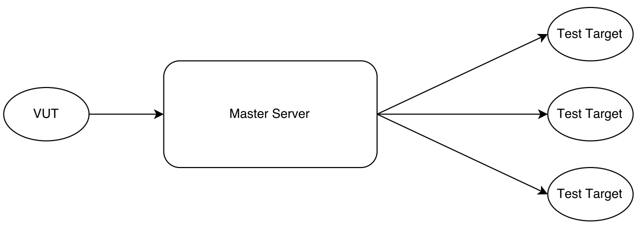


Fig. 1. The control flow of AMPS

### C. Concept and Contribution

This paper proposes a concept called AMPS (Adaptive Master, Path-following Slave) for synchronising Test Targets with the VUT. It is based on a master/slave architecture pattern, similar to the one proposed in R. Portillo-Vélez et al. [21]. AMPS utilises a central server (the Master Server) that tracks autonomous VUTs and controls the Test Targets accordingly. The controlled, path-following behaviour of the Test Targets has been shown to be a good way to achieve reliably successful tests in many conditions [11] [10]. AMPS will preserve this aspect of the current best practice, also allowing for maximum compatibility with existing commercial Test Target platforms.

AMPS will provide a basis for future work on ActiveSS testing. In particular it provides a path toward handling the increasing complexity introduced by increased vehicle autonomy and multi-agent synchronisation.

### D. Evaluation

The evaluation of AMPS will focus on its ability to manage multi-agent synchronisation in ActiveSS testing with the inclusion of an Autonomous VUT. It will be based on qualities and scenarios derived from Autoliv's testing process and the existing best practice identified from literature. The evaluation will include an experimental implementation of one of the key challenges to the feasibility of AMPS, updating paths on-the-fly, based on Test Target development at Autoliv.

### E. Structure and Scope

The background section will look at ActiveSS testing and expand on the idea of existing best practice. There will also be examination of existing work on synchronisation and a brief exploration of path-finding and prediction.

The methodology section will outline the design science methods of construction and evaluation. It will also include a summary of a single embedded case study of current ActiveSS testing and derive criteria for evaluation. The results of the paper are formed by the concept and its subsequent evaluation.

There are some aspects of AMPS that, in implementation, will prove to be complex problems. The path-finding and prediction, as well as clock synchronisation, are examples. These aspects are not explored in depth in the AMPS proposal but some guides to possible solutions are provided in the background.

## II. Background and Related Work

AMPS relies on an examination of related work from a number of different areas, the most important of which is work that helps define the existing best practice. Some areas integral to the success of any implementation of AMPS, such as clock synchronisation and path-finding, are out of scope for AMPS abstract level of design but are briefly examined below for context.

### A. ActiveSS testing: Existing best practice

One way to get a good idea of scenarios and requirements in ActiveSS testing is to look at requirements outlined by the EuroNCAP, and other safety awards organisations and programs such as the US NHTSA's NCAP safety awards [9]. Such organisations define scenarios in which Active Safety Systems must meet strict criteria and are awarded a well-recognised safety rating accordingly. One example, AEB, might require a car braking hard 20 metres ahead of the VUT while it is travelling at 70 km/h [8].

The precision required for these scenarios is the reason Autoliv and other developers of Active Safety Systems use the path-driven, multi-agent system outlined above [12]. Each vehicle uses precision GPS systems, such as differential GPS (dGPS) or Real Time Kinematic GPS (RTK-GPS) that use track-side base stations for additional accuracy, to determine its location on the track. Control loop feedback mechanisms such as PID (proportional–integral–derivative) controllers are used to regulate each vehicle's progress along its predefined path. Each agent is synchronised at the beginning of the test and rely on the accuracy of internal clocks for precision timing.

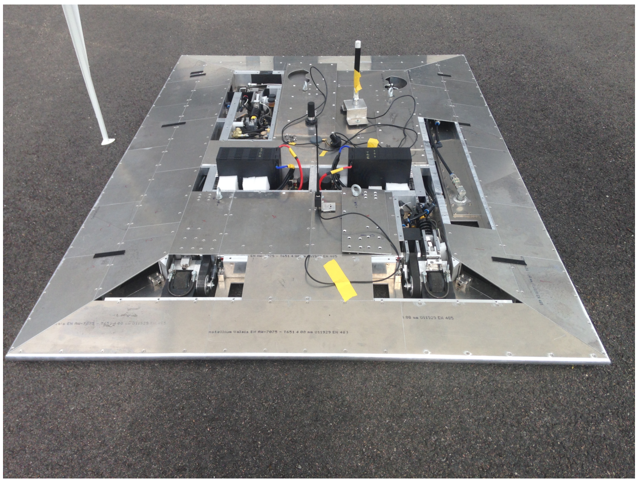

Fig. 2. An Autoliv "low-rider" test target

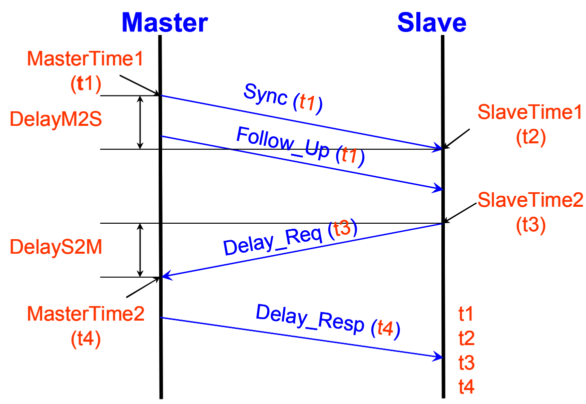


Fig. 3. Delay request-response mechanism from E. Song and K. Lee [25]

Every agent transmits its telemetry to a base station for analysis.

Work by H. Schöner et al. on ActiveSS testing using path-following robot driven Test Targets and VUTs is cited in many publications as a reference for the kind of open-loop testing (a control mechanism not relying on feedback) that is used by Autoliv and others [23]. It defines several common scenarios derived from an analysis of existing ActiveSS testing. Recent work by H. Fredriksson et al. on repeatability in winter testing using path-following robots demonstrates that this system can be suitable even in adverse track conditions with some tweaking [10].

Testing at Autoliv uses this method to achieve results in safety awards and internal test scenarios. They utilise their own "low-rider" Test Target, a sturdy, flattened, low profile driving platform capable of being run-over by light trucks. Radar reflecting foam or "balloon" vehicle mock-ups are mounted atop the platform and it is controlled using a pre-loaded path file. This is a safe, precise and reusable Test Target that allows for flexible scenarios.

Path planning is an integral part of existing practice. The paths of all vehicles must fulfil the scenario within the bounds defined by its parameters, such as relative position and heading. It is important that the physical limitations of the Test Target, such as the top-speed, acceleration and turning circle of the platform, as well as the limitations of the computing hardware and the communication infrastructure are all understood when planning paths.

AMPS builds on this work, taking the concept of path-following Test Targets and adapting the path generation and synchronisation to accommodate an autonomous VUT. Moving to an interactive, centralised system is a significant shift and it is important that the qualities of the existing best practice, as defined above, are maintained.

### *B. Synchronisation*

In order to handle the unpredictable trajectory of a human driver rather than a robot one, some form of synchronisation of the agents in the system is required. The synchronisation (or coordination) of multi-agent systems is something that has been studied for some time in the robotics field [22].

Quite commonly, the focus for this work tends to be either systems with fully or semi autonomous agents that are adaptive [20] or factory-type industrial settings with cooperation and movement strictly controlled [17]. These methods are not ideally suited to the variable nature of the test-track environment combined with precise control needed for ActiveSS testing criteria.

A Master/Slave synchronisation configuration, such as that presented by R. Portillo-Vélez et al. [21], provides a good model for AMPS. The idea of coordinating agents using a master unit that accepts inputs and drives the other agents in the system is adapted for the track. That paper focused on a master robot manipulator and its slaves moving in unstructured non-planar space. AMPS uses instead a Master Server, which serves as a proxy for the behaviour of the autonomous VUT, controlling the Test Targets.

On track synchronisation has been explored in ActiveSS testing by S. Heinlein et al. within a more limited and controlled scope [11]. They propose several ways to synchronise a motorised gantry system that can move a soft pedestrian target across the path of a moving vehicle. One of their three proposals explicitly examines synchronising dummy behaviour with the movement of a turning vehicle, robot driven but with variable speed settings. They found that radio transmission of vehicle position over WLAN could be used to time the movement of the dummy to produce repeatable results. Their system was reliant on accurate timestamping and a good knowledge of network latency. They were able to achieve consistent results even with the highly variable VUT speed and suggest further work is under way to account for lateral vehicle movement. The gantry carried dummies were only capable of moving forward or backward, AMPS goes a step further as it is using Test Targets that must manoeuvre laterally as well as longitudinally.

Achieving accurate, synchronised timing such as that used by S. Heinlein et al. [11] is an important part of the overall synchronisation process. The Precision Time Protocol (PTP), defined in the IEEE 1588-2008 standard [1], is useful for synchronising the clocks of distributed control systems and has sub-millisecond accuracy. It also relies on a Master/Slave style architecture, with a Master clock responsible for the source timing. Figure 3 from E. Song and K. Lee shows how the basic delay request-response mechanism works. It makes sense to use PTP to keep all the clocks in each vehicle slaved to the Master Server for AMPS [25].

### C. Path-finding and prediction

The ActiveSS testing environment is a dynamic, multi-agent problem space and test scenario constraints play a big factor in path planning for Test Targets. It is especially challenging when the paths must update in real-time and the well known Djikstra or A* heuristic-based algorithms may not be fast enough to adapt.

A review of path-finding techniques in robotics and computer games by Z. Algfoor et al. identified a number of techniques suitable for multi-agent and real-time applications that have been recently studied and evaluated in a robotics context [2]. They also found that the study of basic path-finding techniques is a mature field and that ongoing research is extensive, providing many options for optimising the path-finding task of the master server.

Generally the problem is broken down into two steps, representing terrain with graph generation and algorithmically finding the optimal path. Representing the terrain of a test-track needs to be a dynamic process due to the multi-agent environment. It is possible to do this in a number of ways such as visibility graphs, used in the accelerated A* algorithm proposed by D. Šišlák et al. [24], or triangulated representation of free space, used for constraint-aware navigation developed by M. Kapadia et al. [15]. Another option, also navigation focussed, is work by F. Lucas et al. that proposes combining constraint solving and ant colony optimisation algorithms with a waypoint base terrain graph [18].

There are many other methods to explore for path-finding, path prediction for the VUT is also a challenge. This is a relatively more recent problem but one good source of path prediction solutions is in autonomous driving studies. M. Althoff et al. demonstrate a way of using model based, probabilistic collision detection to aid autonomously driven cars to anticipate the movements of other vehicles on the road [3]. This particular work is focused on predicting reachable areas that specifically impact the decisions of the autonomous car, but the authors state the principle is scalable.

## III. Methodology

This paper takes a Design Science (also referred to as Constructive Research) approach to researching synchronisation in ActiveSS. According to A. Hevner et al. "Design science addresses research through the building and evaluation of artifacts designed to meet the identified business needs" [13]. Drawing on a framework adapted from same in figure 4, the general relevance for this research is Autoliv's need to improve their testing capability at the AstaZero test-track facility. Their goal is to meet a growing need for dedicated ActiveSS testing that allows for autonomous VUTs.A preliminary study by A. Knauss et al. on future trends in ActiveSS testing indicates there is also a broad industry desire to have access to more capable test environments, including synchronisation of on track agents and Autonomous VUTs [16].

In this paper we construct an approach, AMPS, for synchronising Test Targets with an Autonomous VUT. The work relies on, and can be evaluated against, a knowledge base that includes Autoliv's current test strategy and existing research into both ActiveSS testing and multi-agent synchronisation. The work on AMPS' will contribute to the Knowledge Base, specifically in the area of multi-agent synchronisation, a gap identified by S. Heinlein et al., and ActiveSS testing in general [11].

### A. Constructing AMPS

AMPS is an artefact in the form of an approach for synchronising multiple agents in an ActiveSS setting. Drawing from related work on both ActiveSS testing and synchronisation in general, the paper describes a high level approach to using path-following Test Targets slaved to a Master Server. It allows for some level of uncertainty as might be expected when incorporating an autonomous VUT to ActiveSS testing.

The Concept section outlines the main components and their roles, the general synchronisation and communication methods of AMPS and factors that need to be taken into consideration if it is to be implemented.

### B. Evaluating AMPS

Referring to the framework in figure 4, the evaluation relies on demonstrating that AMPS meets a business need, in this case, with a specific focus on a company that tests ActiveSS, Autoliv. AMPS should be judged against its ability to achieve these same needs as an improvement over existing knowledge and best practice while introducing the modifications necessary to allow for synchronising with an uncontrolled VUT.

S. March and G. Smith make it clear that evaluation forms an integral part of design science but this is "made complicated by the fact that performance is related to

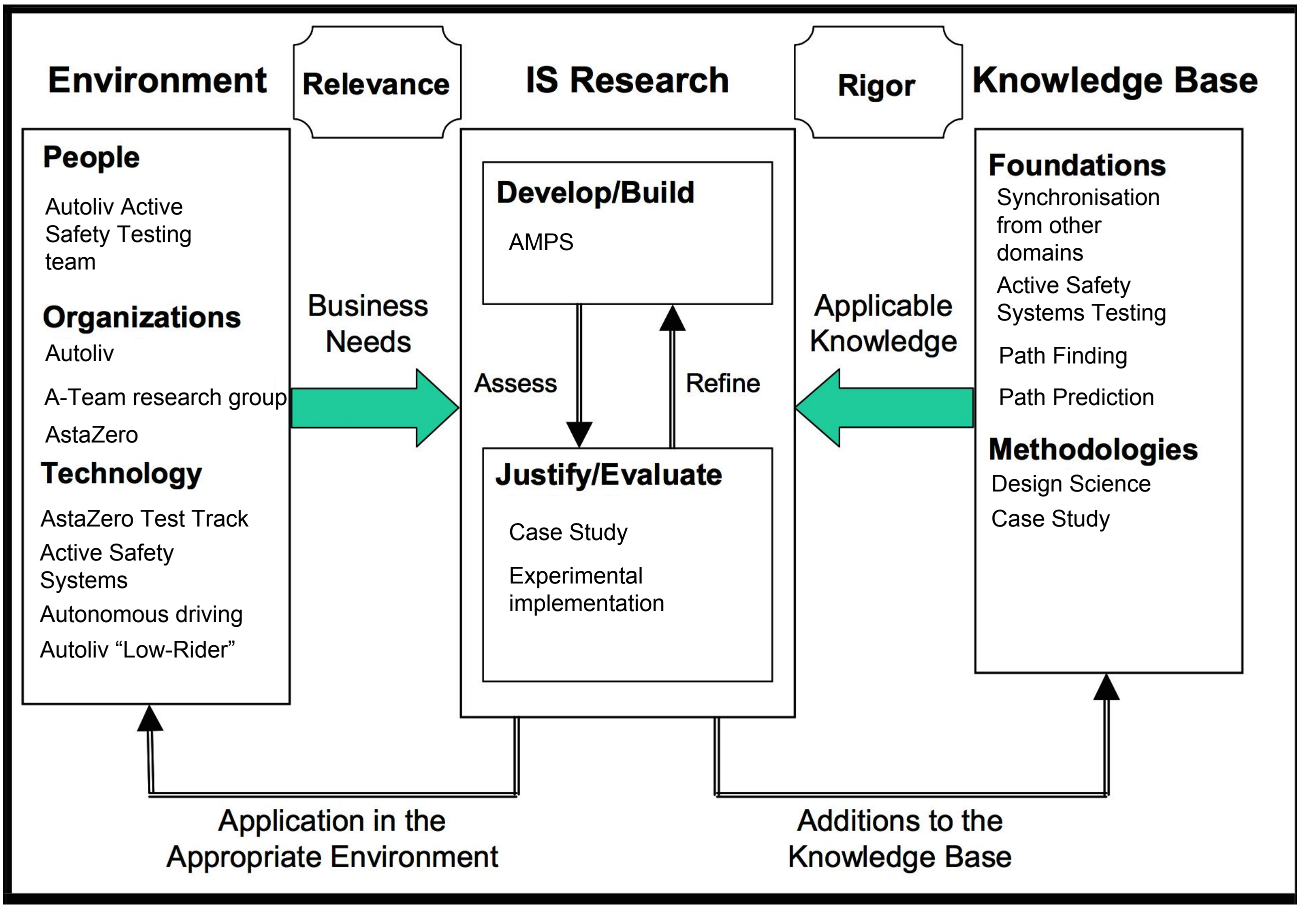


Fig. 4. A Design Science framework from A. Hevner et. al. [13]

intended use" [19]. As AMPS is an abstract concept, evaluating a practical implementation is out of scope for this paper, there are other methods for evaluation however. When evaluating an artefact, A. Hevner et al. [13] recommend a possible 5 categories of evaluation methods; observational, analytical, experimental, testing and descriptive. Observational and descriptive methods are targeted towards abstract artefacts and do not necessarily require implementation. The evaluation of AMPS will rely on two main methods to demonstrate the utility of the artefact: a case study, which is an observational method, and using scenarios of proposed usage, a descriptive method. It will be evaluated against quality criteria established from a reading of the related literature, outlined in table I

H. Schöner et al. and S. Heinlein, for example, propose precision and reproducibility as key qualities needed in any ActiveSS testing model. As described in the Background section, existing best practice is currently to use path-following agents on the track [23] [11]. When a test begins, all agents are synchronised and use closed-loop controllers to ensure that lateral and longitudinal accuracy are maintained. Real-time clocks are used to ensure timing precision. By keeping all parameters equal, the test can be reproduced many times over, although resets are time consuming.

Safety is also critical. Collisions cost time and money and, if proper procedures and fail-safes are not in place, can be dangerous to people with cause to be on or near the track. Autoliv, Anthony Best Dynamics and other manufacturers of Test Targets have developed platforms for Test Targets that are low to the ground and can be run over by a VUT in the event of a "collision" with no ill effects. They coordinate their Test Targets in advance, again utilising the predefined path system. Any new proposal needs to be just as safe.

Flexibility is a requirement that takes into account the business need aspect of the design science framework. Autoliv partners with a number of other companies in the AstaZero test track and this implementation must fit with the expected technology and infrastructure mix as well as take into account the possibility of third party Test Targets.

The feasibility of developing APMS is a quality less easily examined in an abstract manner. This is also a key

TABLE I
EVALUATION CRITERIA

| | |
|---|---|
| Precision | Must have the ability to provide precise Test Target position, velocity and timing to meet strict parameters of test scenarios (e.g. EuroN-CAP AEB protocol) |
| Reproducibility | Tests must be reliably repeatable to provide valid data for analysis |
| Safety | Prioritises significant reduction of damage risk where avoidable and provide manual and automatic emergency or safety critical fail-safes |
| Feasibility | Must be feasible within infrastructure limitations and compatible with existing technology |
| Flexibility | Capable of adaptation to future advances and third party systems within reason |

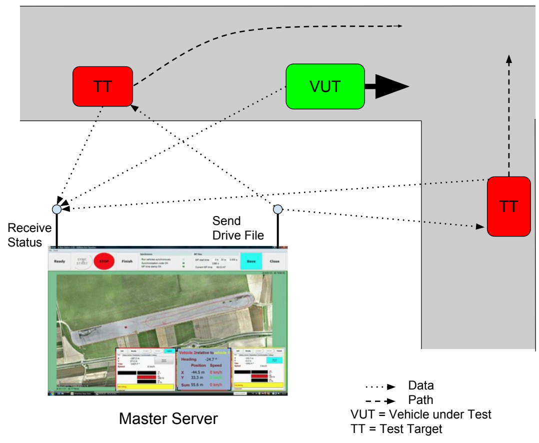


Fig. 5. An overview of AMPS.

concern of Autoliv's, particularly in reference to updating paths on-the-fly due to current work on their "low-rider". This aspect was identified as a particular risk to AMPS and so an extra step in its evaluation was required. Implementing a modified version of existing control code for Autoliv's Test Targets to take path instructions on-the-fly showed the viability of this important aspect of AMPS.

## IV. CONCEPT

As previously noted, the future of ActiveSS is trending towards increased or complete vehicle autonomy and connectivity. Testing the effectiveness and safety of these systems in an environment that comes close to replicating real-world conditions is difficult. These multi-agent systems need a way to continually synchronise the movement of the various agents on track in way that is reactive to the VUT. When considering methods to achieve this, the key requirement is generating usable data through the consistently successful execution of test scenarios.

The proposed concept, AMPS, combines an adaptive master/slave architecture with the existing best-practice strategy of using path-following Test Targets [23]. There are three main component types in AMPS as seen in figure IV:

- The VUT, as input to the system, will behave independently and generates the uncertainty that will trigger decisions in the Master Server
- The Master Server is where scenario parameters are defined, vehicle tracking occurs, paths are predicted and generated and users can interact with the system
- The Test Targets, the slaves in this approach, receive and follow the paths generated by the server

The main advantage of using a centralised approach is that all decisions can be made with as complete a set of information as possible. Track infrastructure, such as sensors, and all of the agents generating data can all be taken into account. Synchronisation will be one-to-many as opposed to a more complex, many-to-many situation. Centralising the decision making also provides room for rapid iterative improvement of the algorithms and process used to determine and predict paths.

### A. AMPS

Looking at AMPS more closely, it can be seen that the basic idea is an extension of the open-loop control concept used in the existing best practice. The controller in this case is the networked Master Server. Its input is the initial test scenario, inclusive of any number of parameters for success, and the actions taken by the VUT. The Master Server is aware of the current position, speed and trajectory of all vehicles and the layout of the track and is responsible for plotting paths for Test Targets. The behaviour of the Test Targets is the output.

*Test Target:* The Test Target will be patterned on existing platforms in common use, using a well understood closed-loop control system [6] [23]. These are nominally independent, each using an embedded real-time processor device, typically running at 100Hz and reading a path (or drive) file, that has detailed instructions for desired position, heading, velocity and acceleration.

Each main thread loop will interpret a line from the path file and, using its own understanding of where it is and where it is going based on internal measurement equipment, adjust its speed and heading as required. These path file are initially loaded before the test begins. In order to update the path files as changes are made by the master server, the Test Target has a WiFi receiver or a 4G modem and the controller is able to receive and interpret information over the air.

The path files are simple text files but should really be considered time based instruction lists. When a change is received, the Test Target program must interpret the

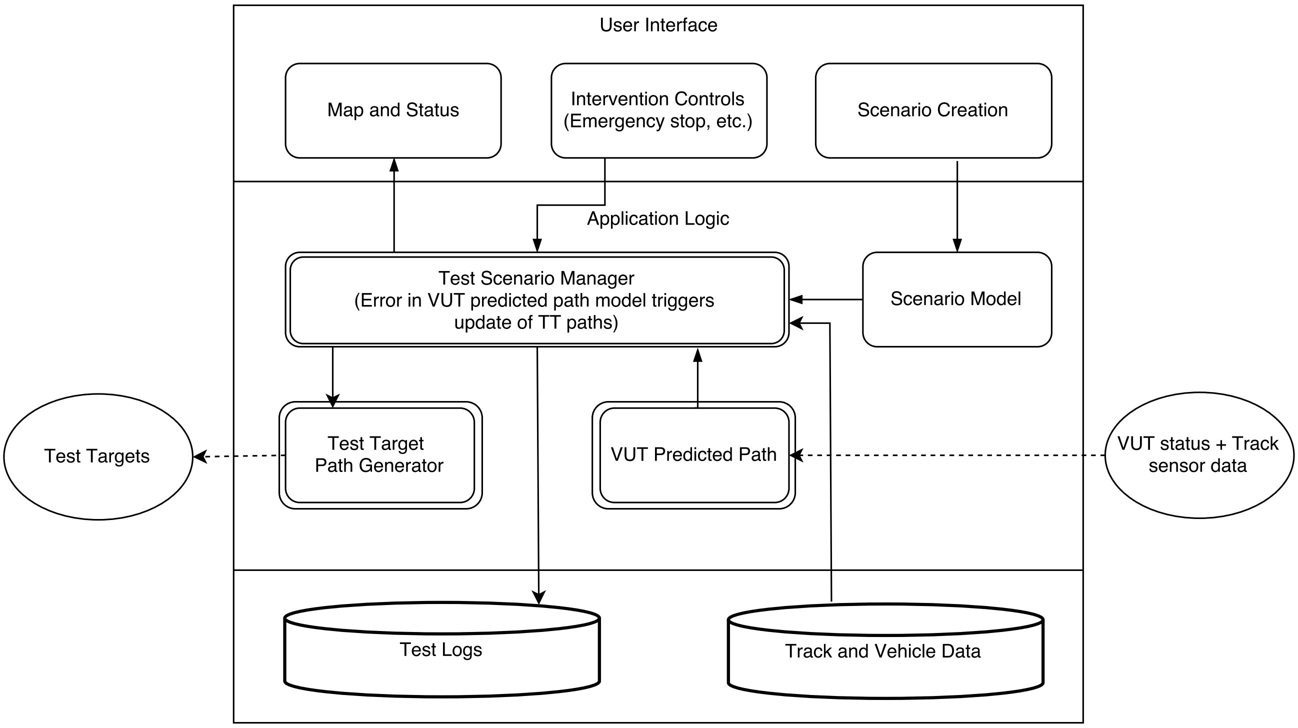


Fig. 6. Proposed architectural view of Master Server

packet, find the insert position for the first instruction and then propagate subsequent changes through the rest of the instruction list without missing a beat.

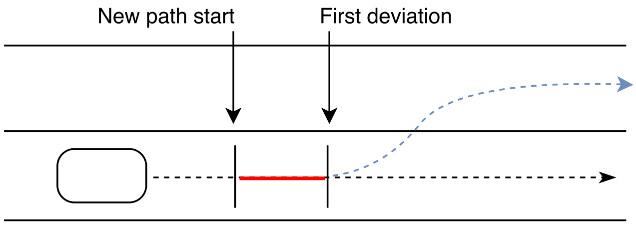


Fig. 7. Demonstrating the path lead time.

A key issue is how to ensure that there are no "jumps" caused by gaps between the existing path and the updated one. To handle this, it is necessary to establish an empirically determined lead time for when the new path will deviate. This enforces continuity between paths. For each new path the Master Server relies on the synchronised timing of each unit and plots its path with the understanding that changes will not propagate immediately, as indicated in figure 7. This makes AMPS unsuitable for quick, granular path updates. This is somewhat by design as Test Targets should operate as stably as possible for the most consistent data generation and so as not to provoke unwanted reactions from the VUT.

The path file is received over a network socket in the background of the main timed control loop. The new path file must replace the instructions received in the control loop, these are processed one at a time using a Real-Time FIFO so the new path file must, even after network transmission and interpretation, have a time variable that is greater than that of the current timing count of the board. There is some room for error as the closed loop system can handle a certain amount before oscillations become too great.

The Test Target is also always reporting relevant data to the master server: position, velocity, heading and other information. This is used for data collection and visualisation but also to aid in fail-safe situations to determine if there has been some unreported error in communication or control, triggering an emergency shut-down or manual intervention.

*Master Server:* The Master Server will handle input data from each vehicle on the track, possibly from multiple scenarios at once, along with any other track based sensor data generated in each test. In figure IV-A, a simplified architectural view of the Master Server, there are three key modules identified as central to the AMPS approach.

The Scenario Manager will be a decision making module that allows test designers to define scenarios by entering the parameters required (such as a potential collision at a T intersection at 60km/h) and provide the data for displaying the current and proposed or predicted paths of each agent as well as triggering the mechanism for generating path files for the slaved Test Targets. This module would also be able to provide limited control, shutdown commands and the like to the test supervisors via an interface.

The Path Generator will be responsible for actually regenerating path files based on constraints provided by the Scenario Manager and sending them to the slaved Test Targets on the fly. The recalculation of a path takes into account the position of a Test Target and the current time-from-start of the test. The updated path should meet both the criteria of the test scenario and the physical limitations of the Test Target (i.e. the recalculated path does not require an impossible manoeuvre too swift an acceleration). If there is no viable path, the test can be stopped or only the particular agent affected manoeuvred out of the way.

The third module, the Path Predictor, is responsible for predicting the likely path of the VUT. It provides this information to the Scenario Manager where it will be compared against the previous prediction and the desired path for the scenario. The module constantly measures deviations from the VUT expected path. If these deviations cross a threshold then a recalculation is required.

*Vehicle Under Test:* The VUT is considered to be an independent actor in this system. Its path is predicted as best as the master server is able but the level of freedom it has in terms of deviation from its initial understanding of the test scenario is dependent on how the test is designed. If a test requires very strict conditions, such as handling a close pass at high speed, a highly trained human driver or a very constrained AD might be used and the master server can consider the initial path prediction more reliable. If the test is fairly freeform, such as a general navigation test, the Master Server will consider the situation to be more volatile and will place less value on the initial proposed path. The VUT transmits all of its own localisation and telemetry data and track sensor data can also be used in getting a better fix on its behaviour. Although not practical in the case of a human driver (unless there are ActiveSS systems installed to make predictions based on driver behaviour), it would make sense to also transmit information about the planned path of the Autonomous Driver itself.

### *B. Communication and Synchronisation*

An overview of the communication process is presented in figure 8. It demonstrates the important signals exchanged between the agents on the track and the Master Server. In previous sections the understanding has been that the VUT operates completely independently, there is an exception, the start signal for beginning the tests. This is important for the synchronising not only the physical agents, but the distributed data collection that is used in analysing the results of the test.

As indicated in the Background section, synchronising the clocks of all devices is an important part of this kind of distributed architecture. PTP is designed for distributed deployment [1] and is suggested as one option for clock synchronisation. The final choice for clock synchronisation could be affected by factors such as already in-place infrastructure that relies on a different synchronisation protocol or consideration for third party requirements. By synchronising the clocks, you can synchronise the movement of each vehicle according to an agreed upon time scale and gather better quality data for test analysis. It also means that a simple protocol can be established for path update messages.

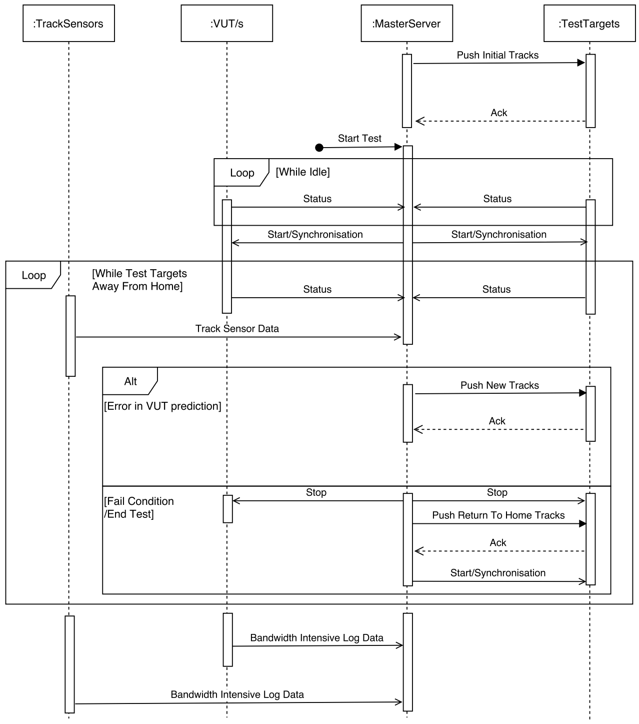


Fig. 8. Overview of proposed communications sequence

As described earlier, a new path should preserve some small continuity to the old to ensure stability. The Test Target should acknowledge an acceptable path once it is received. It will not check the whole validity of the path (except by perhaps a checksum) but, based on timing, the first few steps of the path are checked against the next few in the existing path. A failure can be fatal to the test or simply trigger the Master Server to generate a new path with updated understanding of network lag (or other contributory factor) depending on the urgency of the path correction.

### *C. Factors affecting AMPS*

The central challenge for AMPS is handling the uncertainty introduced by a VUT that is not predictable in real-time. Instead of relying on static paths, in AMPS the master server is responsible for generating new paths adaptively. The decisions required need to be both reactive and predictive. If the VUT deviates from the expected path of the initial scenario, the system needs to react and begin predicting new paths for each Test Target to ensure each is able to meet the scenario's requirements. Each initial path will have been designed to have the Test Target arrive at a destination, likely relative to the VUT, at a certain speed and orientation. These conditions need to be matched in any new planned path.

To be effective at reacting to changes, the AMPS system requires detailed knowledge about the VUT. Existing systems use dGPS or RTK-GPS to internally establish precision position information but effective communication of this information, along with speed and trajectory, is subject to the infrastructure limitations at the track. Establishing these limitations is an empirical problem, testing and measurement in an implemented real world system is required. The potential for error and latency must be built in to the system from the beginning, particularly in the case of Test Targets receiving path file updates.

The real-time reliability of the Test Targets comes at a cost. The boards used as controllers are robust and capable of handling the calculations and hardware component control required to drive but IO and autonomy are limited. Receiving path files and maintaining a smooth trajectory while interpreting them and propagating changes is a key challenge.

The existing best practice is basically an open-loop control system, with time and the scenario conditions the input variables and the behaviour of the VUT and Test Targets being the output. We can also consider AMPS as an open loop control system with multiple independent agents in a similar sense. The difference here is that the Master Server, operating as the controller, is dependent on a great number of influencing factors. Some of these will be measurable variables, such as the error generated by unpredictable VUT behaviour, others will be triggers (often for failure), such as the failure of a path file propagating to a Test Target. Others might be a combination.

Table II is a non-exhaustive list of higher level factors that may impact the system, whether this *factor* is *measured* or a *trigger* (M/T) and the *actions* the system might take in response.

A point to note is that the main driver of instability, the VUT behaviour/status, is considered to be independent at this high level, as in the system does measure its impact on the VUT. This is, of course, not entirely accurate. As we are testing Active Safety Systems, the test scenario will be designed to use the Test Targets to force a particular behaviour from the VUT however we may inadvertently introduce even more uncertainty to the system if a Test Target triggers an unexpected response from a Autonomous/Human driver. Trying to avoid these oscillations is one of the main drivers for (right now) avoiding more autonomy in the Test Targets.

TABLE II
IMPORTANT FACTORS AFFECTING AMPS

| Factor | M/T | Action |
|---|---|---|
| VUT deviation from planned scenario | M | Adapt Test Target paths where possible, show warning, if too great cancel test |
| A scenario parameter becomes unreachable | T | If essential, gracefully stop test. If secondary, prompt warning for user with option to cancel or continue |
| Test Target deviates from expected path | T | Gracefully stop test if possible. Emergency stop if dangerous situations predicted or Test Target is not responding |
| Calculated paths to original waypoint would create dangerous or difficult to achieve conditions | M | Recalculate paths to an alternative waypoint if possible. If not possible gracefully stop test |
| Test Targets and/or VUT stop responding | T | Gracefully stop test if possible. Emergency stop if dangerous situations predicted or vehicles not responding |
| Unanticipated collision | T | Emergency stop test |
| Path continuity not long enough on new path received by Test Target | M | Master Server checks the calculated continuity delay for sent path and adjusts and resends if still time |
| Manual cancel signal | T | Gracefully stop test, prepare for redirect to initial setup |
| Manual emergency stop | T | Emergency stop test |

## V. Evaluation

The best way to evaluate the overall success of AMPS is to put it in context. It is designed to meet a business need, introducing autonomous VUT into ActiveSS testing as identified by Autoliv and emphasised by research by A. Knauss et al. about the future of ActiveSS [16]. A case study with Autoliv and an examination of literature helped define the existing best practice to build upon. From this, the 5 system qualities in table I: Precision, Reproducibility, Safety, Flexibility and Feasibility, were derived. These qualities would have to be met for AMPS to meet this need.

Describing the AMPS design from a system quality perspective requires a frame of reference. The test scenarios defined by EuroNCAP as a part of their safety awards for ActiveSS are common use cases in ActiveSS testing. Specifically we will use a scenario, Car-to-Car Rear Braking, from the AEB Test Protocol version 1.1 [8] to demonstrate the capability of AMPS and how its design maintains Precision, Reproducibility and Safety. The Flexibility of the system will be described in terms of how well it can be integrated with existing and third party infrastructure and technology. The Feasibility of its implementation will be shown by presenting an experimental implementation of updating path files on-the-fly using a simulator for Autoliv's Test Target.

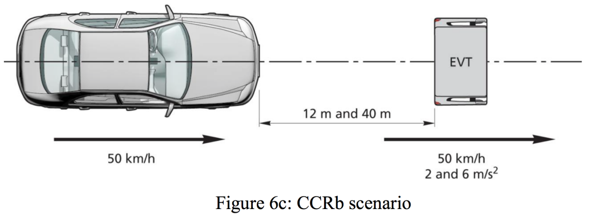


Fig. 9. Advanced Emergency Braking – The CCRb tests will be performed at a fixed speed of 50km/h for both VUT and EVT with all combinations of 2 and 6m/s$^2$ deceleration and 12 and 40m headway ©2015 EuroNCAP

### A. Precision

Figure 9 presents one of the more complex AEB scenarios. The EuroNCAP guidelines state that tolerances for the speeds of both vehicles are ±1km/h, the longitudinal distance tolerance is ±0.5m and the latitudinal error must be within ± 0.1m.

Maintaining the precision of the existing best practice of using only path-following robots is difficult. The uncertainty introduced by the autonomous VUT means that the correct conditions are not certain to be met at an exact point on the track any more. The VUT may be slower to accelerate up to 50km per hour than planned or may miss its lateral path mark for a time due to lane discrepancies.

AMPS preserves the precision of the Test Targets by updating its path so that it remains in bounds of the test scenario without any unstable oscillations that might be caused by a feedback loop or other autonomous alternative. When the VUT hits 50km/h, the Test Target will be in place to decelerate as required.

As ActiveSS advance and more complex scenarios are required, AMPS will prove to be even more valuable. The above scenario only requires two agents. If we consider the scenario presented in figure IV, we can see that the level of complexity quickly increases. If both Test Targets are to provide the tenuous conditions shown and test the decision making process of the VUT, the timing must be precise.

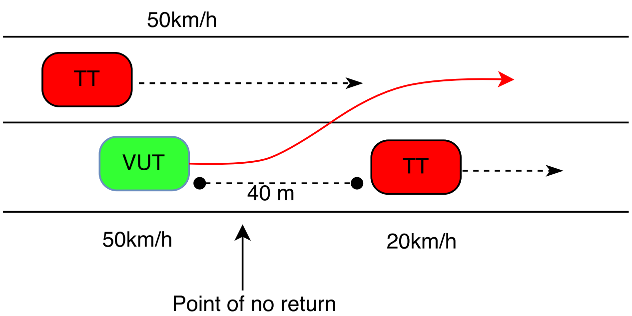


Fig. 10. Example of Autonomous decision making scenario

The cost of maintaining that precision shows, however, when one or more agents deviate from the expected course and the Master Server cannot suitably adapt for one of a number of reasons. A fail case occurs and the test must be reset. The closer deviations occur to a major test event (e.g. a swerve just before potential collision) the less likely a test will be cancelled by the Master Server and the moment must be reconstructed in the logs to determine if parameters were still met. This last is also a threat to precision in general. Overall, these risks are limited and, compared to alternatives such as agent autonomy, the slaved Test Target system provides the best precision in most cases.

### B. Reproducibility

A test scenario will seek to place the VUT in one or more circumstances where its ActiveSS will be triggered and provide data. For that data to be valuable the test conditions must be able to be reproduced consistently. In practice, exact conditions will be hard to repeat where the VUT is not under complete control. What can be reproduced is the relative pose and velocity of each vehicle involved within defined bounds. The AMPS system allows these bounds to be set centrally. Additionally, all data from the tests is logged centrally. The synchronised nature of the tests allows for reconstruction and evaluation to be integrated into the workflow from a central point.

In the case of the AEB scenario in figure 9, if the VUT is running at variable speeds or acceleration, there needs to be a way to adapt consistently. The Master Server has knowledge of the status of all vehicles and is further aided by track side sensors. This gives it an advantage in consistently coordinating all agents successfully over alternatives involving some level of autonomy.

The Master server is designed to record and store all data generated by the agents communicating telemetry. It will record this with a view to keeping a time series, using the PTP synchronised clock as discussed in the Background section. The ability to reconstruct and review the timestamped data in a centralised manner will greatly speed up evaluation.

Any particularly interesting edge case that occurs during a test can also be reproduced. If the ActiveSS on the VUT in 10 makes a strange decision in this complex scenario, it can be recreated. The Master Server stores all of its own decisions as well as the status and log informations of participants and should be able to repeat the conditions, an advantage of the centralised open-loop control. Looking at the AEB scenario again: if a particular angle of impact causes AEB to fail for some reason, say due to an unexpected turn by the VUT, this test can be reproduced either by replicating conditions or actually generating a path file for a robot controlled VUT and Test Target recreating the full scenario.

### C. Safety

A safe system is, in this context, considered to be one that prioritises the safety of both the human participants inside the VUT (drivers or passengers) or track side. It also encompasses the general safety of the vehicles involved. Minimising damage is important from not only a cost perspective but in order to ensure that tests are repeatable. Aside from the physical assurances given by the low-profile and durability of the modern Test Target platforms employed by Autoliv, AMPS provides a number of ways in which safety can be increased.

A central pillar of the safety requirement is that there is an emergency stop protocol in place. As it exists now, the Autoliv Test Target has a radio controlled stop button. This can be augmented by having a centralised Emergency stop intervention. Initially this can be provided by over-the-network signals built into AMPS. Emergency stopping is a last resort, using the AMPS interface to set boundary conditions is way in which emergency situations can be avoided before they occur. Additionally, warnings can be built into the system that will allow for additional information to be provided to operators to call a stop when unforeseen situations occur.

As indicated in H. Schöner et al., full control of agents on the track provides the optimum safety [23]. While the introduction of an unpredictable element adds to system instability, the centralised control of the Test Targets limits uncertainty to just the VUT. While not eliminating the potential for unwanted collision (VUT/Test Target impact is sometimes expected), it drastically reduces the chances. In fact the system has the potential to be quite responsive in cases where uncertainty is high. There has been research that shows that path-following vehicles can be tuned to be stable in poor conditions [10].

### D. *Flexibility*

Autoliv is currently partnering with a number of other organisations on the AstaZero test track project. This is the context in which AMPS would operate and it is conceivable that a number of ActiveSS providers, vehicle manufacturers and even third party suppliers of Test Targets will want to interface with the system. As such any implementation needs to be as agnostic as possible in terms of technology, communication and components.

The Test Targets that the AMPS design is centred around is that developed by Autoliv, however the basic premise of this design is one that has been used by, for example, Dr. Steffan Datateknik (DSD) [7] and Anthony Best Dynamics (ABD) [4]. ABD, in particular, also uses path files to control its Test Targets. Other Test Targets might be static, or even gantry based pedestrian targets such as those used in [11]. It also pays to consider cases where there is a need to go back to the current best practice of solely robot driven vehicles or, inversely, increased autonomy in Test Targets or multiple VUTs.

AMPS is a way of tracking agents and synchronising their moments with timing based on PTP synchronisation, a protocol well established in distributed control systems. As designed above, AMPS synchronises Test Targets this by pushing new paths as needed. It is simply a matter of adjusting the information that is being sent by the Master Server, from simple start signals or timed triggers to direct control using UDP and CAN instructions. If the Test Targets design evolves to become more autonomous, the Master server can conceivably be used to synthesise and relay all the agent status and track sensor data to Test Targets. This would improve each agent's decision making while continuing to monitor for fail or dangerous conditions.

### E. *Feasibility*

Feasibility is not so much a quality as it is a an evaluation of whether the approach can be implemented for real ActiveSS testing. Specifically, an evaluation of feasibility is concerned with the challenges of implementing the design. A full understanding of the feasibility of AMPS is beyond the scope of this paper, as it is dependent on a number of infrastructure details and theoretical concepts such as path-finding that require further study and implementation.

However, from Autoliv's perspective the AstaZero test track will provide a lot of the necessary infrastructure such as a V2I Network over 4G and various track based sensors. They are a part of a working group that is tasked with solving some of the communications problems an AMPS implementation will rely on [14] [5]. There is also current and ongoing research into the path generation aspects of the Master Server, as described in the Background section. What Autoliv are most concerned with is the feasibility of updating path files on-the-fly on their Test Targets and the implications this has for synchronisation.

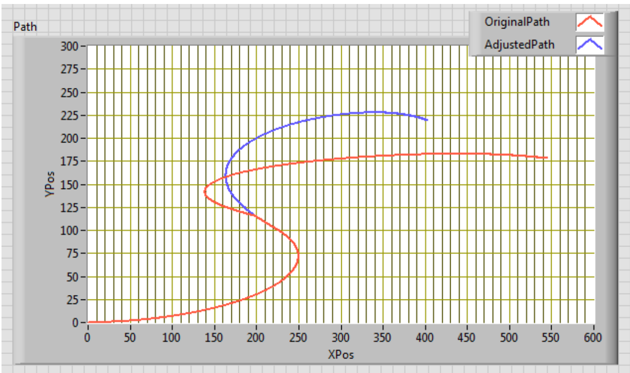


Fig. 11. Result of updated path file simulation

To meet this challenge, a small prototype was implemented in LabView, targeting a National Instruments embedded controller. Figure 11 is generated by an existing path-following simulation designed to emulate Autoliv's Test Target behaviour. The simulator is fed command signals by a timed loop, running at 100Hz. The tolerance for missed signals, according to the vehicle designers, is less than 100ms at a high speed. It is important, therefore, to both receive the data efficiently and feed it to the simulator.

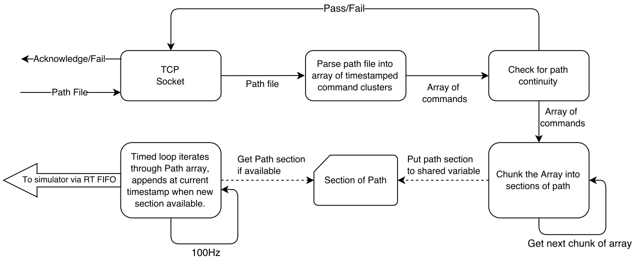


Fig. 12. Illustration of path file updating on-the-fly

The prototype process involves receiving an initial path file from the Master Server and starting the simulator and timed feed loop. When the Master Server wishes to update the path file (manually triggered in this case), it sends it over a TCP socket. The file is then parsed into an array of line-by-line drive commands. This is checked for continuity and an acknowledgement or fail signal returned to the Master Server. If not a fail case, the array is then "chunked".

The embedded board does not have a great deal of memory and passing large arrays around (at 100Hz a 10 minute file can be 60000 lines) is expensive and can potentially interrupt the real time flow of the board. To limit this the path is split into smaller chunks before it is copied to a shared variable (protected against race conditions) and a signal is given to the timed loop to append the new section of the path at its first point. This is still an expensive process but the smaller the chunk, the less onerous the process.

Path file updating on-the-fly is, therefore, shown to be feasible in a demonstration sense. The process works and there are no timing errors or interruptions in the simulated trajectory. However the prototype needs to be implemented on the real controller for true assurance.

## VI. Discussion

The AMPS concept is one way of approaching the topic of testing autonomous or human driven VUTs in Active Safety testing. It can be thought of as upgrading the infrastructure of the test track. If the system works as it is supposed to, the Test Targets are a part of the track in the same way the gantry-carried pedestrians in S. Heinlein et al. are [11]. They are controlled centrally and always where they need to be. The obvious key difference over existing best practice is that there is now an uncontrolled element.

Slightly less obvious is the enhancement to interactivity with the system. Certainly, the partners at AstaZero are already moving towards greater interaction with track infrastructure and it is an obvious next step to include the Test Targets along with the rest.

This paper only provides a high level view of what the AMPS concept might look like. It is not a comprehensive examination of how this particular solution could be implemented, nor are all of the implementation challenges necessarily easy to overcome. The paper does provide some direction on how to proceed with aspects like communication protocols and path-finding for example but these require further work.

AMPS as it is proposed here is not the only possible direction, however, and in the Autoliv case study there were proposals for alternative ways to introduce autonomous VUTs into ActiveSS testing.

### A. *Alternative Approaches*

Autonomous agents are presented in some sections above as unsuitable as Test Targets but are not an inherently bad idea. The Master/Slave pattern was adopted for AMPS to try and simplify and control the Test Target situation on the track. By moving the adaptivity to a central server, however, any response to changes in VUT behaviour is, by necessity, delayed. This increases the chance that there may be a scenario breaking event - such as an unexpected lane change into the path of a Test Target - that cannot be handled. There is scope for incrementally allowing some level of autonomy, perhaps eventually full autonomy, as the technology and algorithms become more stable and predictable. Of course a central server can still be used to monitor the projected paths of all the agents to avoid dangerous conflicts. It also bears considering that in a static test track environment with communication possible between agents, the Test Targets can be given significantly more assistance in estimating ground truth. The obvious disadvantage, as mentioned above, is that any Test Target independence will increase uncertainty across the system. A system that relies on autonomy by design will also tie the track environment to those vehicles that meet a very specific specification, reducing the flexibility of the system.

On the other side of argument is the possibility that allowing both lateral and longitudinal flexibility in the system from the beginning may prove too difficult. Any proposed system must be able to eventually handle multiple VUT agents with uncertain lateral and longitudinal movements. It could be a reasonable proposition, however, that an initial incremental stage, restricting the system to longitudinal uncertainty (requiring this to be controlled in the VUT) would be preferable. It would allow for more stable testing for those tests, such as AEB, that do not require lateral parameters. Turning, swerving or lane changing involves much quicker and more disruptive movements that could difficult to handle adaptively, particularly with a central server handling the adjustments. The existing best practice, using robot driven VUTs, and/or limited interaction with Test Targets (e.g. restricting close lane passing) could be used where lateral movement is a key requirement of the scenario.

### B. Threats to Validity

The main threat to the validity of this paper is due to the abstract nature of the design: Implementation is out of scope for the paper as written but regardless, hard data is limited and many key details are left to the implementation stage. The focus on extending the existing best practice provides some solid ground but, so far as we are aware, no testing tracks have implemented an adaptive system with this particular mix of adaptation and control.

As well as the exclusion of implementation details, it must be pointed out that the evaluation is not necessarily exhaustive. Relying on a solitary case study does not provide a total picture of the feasibility and flexibility of the system for example. Fleshing out the details of AMPS and providing a more rigorous evaluation of its qualities in a more in-depth study will be necessary next step to demonstrating its viability.

AMPS is one way of looking at the problem but it is not the only solution, as we see in the alternative approaches section. By deliberately avoiding autonomy for the sake of stability it is possible we are on the wrong side of history. Arguments were made to support this decision but even as a part of the case study there were suggestions that maybe some level of autonomy was desired. At the very least, however, AMPS is designed to be flexible and there is no reason why it cannot function effectively as a clearing house for information and an interactive interface for test engineers.

The abstract nature of the AMPS design means that it is flexible and technology agnostic. For the scenarios as defined it should be generalisable where there is call for controlled testing of ActiveSS with Autonomous VUT. However, many ActiveSS test environments do not rely on the near flat platforms used by Autoliv and others. It may less useful in situations where these are deemed unnecessary or impractical. One example is the pedestrian tracking work being done by S. Heinlein et al. They have a perfectly functional test environment that is better served by the direct control systems they use.

In this paper we provide an example of how to update path files on-the-fly. This relies on the particular technology in use by Autoliv. It may be impractical to expect third party manufacturers to have the ability to implement similar update on-the-fly systems. Additionally, without specifying a defined protocol, something that is definitely required in any future work, updating files could prove an unstable "hacky" solution to the problem.

Finally, the basis for this work is to improve the testing of ActiveSS on proving-grounds. The existing best practice uses test scenarios derived from, amongst other sources, work by organisations such as EuroNCAP. The scenarios are created to emulate real world conditions but if they are shown to be inadequate in this regard or outdated, the validity of the AMPS design as a way to adequately test ActiveSS may be threatened.

As autonomy in vehicles becomes more of a factor in real world driving, test scenarios in general use are expected to be updated as well. AMPS has been designed to take into account what are likely to be more complex, multi-agent scenarios. It is possible, however, that there will be unforeseen requirements that render the current AMPS design unsuitable. Similarly there may be existing, unexamined scenarios in ActiveSS testing that are outside the capabilities of AMPS that may threaten the generalisability of the findings presented.

## VII. Conclusion and Future Work

The AMPS concept is a step towards accommodating autonomous vehicles into ActiveSS testing. AMPS takes the existing best practice and adapts it using synchronisation methods that are adopted from other robotics fields. In doing so, it takes into account the unique requirements of vehicle and ActiveSS testing, preserving as much control for test designers as possible over the parameters of a scenario. It also centralises much of the existing process reducing the overhead of analysis and resets.

Future work on AMPS and autonomous VUTs in ActiveSS testing should be based on implementation. Clarifying some of the unknowns in the design, such as the path generation problems and the specifics of necessary protocols for communication on real networks is an expected next step. It would also be of value to explore some of the alternatives suggested, increasing or decreasing agent autonomy as the situation calls. It will inevitably become necessary to accommodate multiple, fully autonomous vehicles as the technology becomes more prevalent.

## VIII. Acknowledgements

We would like to thank Per Gustafsson and John Lang from Autoliv Development AB and Johan Forsberg from RCI Robocraft Innovation AB for their kind assistance. We would also like to thank Christian Berger of Chalmers University and Henrik Eriksson of SP Technical Research Institute for their invaluable feedback.

## References


[1] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)*, pp. 1–269, July 2008.

[2] Z. A. Algfoor, M. S. Sunar, and H. Kolivand, "A comprehensive study on pathfinding techniques for robotics and video games," *International Journal of Computer Games Technology*, vol. 2015, p. 7, 2015.

[3] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, no. 2, pp. 299–310, 2009.

[4] Anthony Best Dynamics. (2016) Vehicle and pedestrian soft crash targets. [Online]. Available: abd.uk.com/en/adas_soft_targets

[5] AstaZero. (2016) Astazero test track. [Online]. Available: http://www.astazero.com

[6] Autoliv, "Driving robot control document." Autoliv, 2016.

[7] Dr. Steffan Datateknik. (2016) Ufo. [Online]. Available: http://www.dsd.at

[8] EuroNCAP, "EuroNCAP Test Protocol – AEB systems : Version 1.1," 2015.

[9] G. J. Forkenbrock and A. S. Snyder, "NHTSA's 2014 Automatic Emergency Braking Test Track Evaluations," Tech. Rep., 2015.

[10] H. Fredriksson, J. Van Deventer, N. Engström, and J. Ash, "Automotive winter testing with path-following steering robot," *International Journal of Vehicle Systems Modelling and Testing*, vol. 10, no. 1, pp. 29–52, 2015.

[11] S. Heinlein, S. Hahn, and K. Zindler, "Control methods for automated testing of preventive pedestrian protection systems," *International Journal of Vehicle Systems Modelling and Testing*, vol. 10, no. 2, pp. 127–147, 2015.

[12] F. Hendriks, R. Pelders, and M. Tideman, "Future testing of active safety systems," *SAE International Journal of Passenger Cars-Electronic Electrical Systems*, vol. 3, no. 2, p. 170, 2010.

[13] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[14] J. Jacobson, H. Eriksson, P. Janevik, and H. Andersson, "How is astazero designed and equipped for active safety testing?" in *24th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, no. 15-0071, 2015.

[15] M. Kapadia, K. Ninomiya, A. Shoulson, F. Garcia, and N. Badler, "Constraint-aware navigation in dynamic environments," in *Proceedings of Motion on Games*. ACM, 2013, pp. 111–120.

[16] A. Knauss, C. Berger, and H. Eriksson, "Towards state-of-the-art and future trends in testing of active safety systems," in *Proceedings of the 2nd International Workshop on Software Engineering for Smart Cyber-Physical Systems*. ACM, 2016, pp. 36–42.

[17] Y.-C. Liu and N. Chopra, "Controlled synchronization of heterogeneous robotic manipulators in the task space," *Robotics, IEEE Transactions on*, vol. 28, no. 1, pp. 268–275, 2012.

[18] F. Lucas, C. Guettier, P. Siarry, A.-M. Milcent, and A. De La Fortelle, "Constrained navigation with mandatory waypoints in uncertain environment," *International Journal of Information Sciences and Computer Engineering*, vol. 1, no. 2, pp. 75–85, 2010.

[19] S. T. March and G. F. Smith, "Design and natural science research on information technology," *Decision support systems*, vol. 15, no. 4, pp. 251–266, 1995.

[20] F. Perez-Diaz, R. Zillmer, and R. Groß, "Firefly-Inspired Synchronization in Swarms of Mobile Agents," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 279–286.

[21] R. d. J. Portillo-Vélez, C. A. Cruz-Villar, and A. Rodriguez-Angeles, "On-line Master/Slave Robot System Synchronization with Obstacle Avoidance," *Studies in Informatics and Control*, vol. 21, no. 1, p. 18, 2012.

[22] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *American Control Conference, 2005. Proceedings of the 2005*. IEEE, 2005, pp. 1859–1864.

[23] H.-P. Schöner, S. Neads, and N. Schretter, "Testing and verification of active safety systems with coordinated automated driving," in *Proceedings of the 21st international technical conference on the enhanced safety of vehicles ESV, Stuttgart, Germany*, 2009.

[24] D. Šišlák, P. Volf, and M. Pechoucek, "Accelerated a* trajectory planning: Grid-based path planning comparison," in *Proceedings of the 19th International Conference on Automated Planning & Scheduling (ICAPS)*. Citeseer, 2009, pp. 74–81.

[25] E. Y. Song and K. Lee, "An application framework for the IEEE 1588 standard," in *Precision Clock Synchronization for Measurement, Control and Communication, 2008. ISPCS 2008. IEEE International Symposium on*. IEEE, 2008, pp. 23–28.