Using machine learning to develop natural, human like vehicle control

Test Methods for Interrogating Autonomous Vehicle Behaviour – Findings from the HumanDrive Project

HORIBA MIRA

Richard Hillman

HUMAN DRIVE

March 2019

Executive Summary

HumanDrive is a collaborative R&D project, part funded by innovate UK, that is developing an autonomous vehicle (AV) that uses artificial intelligence and deep learning to enable more humanistic driving behaviour than that typically associated with AV prototypes. The culmination of the project was a 'Grand Drive' of over 200 miles, incorporating a range of challenging road types. The project is led by Nissan, with a neural network-based control system being provided by Hitachi, and other consortium members conducting research on what constitutes 'humanistic' driving behaviour and on the effect of such automation on the transport network.

HORIBA MIRA is a member of the consortium, providing facilities and support for dynamic testing upon the HORIBA MIRA proving ground and supporting the safety work package (led by the Connected Places Catapult). This report focusses upon methodologies for integrating the topics of safety and testing such that the behaviour of the AV can be interrogated sufficiently to provide safety assurance, and proposes the use of 'Scenario Based Testing', as opposed to extended-mileage uncontrolled validation testing, as a more viable way to expose the system to a sufficient range of challenges.

In order to derive the test cases, HORIBA MIRA developed a process to develop scenarios from a high level of abstraction to a lower level of abstraction. This process starts with an examination of the use cases for the system, in order to identify the underlying functions that the system is required to be capable of performing (such as maintaining a driving path within the lane boundaries, responding to traffic lights, detecting other vehicles etc). A set of high-level 'Functional Scenarios' are then designed such that the complete set exercises each required function in a range of permutations representative of the Operational Design Domain of the system.

The parameters that can be varied within each Functional Scenario are identified, together with their possible ranges, to create 'Logical Scenarios'. Multiple 'Concrete Scenarios' are then created from each Logical Scenario by selecting suitable combinations of parameters such that a representative sample of the possible permutations is taken. Finally, Concrete Scenarios are converted into Test Cases through the addition of logistical information and success criteria.

In order to select suitable combinations of parameters for Concrete Scenarios from the available ranges within the Logical Scenarios, an approach was developed where each parameter range was divided into multiple sub-ranges ('levels') and a pairwise matrix was used to select test points such that each possible combination of levels for any pair of parameters was included in the Concrete Scenarios. A fractional factorial approach such as this was needed as it would be impractical to perform a full factorial study, i.e. to test every possible combination. The selected test points were then randomised within the sub-range provided by its given level to ensure a good spread of values was provided for each parameter. This report describes this novel approach, dubbed 'Fuzzed Pairwise Testing', examines how it can be extrapolated to other forms of orthogonal array beyond pairwise arrays, and compares it with another multi-dimensional sampling method, Latin Hypercube Sampling with Multi-Dimensional Uniformity.

This report also recommends methods to make scenario-based testing more effective and more efficient, such as the need for a repository to capture information on what parameters should be tested for a given Operational Design Domain and what the acceptance criteria should be, to ensure that the test scenarios include a range of permutations equivalent to millions or billions of miles of real-world driving. This would reduce the likelihood of permutations that trigger system errors remaining undiscovered until the system is deployed. Furthermore, the use of 'sequential testing' methodologies to provide a statistical basis for early termination of testing if the performance is suitably far above or below the acceptance threshold is proposed; given the volume of testing required for AVs, this approach could result in significant savings.

Contents

1. Introduction

1.1 Project Description

The HumanDrive consortium consists of the organisations shown in Table 1. The project was divided into 'work packages', with HORIBA MIRA supporting work packages relating to 'Trials, Demonstration and Validation' and 'Safety Management'.

Organisation	Role
Nissan	Lead
Atkins Ltd	Collaborator
Cranfield University	Collaborator
Highways England	Collaborator
Hitachi	Collaborator
HORIBA MIRA	Collaborator
SBD Automotive	Collaborator
Connected Places Catapult	Collaborator
Aimsun Ltd	Collaborator
University of Leeds	Collaborator

Table 1 : Organisations involved in HumanDrive

The consortium developed a prototype Autonomous Vehicle (AV) capable of achieving high levels of automation, with the aim being to successfully demonstrate an autonomous 'Grand Drive' from Cranfield to Sunderland (over 200 miles) in live traffic. This was completed in November 2019 and demonstrated successful navigation of country roads, motorways and dual carriageways. The project commenced in July 2017 and runs for a duration of 33 months, finishing at the end of March 2020.

One of the major innovative aspects of HumanDrive is the development of an advanced control system designed to allow the AV to emulate a natural, human-like driving style. A key enabler for this is the integration of an Artificial Intelligence (AI) controller utilising Artificial Neural Networks (ANNs) and deep learning for perception and decision-making.

In addition to learning from HumanDrive, this report also draws upon the knowledge gained within the SAVVY project, as there was significant cross-pollination of ideas between HORIBA MIRA's contributions to the two projects. SAVVY is a collaboration between AVL (lead partner), WMG, Vertizan, HORIBA MIRA and Myrtle, and like HumanDrive is also in receipt of funding from Innovate UK in line with the strategic objectives laid out by the Centre for Connected and Autonomous Vehicles (CCAV). The aim of the project was to develop a lane centering system for a passenger vehicle in order to serve as a case study for developing advanced processes for Model-Based Systems Engineering, simulation testing and physical test case development. HORIBA MIRA therefore gratefully acknowledge the support of AVL, WMG, Vertizan and Myrtle in addition to the support of the HumanDrive consortium.

1.2 Complexity of Control Systems

The use of AI within the perception layer of prototype AVs and production driver assistance systems (e.g. Autonomous Emergency Braking, Traffic Sign Recognition) is well established. This is due to the proven ability of Convolutional Neural Networks (CNNs), a sub-category of ANNs particularly suited to vision systems, to perform complex classification tasks effectively.

Hitachi are developing their own CNN-based perception system, including a number of innovations that are subject to patent submissions, but perhaps the most innovative aspect of the vehicle is the use of ANNs for decision making (i.e. deciding what manoeuvre to make and when) and path planning (i.e. defining the exact trajectory to be used for that manoeuvre). The use of ANNs within an AV's control system presents a significant challenge, as it is not possible to infer any human-understandable meaning from the weights assigned to nodes within an ANN. The system therefore has to be tested as a 'black box', as it is not possible to apply 'white box' test methods that specifically target the inner workings of the system.

However, this problem of testing a system that is not fully understood is not restricted to ANN-based control systems. Even when using a traditional algorithmic approach for the control system, the extreme complexity of both the control system and the environment that it is deployed within mean that the behaviour may not be able to be understood as a white box, and may therefore have to be tested as a black box, or potentially a grey box (where the inner workings are partially understood).

Furthermore, such complexity may blur the lines between deterministic systems (where the response to the same input is always identical) and non-deterministic systems (where the response varies according to time history, further learning undertaken, pseudo-random numbers affecting decisions etc.). For example, the smallest variation such as a bird flying past could affect the response of the vehicle in two otherwise identical test runs; if the presence and exact position of that bird is not logged within the test data, a deterministic system would appear to be behaving non-deterministically to anyone analysing the results. Note that ANNs and algorithmic systems can be either deterministic or non-deterministic, even though the term deterministic is often erroneously used to mean algorithmic (as opposed to ANN-based) systems.

HORIBA MIRA propose the terms 'interrogable' and 'uninterrogable' to capture whether it is possible in practice to fully understand the system behaviour such that outputs can be mapped to inputs. This overcomes the ambiguity caused by white boxes being so complex that they effectively become black boxes (especially to industry regulators who may not have access or the ability to analyse all details of the internal workings), deterministic systems and their environment being so complex that they appear non-deterministic, and the term deterministic commonly being mis-applied.

Whilst in theory it is possible to decode the inner workings of even the most complex systems in the most complex environments, given enough testing and analysis, in practice the number of patterns needing to be decoded increases exponentially with the number of parameters acting as inputs to the system. Therefore, it is expected that all AV control systems will be found to be uninterrogable in practice.

However, even if it is impossible to fully understand the relationship between inputs and outputs, this report argues that it is possible, albeit very challenging, to develop a test programme that samples enough points within the range of scenario permutations to provide acceptable confidence that system errors will be sufficiently rare to allow deployment.

Validating updates to autonomous systems poses a challenge due to the limited interrogability. Whereas for a rules-based system it would be possible to undertake a limited suite of 'regression tests' following system updates, to confirm that the intended behaviour has been introduced without adding any unwanted 'emergent behaviours', in the case of ANNs it is impossible to draw any inferences about how far-reaching the change to the output could be. Even adding a single extra sample to the training set could cause behaviours to change in completely unexpected ways.

The need to validate the safety of AI-based behaviour and control subsystems can be partially mitigated by providing an algorithmic system as a 'safety cage' that is able to over-rule the AI when it determines an output is hazardous; this potentially makes testing easier as safety becomes dependent upon a more traditional system. However, there are two major limitations that will prevent this approach from being a complete solution that removes the need to validate the AI:

1. The 'safety cage' would need to allow a wide enough tolerance to ensure the AI is driving the vehicle under normal circumstances, but a narrow enough tolerance to ensure it intervenes when required, with these requirements not necessarily being compatible. For example, setting the maximum lateral acceleration allowed by the safety cage too high would mean that the safety cage would provide no protection in low grip conditions, but setting it too low would mean excessive intervention in high grip conditions. Taken to an extreme, an overly restrictive tolerance would cause the safety cage to intervene so frequently that it effectively becomes the main control system;
2. The safety cage would need to understand what constitutes acceptable behaviour in a vast range of scenarios. Analysis of the various permutations available in even a simple scenario (e.g. pulling out from a T junction) will quickly reveal the enormity of the task in trying to define what constitutes acceptable behaviour exhaustively. The safety cage would therefore have to assume a level of complexity not dissimilar to the AI, making it challenging (if not impossible) to program and making it uninterrogable (therefore similarly challenging to test as the AI).

Furthermore, a safety cage would not provide any protection against mistakes made by the perception subsystem. The use of an algorithmic safety cage is therefore not seen as a complete solution, and testing will also have to be undertaken to demonstrate the safety of the AI itself.

1.3 The role of Safety Processes

Typical safety cases for production Advanced Driver Assistance Systems (ADAS) are currently based upon compliance with ISO 26262 (Functional Safety, i.e. ensuring the system is safe with regard to faults within the system) and ISO PAS 21448 (Safety of the Intended Function, or SOTIF, i.e. ensuring that the inherent system weaknesses and functional insufficiencies have been addressed sufficiently such that the system is safe when operating as designed).

The challenge of achieving functional safety, although significant due to the complexity of AVs and the high level of safety integrity required, is not fundamentally different to the challenges faced in applying ISO 26262 to current ADAS systems. As such, existing functional safety processes are broadly compatible with AV's, albeit with the state of the art needing to be developed in some areas, such as:

- the need for a risk matrix approach that does not depend upon controllability by a driver;
- the need to agree upon a means to demonstrate ANN robustness against faults, and;

- the need to develop approaches to 'tool qualification' that will allow validation of highly complex new tools for simulation and for AI training.

Indeed, it may be the case that a tool is never 'qualified' in its entirety, with use being conditional on validation within the specific use cases and ranges of parameter values required, further validation being needed before the tool can be relied upon outside the previously-validated envelope.

SOTIF is arguably the more challenging aspect for AV's; indeed, ISO PAS 21448 is specifically limited to systems of SAE level 2 or lower, so further development will be needed to allow application to higher SAE levels. The extreme complexity of both AVs and their Operational Design Domains (ODDs) means that understanding the full range of functionality required for safe operation becomes an intractable task. There will always be residual risk due to error-triggering scenarios that are yet to be identified, and also due to scenarios have been identified as capable of triggering errors but deemed acceptable due to low exposure and due to the technical difficulty and cost of eliminating the risk altogether.

SOTIF requires extensive test data to be collected to reach a level of statistical confidence that the number of hazards that remain undiscovered is acceptably low and that the assumptions made about the acceptability of known residual risk are accurate. This report directly supports both these goals by presenting methods to make comprehensive interrogation of the vehicle's functionality feasible and methods to collect and share data on what scenario permutations an AV will need to be able to respond to within a given ODD.

It is tempting to think that such testing primarily serves to ensure SOTIF, which is understandable given that trials of AV prototypes typically require far more safety driver interventions due to SOTIF issues than due to failures (i.e. Functional Safety issues). However, it must be borne in mind that for AV prototypes to approach a level of robustness appropriate for commercial deployment, the frequency of errors would have to be exceptionally low. As such, it must be presumed that that the number of SOTIF-related issues would be of a similar magnitude to the frequency of functional safety-related issues in current production vehicles. As such, the test processes discussed here should not be seen as exclusively testing SOTIF, as the testing will also be capable of uncovering Functional Safety issues such as software bugs and hardware malfunctions that are only exposed in particular scenario permutations.

2. Development of Test Cases

2.1 Scenario-Based Testing Overview

One method of validating an AV involves accumulating large mileages with the expectation that the system will randomly encounter a representative range of scenarios, thereby providing acceptable confidence that the system would perform safely when a fleet is subjected to a far larger total mileage in service. Performance metrics such as frequency of fatalities, injuries, accident damage, near misses or traffic rule violations can be used as a proxy for the overall safety, allowing comparison to minimum acceptable levels to be made, but it has been shown (RAND, 2016) that extremely high test mileages would be required to demonstrate an autonomous system is safer than existing manually driven fleets.

Whilst such mileage accumulation holds an appeal due to the simplicity of the concept and the level of assurance that would be provided, in practice such testing would take far too long to be feasible, even with an extensive fleet of vehicles testing concurrently, and would also be prohibitively expensive. Such an approach would be unprecedented from a regulatory perspective; existing regulations and standards focus upon ensuring due diligence is taken in the development process and upon performing a limited suite of specific tests covering the most prevalent accident types recorded, not upon demonstrating acceptable accident rates in extended real-life testing prior to commercial deployment. It is not feasible to accumulate sufficient real-world safety data for a pre-deployment transport system to allow a statistically valid comparison to be made with other transport systems, which is why this metric has never been used for regulatory approval in any transport sector.

An alternative acceptance method for AVs is to identify a representative set of parameters defining scenarios (e.g. lane width, corner radius, weather, behaviour of other vehicles), and the ranges within which the parameters can be varied, to put together a test programme that provides coverage of every permutation. This has the potential to expose the system to the same breadth of scenarios, but with reduced mileage as the test volume would not be skewed towards repetitive and uneventful driving in non-challenging scenarios. This approach is referred to as ‘Scenario-Based Testing’ (Daimler et al, 2019).

Care must be taken when attempting to use results from scenario-based testing to infer accident rates. For example, a system that performs poorly in common use cases but well in rare edge cases would be rated disproportionately favourably if test scenarios were evenly distributed, but may be found to produce unacceptable accident rates in service, where the balance of scenarios would be strongly skewed to the more common use cases. This limitation can be mitigated by adopting a hybrid approach where there is a partial bias towards the distribution in the real world, i.e. ‘normal’ scenarios would be more densely sampled, but not as densely as they would be if the balance was the same as real life (Koopman, 2019).

It is likely that regulators will require some form of physical testing for validation of the system itself, as opposed to the simulation, although this is an area where there is considerable uncertainty and debate at the current time (Law Commission, 2019). However, it is anticipated that the bulk of scenarios would be performed in simulation, thereby allowing a wider range of permutations to be tested than is physically possible upon a proving ground, whilst also allowing significant reductions in time and cost. Physical testing would be used to validate the simulation across a wide range of scenario permutations and to provide an overcheck that the complete vehicle can demonstrate basic competencies in the real world when all subsystems have been integrated.

2.2 How the Process was Applied

To facilitate effective scenario-based testing, HORIBA MIRA developed a methodology that takes as an input the use cases and/or high-level requirements for the system, and outputs test cases. This methodology draws upon the terminology that was developed within the PEGASUS Project (2017), and uses three 'levels' of scenario, progressively developed to incorporate more specific detail. This is summarised in Figure 1 and described in the following subsections.

Scenario Type	Description	Level of Abstraction	Number of Scenarios
Functional Scenarios	Scenarios described using natural language/ simple diagrams	High level	One per scenario type
Logical Scenarios	Parameters that define scenario identified (e.g. lane width), together with ranges	Medium level	Generally one per scenario type, can be split into multiple if needed
Concrete Scenarios	Specific values selected for each parameter to define specific permutation of scenario	Low level	Multiple concrete scenarios for each logical scenario (needed to sample range of parameter permutations)

Figure 1: Summary of the three levels of scenarios used to generate test cases from the system's use cases

Throughout the process, there must be full traceability to the previous stage (e.g. logical scenarios must be traceable to functional scenarios). Furthermore, it must be ensured that the final test cases and their pass criteria provide full coverage of the requirements for the system, including safety and performance requirements. This results in a complex matrix of test cases, as illustrated in Figure *2*; every possible combination of system behavioural requirements and surrounding environment permutations (as defined by the ODD) is a possible scenario, resulting in a multi-dimensional 'problem space' to investigate.

2.3 Functional Scenarios

Functional scenarios use natural language and/or simple illustrations to provide a high-level description of the scenario being performed. This includes the road layout involved, and the manoeuvres being performed by the AV and by other actors. Every requirement for the system must covered by at least one functional scenario – requirements may be covered by more than one functional scenario, and a functional scenario may cover more than one requirement, but there should be no requirements that go uncovered, and no scenarios that are not linked to the requirements. Note that in line with established systems engineering practice, it would be expected that the requirements deck provides full coverage of every step of every use case, including alternate flows, and therefore full coverage of the requirements should be taken to indicate full coverage of the use cases.

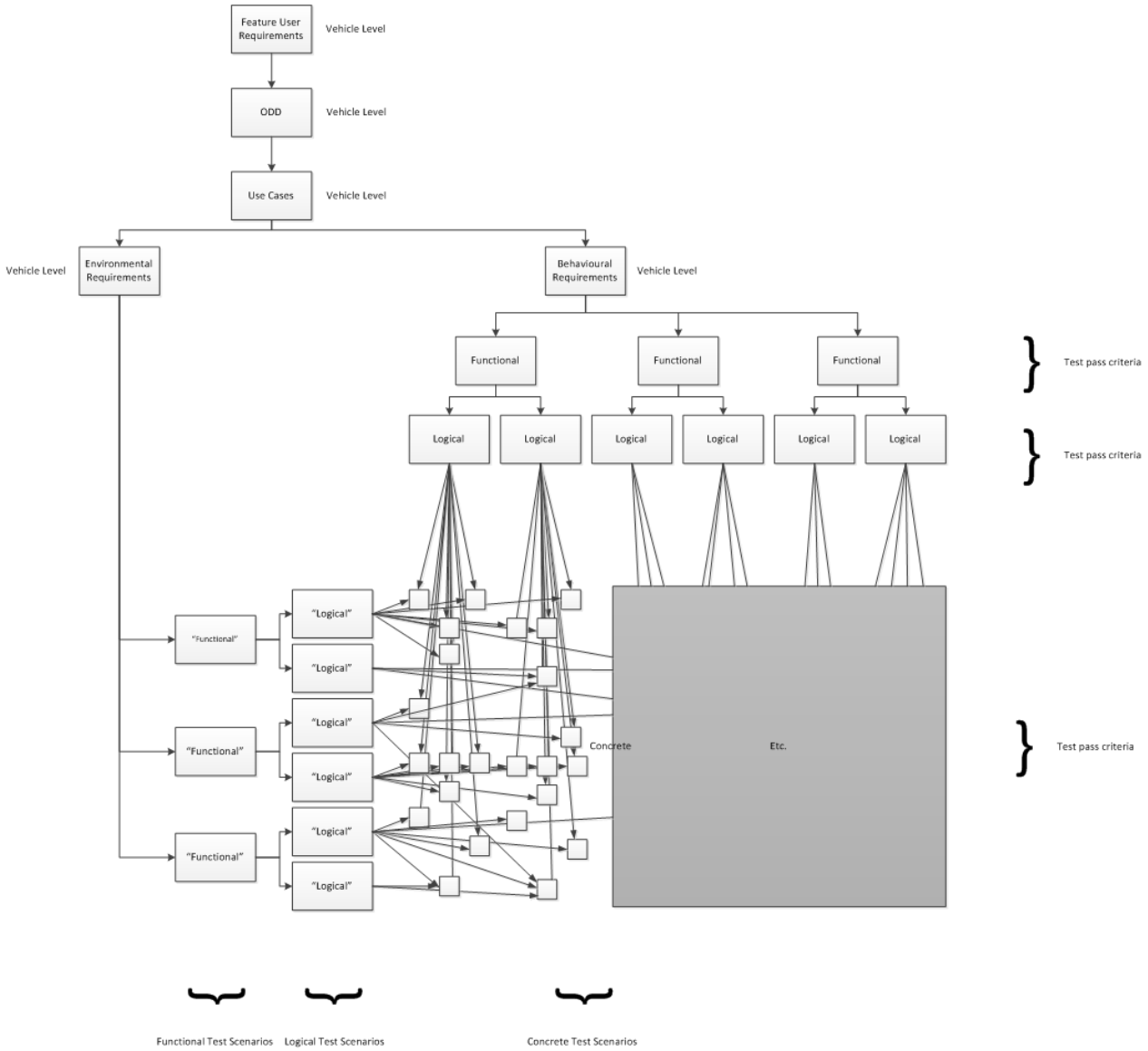


Figure 2: Complex matrix of test parameters resulting from the need for testing to cover behavioural requirements and environmental requirement

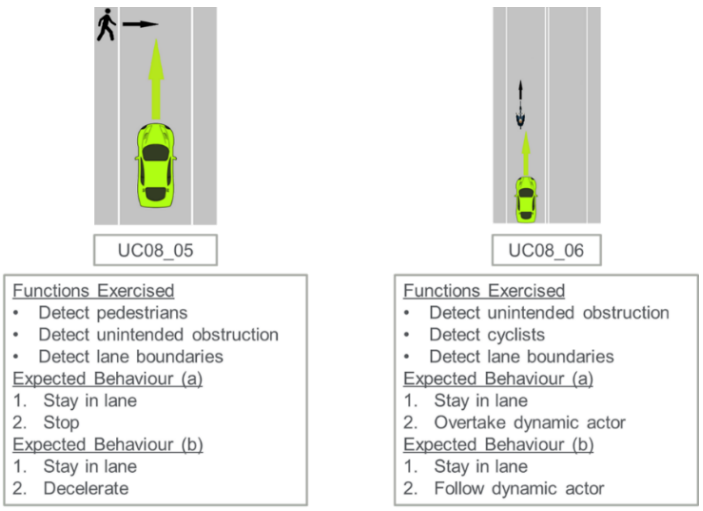


Figure 3: Examples of functional scenarios

Figure 3 illustrates two Functional Scenarios derived from the HumanDrive use cases, with a diagram to illustrate the scenario, a list of functions that the AV must perform within the scenario, and the expected behaviour (in the case of these scenarios, there are two allowable responses, a and b). This format was found to provide a good basis for discussion with other stakeholders.

2.4 Logical Scenarios

The Logical Scenarios represent an intermediate step to get from Functional to Concrete Scenarios. The aim of the logical scenarios is to identify all the parameters that could affect the behaviour of the vehicle, including the ranges that they could vary within, in preparation for selecting specific points within these ranges to create Concrete Scenarios. HORIBA MIRA developed an approach for this where **'known parameters'** (i.e. those that engineers are aware of and incorporate within the test design) are separated into two categories:

1. **Intended Parameters** – these are parameters where it is intended that the AV will change its behaviour in response to changes to the parameter value. For example, in UC08_06 (Figure 3), the AV is legally permitted to cross the solid white lines to overtake the cyclist if the cyclist is travelling at 10mph or less, but must not cross the white line otherwise. Speed is therefore an intended parameter, as is lane marking type;
2. **Unintended Parameters** – these are parameters that are not intended to influence the AV behaviour, but could reasonably be expected to. An example of this would be degraded lane markings in UC08_06, which ideally would have no effect, but in practice may cause the system to misclassify the line as being broken, resulting in a change in behaviour.

In addition to this, **Unknown Parameters** are parameters that either hadn't been identified as a possible variable (unknown unknowns, a problem that chapter 3.3 seeks to address), or that have been identified but hadn't been suspected to be capable of affecting vehicle behaviour. If any of the unknown parameters are discovered (i.e. become known parameters) in the course of testing, and the logical scenarios, concrete scenarios and test cases will need to be updated accordingly.

Two logical scenarios from the project are shown in Figure 4, where intended and unintended parameters can be seen, subdivided according to whether they are an attribute of the road layout, a static obstacle, a dynamic obstacle or part of the environment. Some parameters have a continuous range specified, whereas others have discrete variables (e.g. the type of target car is limited to two specific soft target vehicle types). Parameter values marked with a '?' denote that the ability to control the range is limited (e.g. where it will be defined by the pre-existing road characteristics at that point). A production system will require more exhaustive testing, so it will be necessary to investigate the effect of varying these parameters, requiring the use of simulation and/or a highly configurable test bed.

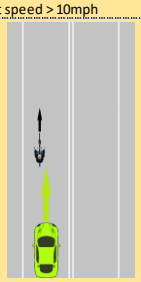
Logical Scenario ID	08_02				08_06			
Summary	Approach and overtake static car				Approach and overtake longitudinal cyclist, solid white lines			
	Intended Parameters	Range	Unintended Parameters	Range	Intended Parameters	Range	Unintended Parameters	Range
Road Layout	Prevailing speed limit	30-60 mph	Lane width Curvature Camber Gradient Lane marking condition	? ? ? ? norm/worn	Prevailing speed limit	30-60 mph	Lane width Curvature Camber Gradient	? ? ? ?
Static Obstacles			Type of car Lateral offset of POV	GST, EVT lane limits				
Dynamic Obstacles					Speed of cyclist	0-20 mph	Lateral offset of cyclist	lane limits
Environment			As measured during test				As measured during test	
Expected Behaviour	Stay in lane Overtake static actor				Stay in lane Overtake dynamic actor if cyclist speed <= 10mph Follow dynamic actor if cyclist speed > 10mph			
Illustration								
Functions Exercised	Detect unintended obstruction Detect Cars Detect lane boundaries				Detect unintended obstruction Detect cyclists Detect lane boundaries			

Figure 4: Two Logical Scenarios from HumanDrive

2.5 Concrete Scenarios

The next stage is to select specific points within the available range for each parameter in order to generate concrete scenarios. To ensure coverage of the possible permutations, it is necessary for each logical scenario to result in multiple concrete scenarios; it is not possible to understand how the system reacts to the parameters unless they are varied. As multiple parameters are being changed, rather than a single independent variable, the problem lends itself to a factorial experiment design approach.

By quantising continuous variables into discrete 'levels' within their available ranges (so that the number of levels remains finite), a 'full factorial' approach could be used where every combination of possible values is tested. However, the number of test cases rapidly escalates as the number of factors to be investigated increases, making this exhaustive approach impractical.

It is therefore necessary to take a 'fractional factorial' approach, where not every combination of parameter levels is tested, but the combinations chosen are suitably dispersed throughout the problem space such that good coverage is still obtained. Using such an approach can result in drastic reductions in the time and cost of testing, with minimal reduction in the quality of the results. The approach used within the HumanDrive scenario development process was derived from a method called 'pairwise testing'.

The use of pairwise testing within software verification has become well established as a means to capture the most prevalent defects efficiently. The matrix of test cases forms a '2nd order orthogonal array' where parameters are not just investigated in isolation (1st order effects), but also each possible pair of parameters is fully investigated (Software Testing Help, 2019).

A simple example of a second order effect is that a system may function as intended on all bend radii that are within the ODD if just tested at a medium speed, and at all speeds if just tested on a medium radius bend. However, if the two parameters are investigated together, it is entirely plausible that performance would degrade when testing at higher speeds on lower radius (i.e. tighter) bends. This is therefore a 2nd order effect, as neither parameter can cause a requirement to be violated in isolation, but two parameters being varied together can result in permutations that cause a requirement to be violated.

Higher order effects are not covered, so if three or more parameters have to be at certain combinations of values for a fault to occur, there is no guarantee that the fault will be found; it is possible that the critical combination of parameters may be included by chance, but not certain. This was deemed appropriate due to the limited timescale and budget, as interrogating higher order effects would have resulted in significantly more test cases. Another measure taken to control the number of test cases was limiting each parameter to three levels within the available range (low, medium and high). For a production AV test program where a safety driver is not available as an additional layer of protection, it is expected that far more test cases would be undertaken to ensure satisfactory coverage of the problem space, with higher order effects being interrogated and each parameter quantised into more intervals. It is also envisaged that far more parameters would need to be investigated than was feasible within an R&D trial.

4 X 3 Pairwise Matrix

Test	Parameter 1	Parameter 2	Parameter 3	Parameter 4
1	L	L	L	
2	L	M	M	
3	L	H	H	
4	M	L	M	
5	M	M	H	
6	M	H	L	
7	H	L	H	
8	H	M	L	
9	H	H	M	

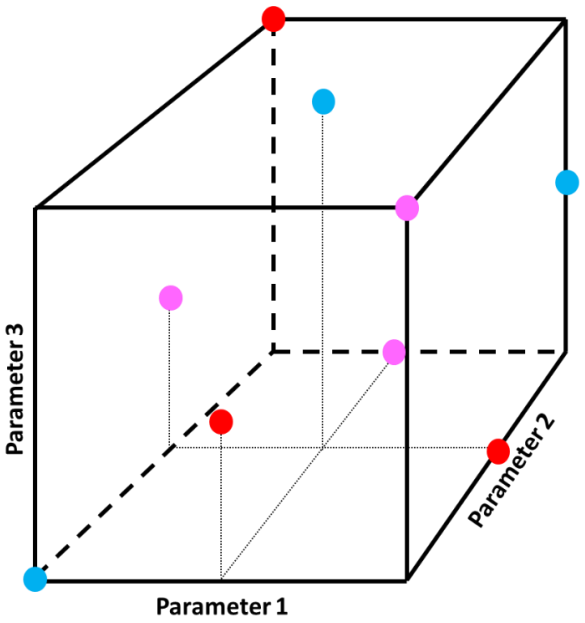


Figure 5: Test cases required using a pairwise approach if 4 parameters are investigated, each having 3 levels. Shown in table and graphical from

Figure 5 shows a 4 x 3 pairwise matrix, i.e. there are four parameters, each of which are able to assume three levels. These levels are low, medium and high for the first three parameters, but blue, purple and red have been used for the fourth parameter (the use of colours allowing illustration on three-dimensional axes). Choosing any two of the four parameters, it can easily be confirmed that each possible combination of levels for that pair is present, meaning that 2nd order effects are covered. The sampling of the problem space is illustrated on the right of the figure.

Note the good spacing of parameters; each plane has three test cases, and wherever there appears to be a gap (e.g. top-left of the front face), there are other test cases nearby (in this case, in the middle of all three planes adjoining the top left point). The result could, however, be criticised for over-representing

extreme values, with no points in the centre, whereas in real-world driving parameters will tend to be biased to medium values. This issue is addressed later.

For comparison, Figure 6 shows the number of test cases that would be required if using a full factorial approach. Bearing in mind that each 'ball' has to be tested three times (to cover each colour representing parameter 4), 81 test cases would be required, which would have been unfeasible within the HumanDrive project.

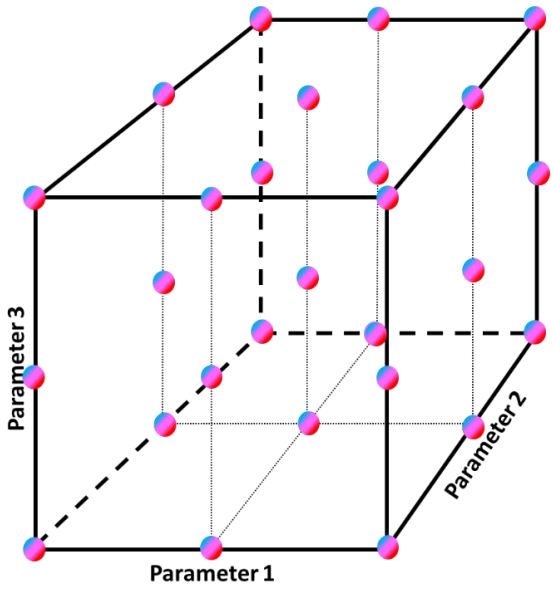


Figure 6: Illustration of the number of test cases (81) that would be required using a full factorial approach for the same problem - note that each 'ball' has 3 colours, representing 3 test cases

Prior to the application of pairwise arrays, an attempt was made to parameterise a number of Logical Scenarios by simply using judgement to attain good coverage. A retrospective comparison of the two processes was interesting; plotting both upon three-dimensional axes, it was immediately apparent that the pairwise approach resulted in a much better spread of points for sampling, and therefore better coverage of the problem space. In contrast, manual selection of parameters, despite the intention to achieve a good spread, resulted in some areas being sampled more densely at the expense of other areas not being sampled at all. Furthermore, manually selecting parameters is not scalable, preventing automation of large test programmes.

The final stage in selecting the values was to introduce randomisation within the levels defined for low, medium and high for each parameter. This allows a wider range of possible values for each parameter to be covered; otherwise, parameters would only be tested at maximum, minimum in mid-point values, with multiple repeats of these values and no coverage of the possible sub-ranges in between.

The approach also has the advantage that it makes it more difficult to optimise a system to the test as it is not possible to predict exact parameter values that will be chosen by the randomisation. Further unpredictability can be added by randomising which parameter from the logical scenarios is assigned to which column in the pairwise matrix.

Because of this randomisation, the approach developed for the project has been dubbed 'fuzzed pairwise' testing, as there is more flexibility in the values adopted relative to a strict application of the pairwise

approach. More generally, this report proposes the term 'fuzzed orthogonal array' to cover arrays that include higher orders than pairwise arrays.

Other parameters that were not included within the orthogonal array should then be assigned values randomly. Note that in some cases, the selection will be outside the control of the testing organisation, e.g. colour of another vehicle may be dictated by the availability of a suitable soft target dummy vehicle for physical testing.

A summary of the 'loose orthogonal array testing' approach developed for the project is as follows:

1. Select the highest-priority parameters (i.e. those that warrant the most detailed investigation) – this will be the 'intended parameters' and 'unintended parameters';
2. Divide each parameter range for a logical scenario into multiple 'levels' (e.g. low, med, high) – the more levels selected, the more tests will be required, so this can be used to control the volume of testing;
3. Identify suitable orthogonal array type for the number of variables and levels (e.g. pairwise matrix), and generate the matrix – again, the array selection can control the volume of testing, as covering higher order effects will require more samples;
4. Map the parameters onto the matrix (e.g. assign road width to the first column, oncoming vehicle speed to the second column etc.);
5. Optionally, include additional centre point if not already present, to improve representation of non-extreme values;
6. Randomise independent variables to points within the subrange determined for that level
    - e.g. if 'low' is less than 15mph, variable could assume any value from 0 to 15mph;
7. Randomise the noise factors not suspected capable of having an effect;
    - It may be preferable to include a bias towards values that are more prevalent in real-life;
    - Some noise factors may not be controlled due to practical limitations, e.g. weather;
8. Eliminate or modify any test points that are not feasible (impossible combinations, facility limitations). Any resulting 'gaps' need to be reviewed, and mitigated if possible (e.g. testing similar functionality in other scenario permutations, testing in simulation rather than physical world)

Pairwise testing is a well-established and popular approach to software testing, where experience has shown that the majority of defects are 1st or 2nd order, making searching for higher-order effects inefficient. However, bearing in mind the complexity and the safety-critical nature of approving production AVs where a safety driver would not be available, searching for higher order effects would be justified.

The number of orders needing to be investigated can only be determined empirically, based on experience of how many higher-order effects are found. This therefore requires testing in the early years of CAV development to be conservative and search more orders than may be necessary, until data exists to justify being able to reduce the test volume. Further consideration of how to optimise the sampling of the problem space, including the use of higher-order orthogonal arrays and of alternative methods such as Latin Hypercubes, is included within Appendix 1.

The final stage in the test case development process is to assign each concrete scenario to specific locations and specific test apparatus, and to include success criteria. As full traceability should be maintained throughout the process, it will be possible to directly infer the acceptance criteria from the requirements.

3. Wider Considerations for Future Testing

3.1 'Divide and Conquer' Approach to Testing

Sub-Dividing the Problem

Enhanced test efficiency could be gained by applying a 'divide and conquer' approach to validating AVs, such that the type of testing selected (e.g. software-in-the-loop, proving ground verification) is chosen to be most appropriate for the particular layer of the AV control system that is under test (e.g. perception layer, actuation layer). Each layer would primarily be tested in isolation, although integration testing would also be needed to provide assurance that there are no adverse effects introduced when the layers are combined. A summary of the approach can be seen in Figure 7.

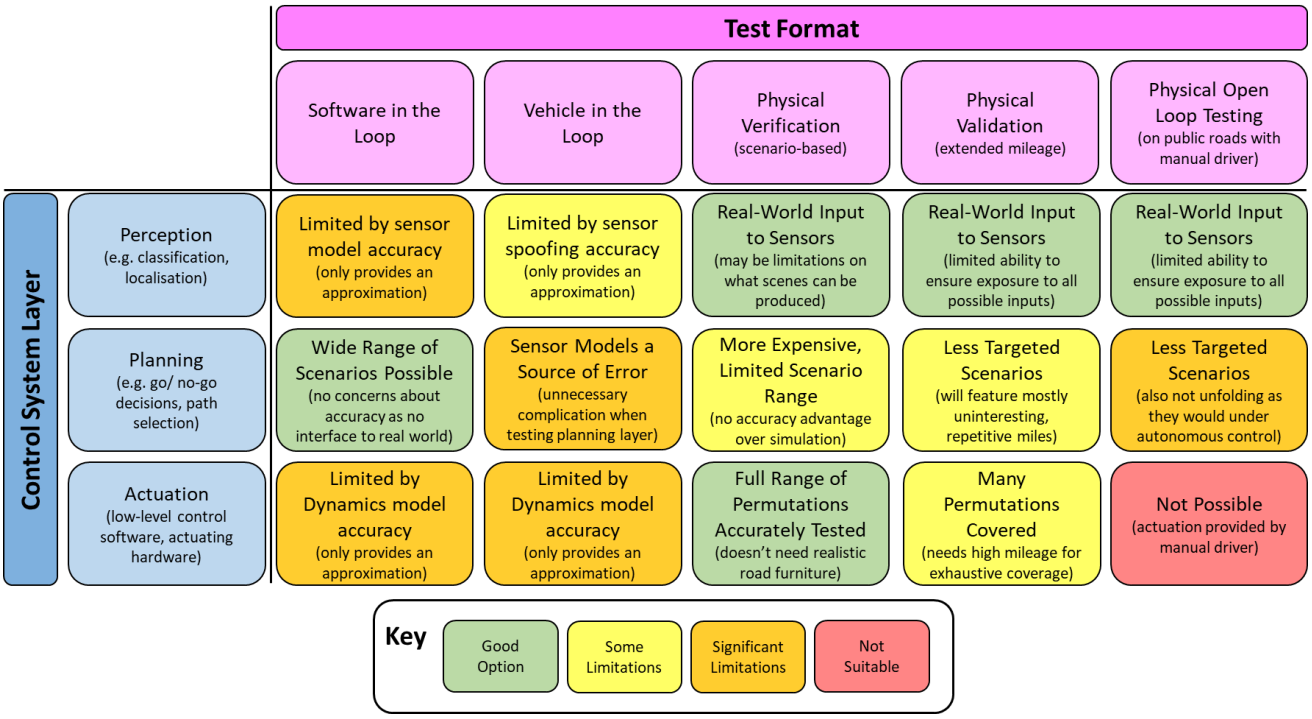


Figure 7: Relative merits of different test formats, mapped against different layers within the AV control system

For maximum control of the inputs presented to that layer, it would be preferable to be able to tap directly into the data being passed between layers at the interfaces. However, it would also be possible to set up scenarios up that are designed to work with the whole stack as an integrated system whilst specifically targeting the functionality of one layer, e.g. by providing inputs to the perception layer that are simple enough that the signal passed to the planning layer is predictable. Working with the full system in this way would introduce some error from other layers (for example, the perception layer may not estimate relative distances or speeds with perfect accuracy).

The 'divide and conquer' approach advocated here should be seen as consistent with, and building upon, the approach advocated by Amersbach and Winner (2019).

Testing the Perception Layer

For the perception layer of the control system, realism of the inputs to sensors is crucial, and this is a significant concern with software-in-the-loop testing (SIL) as the accuracy is dependent upon models of both the physical environment (including permutations such as fog and rain, which are challenging to simulate accurately) and the sensors. Beyond the fundamental challenge of simulating the physics of a sensor, there would be limited incentive for a Tier 1 supplier to want to incorporate the limitations of their product into any model they create, and questions as to how well a neutral 3rd party will be able to understand the inner workings of the sensor. This means that the accuracy of sensor simulation can never be assumed to be perfect.

Simulation testing can incorporate various different levels of hardware, referred to as 'hardware-in-the-loop' (HIL) testing'. The ultimate level of hardware possible to incorporate into a simulation is 'vehicle-in-the-loop' (VIL) testing, shown in the second column of Figure 7, where an entire vehicle is subjected to testing in the virtual world. Testing with real sensors receiving a simulated input removes concerns about the realism of the sensor simulation, but at the cost of needing to develop and validate the ability to spoof sensors using simulated inputs.

In practice, there is a continuum from pure SIL testing to full VIL testing, with variation in the amount of hardware elements included in the loop and also the maturity of those elements (i.e. how representative of the production version they are). For clarity, Figure 7 shows just the two extremes of the continuum, but it is anticipated that production test programs would incorporate progressively more hardware elements as they become available and the task of integrating the subsystems progresses.

Ultimately, it is impossible to achieve perfect realism in a virtual world, and it will therefore be necessary to include extensive real-world testing of the perception layer within the test programme. Evidence of acceptable performance can be gained through scenario-based verification testing on a proving ground (column 3) or through extended-mileage validation on public roads (column 4), but in many cases the most convenient and lowest-cost option will be to perform open loop testing (column 5). This could involve either manually driving the AV with the system performing perception tasks in the background, or using a manual vehicle fitted with the same sensor suite as the AV to collect data to use in post-processing. The data collected would also have value in helping to identify new edge cases to include in the scenario-based testing (i.e. uncovering 'unknown unknowns').

Labelling data to form that ground truth to assess perception systems against will be a significant challenge. This could require the development and validation of an automated process or require extensive manual work, but perhaps a more feasible approach would be a hybrid solution where human oversight is used when the automated process has low confidence or when it produces a different result to the AV perception layer under test. It is worth noting that open loop testing in the real world would also typically be used to collect training data for a CNN used to classify objects within the perception layer. As the training and testing process for the CNN would normally be iterative, much of the data collected from the sensors during open loop testing would also be suitable to use as training data for future iterations.

As it will not be possible to cover all scenario permutations in all test environments, it may be necessary to cover some challenges to the perception layer solely in the virtual world and others solely in the physical world. Scenarios that can be performed in both real life and in simulation would allow validation of the simulation to enhance confidence in the accuracy of results of all simulations.

Testing the Planning Layer

The 'planning' layer of the AV control system incorporates such functionality as deciding what manoeuvres the vehicle should undertake (e.g. go/no-go decisions on performing a lane change or pulling out from a junction) and the planning of a suitable trajectory and speed profile for the manoeuvre. Depending on the architecture of a particular AV, these functionalities may be provided by a single subsystem or multiple subsystems, the latter being able to be tested as a whole or further 'divided and conquered'.

Simulation is the most effective option for testing the planning layer as it allows an infinite range of scenarios to be tested, including geometries that cannot be found on a proving ground. In addition, simulation testing can be done at significantly lower cost and in less time that physical testing.

Being sandwiched between the perception and actuation layers, the planning layer is isolated from the physical world, meaning that inaccuracies within physics models do not come into play. For example, a set of inputs to the planning layer (e.g. some form of map with an occupancy grid to show the location of dynamic obstacles) in the virtual domain would be indistinguishable from the same set of inputs received in response to sensor outputs in the real world. As such, simulation testing of the planning layer can be far less computationally intensive than simulation of layers that require the complex physics of the real world to be replicated.

On the other hand, VIL testing of the planning system is a less favourable option, as testing the planning layer in isolation while the full vehicle is sat in a VIL rig makes little economic sense. However, simulation testing should use the actual processing hardware from the planning layer where possible (i.e. HIL testing rather than SIL testing). This will avoid inaccuracy due to hardware limitations (e.g. latency), and will also allow some validation of the hardware (albeit in laboratory conditions – testing in more extreme conditions would be needed to demonstrate production-level robustness). A typical approach would be to start off with SIL, and then progressively add hardware as representative subsystems become available.

Physical testing for the planning layer has similar limitations to that discussed for VIL, with a higher cost and less ability to control the inputs to the planning layer relative to SIL. In the case of physical testing, the range of possible scenarios will also be limited as not all scenarios will be able to be physically recreated using available facilities and without compromising safety.

Open loop testing of the system, with a manual driver on public roads, is also an option, as the system can run in the background, outputting the intended path at each timestep. This has the same disadvantage as extended-mileage validation in that it is difficult to control what inputs are received, resulting in many miles of repetitive and uninteresting data. Furthermore, it would not be possible to see how scenarios unfold over time as the open loop system has no control over where the vehicle will be at the next time step, and will therefore have to keep recalculating paths from a hypothetical position that it may never have found itself in if operating autonomously. It would therefore be difficult to evaluate how the system would behave when deployed.

Testing the Actuation Layer

For the actuation layer of the AV control system, simulation has significant limitations as the path adopted by the vehicle will depend upon the accuracy of the vehicle dynamics model and the models of the actuators controlling steering, braking and acceleration. There is likely to be value in using simulation early in the development process to set an initial baseline for the low-level control algorithms quickly and cheaply, which can then be refined, calibrated and later verified by physical testing. It is worth noting that

vehicle dynamics simulation is an established field that is well supported by specialist tools, making state-of-the-art simulation relatively accessible.

Proving ground testing has a particular advantage for the actuation layer as a wide-open area of tarmac can be used to compare the input (the desired path sent by the planning layer) to the output (the path actually taken). There is no need for the scene to appear realistic as the perception sensors are not involved, so any radius of curve, for example, can be tested without the need to apply white lines, add kerbs etc. Furthermore, proving grounds often feature challenging road geometries such as off-camber bends, extremely high crowns on the road etc., which can validate the actuation layer's ability to adapt to extreme outliers.

This testing can be further supported by extended-mileage validation, which will provide a wide range of permutations, but also feature a significant volume of repetitive and uninteresting data. As extended mileage validation would by necessity include the entire control system, it would be necessary to record the outputs from the previous layers in the stack, as the actuation system can only be validated by comparing its outputs to its inputs; an undesirable path could be caused by any of the layers, and cannot be assumed to be due to the actuation layer.

Open loop testing of the actuation layer is not possible, as the driver will be manually providing control inputs rather than the system. However, there may be some rare exceptions to this where it is desirable to test the low-level control algorithms in isolation, without this being connected to the actuators themselves; this could be viewed as another level of 'divide-and-conquer' subdivision within the actuation layer.

Advantages of Subdivision

There are three key advantages in subdividing testing of the system in this way:

1. **The method of testing can be optimised to the layer of the control system stack that is being tested, for maximum efficiency**, as was shown in Figure 7.
2. **The total number of permutations that would need to be tested is much lower**. For example, if we take a hypothetical system operating in an (extremely limited) ODD where there are 20 different things that need to be perceived (e.g. roundabout, pedestrian), 10 different manoeuvres that can be performed, and 15 different paths that the actuators can be required to follow, if the system is tested in its entirety, the number of permutations to test is 20 x 10 x 15 = 3000. However, if the possible combinations for each layer are tested in isolation, the total number of tests required is 20 + 10 + 15 = 45, which is 1.5% of the number of test cases for whole-system testing.
3. **Less testing is required to introduce updates to the system**. For example, if an update is introduced to the planning layer, it would be feasible to validate this purely through simulation, without having to repeat any of the test cases associated with the perception and actuation layers, provided that the interfaces between the layers remain unaltered.

Whole-Vehicle Testing and Correlation

While the 'divide and conquer' approach will provide a significant reduction in the overall test burden, nonetheless it is expected that some validation testing would be undertaken using the vehicle as a whole in a real-world environment, as this would be needed as an overcheck to ensure there are no unexpected

emergent behaviours when the subsystems are integrated. Prior to testing the whole vehicle, however, the separate subsystems would need to be progressively integrated and verified, in line with traditional systems engineering practice.

It is anticipated that the whole-vehicle physical tests would be chosen to sample different points within the problem space (i.e. the possible points within the many axes of all the parameters that can be varied). This would cover as much of the ODD as is possible within the physical constraints of proving grounds and/or real roads, but take samples at a much lower density than simulation. Validation of the simulation would be achieved though duplicating tests in simulation and the real world, with the comparison not just considering whether the results are within an acceptable tolerance, but also whether they show consistent trends.

There is currently no standard for what should be accepted as a suitable level of correlation, or indeed how far the parameter values used within a simulation can stray from the previous correlation tests before re-correlation in a more similar scenario is required. This should be considered for future regulations and/or standards, and could potentially draw from other industries, e.g. aerospace.

If particularly critical areas of the problem space are identified during the simulation, such as a threshold where the vehicle switches between using braking and using swerving to avoid a hazard, or a combination of parameters where the vehicle is close to an acceptance threshold, a high level of confidence will be needed in the accuracy of the results. It may therefore be deemed necessary to replicate such scenarios in the physical domain, or at least get as close to them as the infrastructure will allow, if the critical case is far enough from one of the previous correlation points to be a concern. Again, it is difficult to say how far is ‘too far’, so an informed decision would have to be made. It may well be decided that the limit for ‘too far’ is shorter for test cases that are found to be critical, and therefore that further correlation is needed.

It is worth noting that HORIBA MIRA will have more flexibility with regard to validation of AV simulation in the future; the TIC-IT facility being built by HORIBA MIRA as part of the Zenzic network of test beds for CAVs in the UK will consist of a large open space that can be configured into a wide variety of layouts including temporary road markings and roadside furniture. This will provide the ability to replicate in the physical domain scenarios that are found to be of particular interest in the virtual domain.

Not all variables can be tested in all environments; in some cases the environment will only be able to be recreated virtually and not physically, or vice-versa. This means that the list of variables that can be tested in each environment is shorter than the overall list of variables, and that there are some scenario permutations that cannot be fully tested in any environment. For example, fog can be added to simulations but the accuracy of the modelling is limited, smoke machines to replicate fog on the proving ground would result in very different properties to real fog, and any climatic testing facilities able to create real fog would be limited in the range of road geometries available within that environment. Unexplored scenario permutations would therefore remain as a residual risk.

Where it is not possible to replicate a scenario, it may be possible to separate out elements of the underlying functionality to target some areas of the residual risk. For example, even if the effect of fog upon the sensors cannot be adequately tested, it may be possible to test some components of the response, e.g. if the planning layer is required to reduce speeds in response to reduced visibility, the response of the planning layer when a low visibility flag is injected could be explored.

3.2 Bounded Sequential Testing

The Sequential Testing Concept

Sequential testing, also known as sequential analysis or sequential hypothesis testing, is an approach where the sample size (i.e. the volume of testing required) is not fixed in advance, but instead there is a fixed target to achieve a predetermined level of statistical confidence in the results such that the testing can be terminated (referred to as the 'stopping rule'). If the system performs far better or far worse than the acceptance threshold for frequency of errors, sufficient statistical confidence to accept the system or reject the system respectively would be reached at an earlier point, relative to a marginal system. This allows testing to be terminated sooner, thereby reducing cost and time (Georgiev, 2017).

The corollary of this is that if the system performance is very close to the minimum level for acceptance, it could take an extremely long time to achieve statistical confidence that the system is above or below the acceptance threshold. Indeed, if the system performance throughout the testing was exactly equal to the threshold, the test duration would theoretically extend to infinity. In practice, therefore, this report recommends that a form of time-bounding should be applied to ensure results are available within a finite duration.

A graph illustrating the generic approach is shown in Figure 8, where testing can be terminated at the first test point that lies outside the corridor defined by the rejection (red) and acceptance (green) lines. The observed error rate crosses the rejection line, indicating acceptable confidence that the system is less robust than the threshold (i.e. that if testing were continued to infinity, the recorded failure rate would exceed the target). Whilst the failure rate remains between the two thresholds, however, there is insufficient confidence to conclude that the system is either more or less robust than required, and therefore testing should continue.

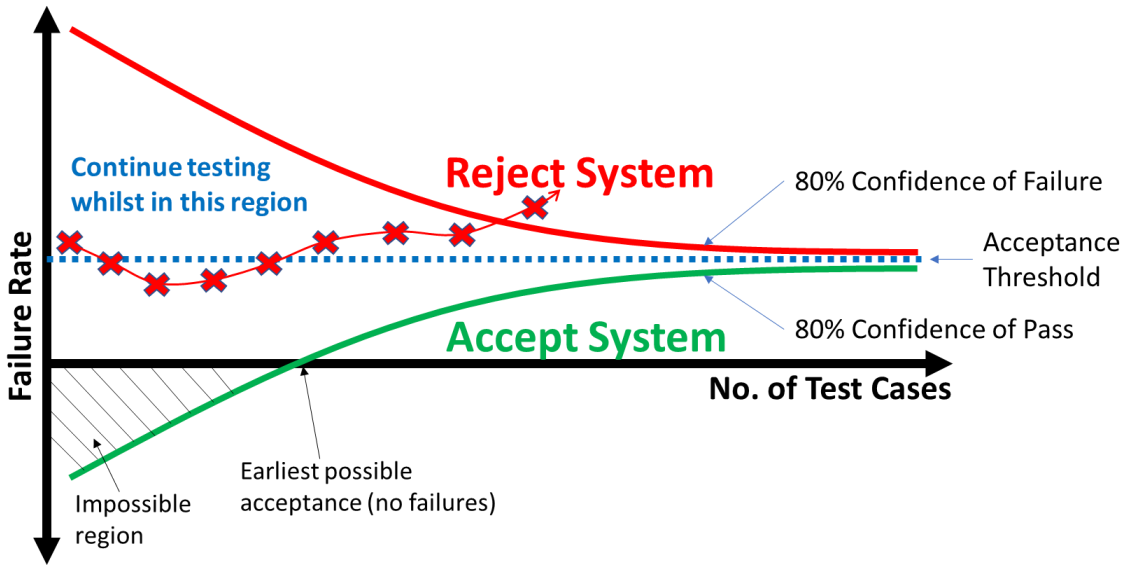


Figure 8: Illustration of sequential testing, showing the upper limit (red) being exceeded, meaning the system is rejected - if the lower threshold (green) had been crossed, the system would be accepted

Note that this central region is wider early on in the test programme, as the low number of samples mean the results have a wider error band, and therefore the recorded error rate would have to be further from

the threshold to be suitably confident of either a pass or fail. There is a region on the graph that goes to a negative failure rate, which is, of course, impossible in real life. Therefore, the first point where a pass is possible is the point where the x-axis is crossed, which would only occur if zero errors were observed when this test duration was reached; if any errors have been observed, a longer test duration would be needed to cross the green acceptance line.

In practice, for traditional testing with a pre-defined sample size, a poorly performing system would typically be rejected before completion of a test program on the basis of a 'gut feel' that it is not up to standard and that it would fail if the tests were continued, this 'stopping rule' being conceptually similar to sequential testing. The advantage of sequential testing, however, is that it allows the decision to be made on a statistical basis, thereby allowing early termination of the testing to be decided consistently, rationally and fairly.

This is especially important for a developer if the test is ended early because of exceptional performance providing confidence that the system exceeds the threshold, as it would not be acceptable to put such a product into public service on a 'gut feel' that it would have passed, and a more scientific justification would be needed to back up the decision. Similarly, if an industry regulator is to prevent a product from going on the road due to it failing to meet the threshold, a solid statistical argument would be required to allow early termination of the testing.

It should be borne in mind that in order to cease testing, it must be shown not just that the failure rate is acceptable, but also that the coverage of the problem space is sufficient; it would not be acceptable to deploy a system that hadn't been exposed to a wide enough range of problems, regardless of the performance in the tests that had been undertaken. As such, it may not be possible to stop trials at the earliest possible point identified by the stopping rule in sequential testing, as more scenarios may need to be performed until suitable coverage is achieved. In particular, it is essential that coverage includes samples spread throughout the range of possible scenario permutations, as otherwise early stopping could result in significant system limitations remaining undiscovered.

Conventional approaches to assessing confidence intervals in fixed-sample testing cannot be applied to sequential testing, and hence more complex mathematical approaches are required. This is due to the effects of 'peeking'; the more times the latest data is compared to the thresholds for accepting or rejecting the system, the more opportunities there are to observe extreme results. This concept, and approaches to mitigate it, are further examined in Appendix 2.

Adding Time-Bound Aspect

A significant disadvantage of sequential testing is the difficulty in predicting how long a test programme will last. However, by taking a time-bounded approach where a decision on pass/fail will be made after a pre-set volume of testing, assurance of the maximum duration is provided, allowing timing plans and budgets to be produced; any time and cost savings through early stopping can be seen as a bonus. Given the vast volumes of testing required to gain sufficient confidence that AV mistakes are acceptably rare, even a modest reduction in test volume could result a very significant savings.

It is advised that robustness requirements are written in a similar manner to conventional practice – this could either take the form of:

- Defining the test duration and the maximum number of incidents in the requirement (e.g. "the system shall disengage no more than 5 times in 100,000 miles of operation"), or;

- Defining requirements for the performance (e.g. "the system shall remain within 0.4m lateral distance from the lane centre") with a required robustness level (e.g. "the system shall remain within lateral limits at least 99.9% of the time") and a test duration (e.g. 1,000 miles), which would result in it being allowed to be outside tolerance for (1 - 0.999) x 1,000 = 1 mile.

This then defines the longest that the test can run for and the acceptance criteria at conclusion.

One solution if statistical confidence to allow acceptance or rejection cannot be reached prior to the maximum test duration, referred to here as option A, is to cease testing and make a decision on the traditional basis of whether the system performed better or worse than the target. This approach takes no account of statistical confidence, and it is, of course, very plausible for a system to record a pass when the true level is worse than the threshold, or vice versa. However, simple acceptance criteria of this format are widely accepted by industry, the law and the general public. Option A is shown in Figure 9, where there is no early stopping and therefore the system is rejected when the maximum duration is reached as the failure rate is (just) over the limit. Comparison with Figure 8 highlights the time-bound nature; option A effectively chops off the right-hand portion.

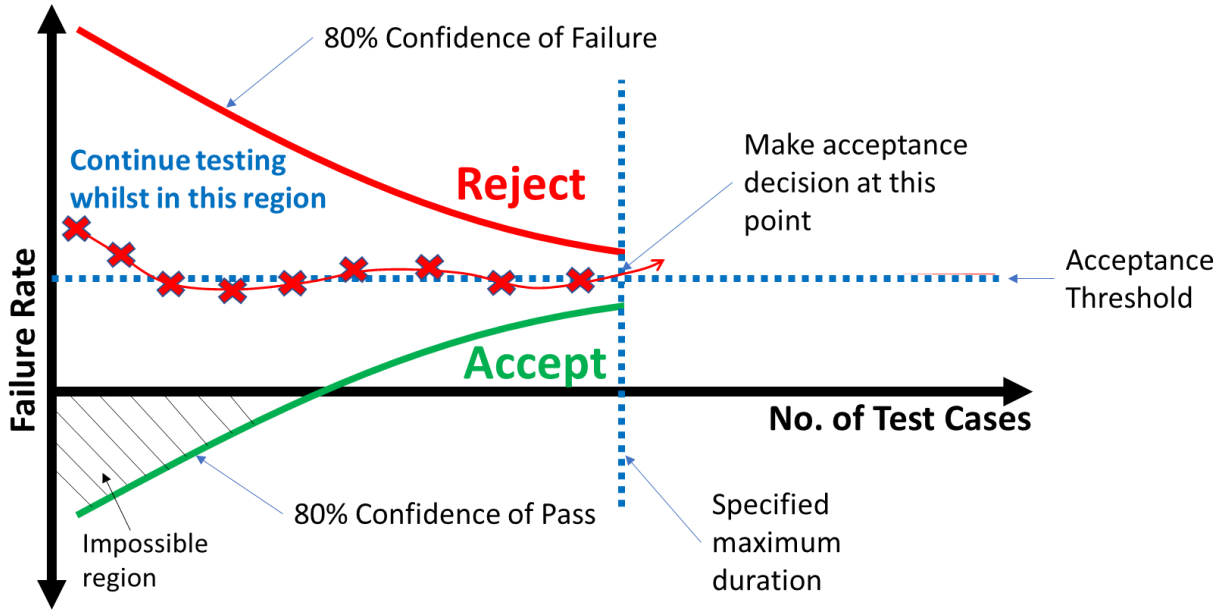


Figure 9: Bounded Sequential Testing option A - look for a confidence level that the true performance of the system is above or below the threshold, but cut the test short and make simple pass/ fail decision if maximum duration is reached

An alternative approach to bounded sequential testing is for the stopping criteria to be based upon the confidence that a pass or fail would be obtained once the defined maximum test duration is reached, as opposed to the probability that the true value for the system would be found to pass or fail if infinite test runs were completed. This approach is referred to here as option B and is shown in Figure 10. The maximum duration can be selected to be the same as that in option A (Figure 9), but as the number of samples approaches the maximum, it becomes more difficult for remaining results to cause large changes to the failure rate, and therefore the zone within which testing should continue becomes extremely narrow, making it less likely for the test programme to reach full duration.

This removes the inefficiency that can be seen in Figure 9 just before the maximum duration is reached, where it can be required that tests continue until the final duration despite it being impossible or almost impossible to cross the acceptance threshold before the maximum duration is reached. It is also worth noting that the first possible point of acceptance (where the green line intersects the x-axis) is sooner for option B. It is therefore suggested that Bounded Sequential Testing option B is likely to be the most efficient approach, reaching a satisfactory confidence level that is roughly equivalent to traditional pass/fail acceptance whilst minimising the average duration and therefore cost of the test programme.

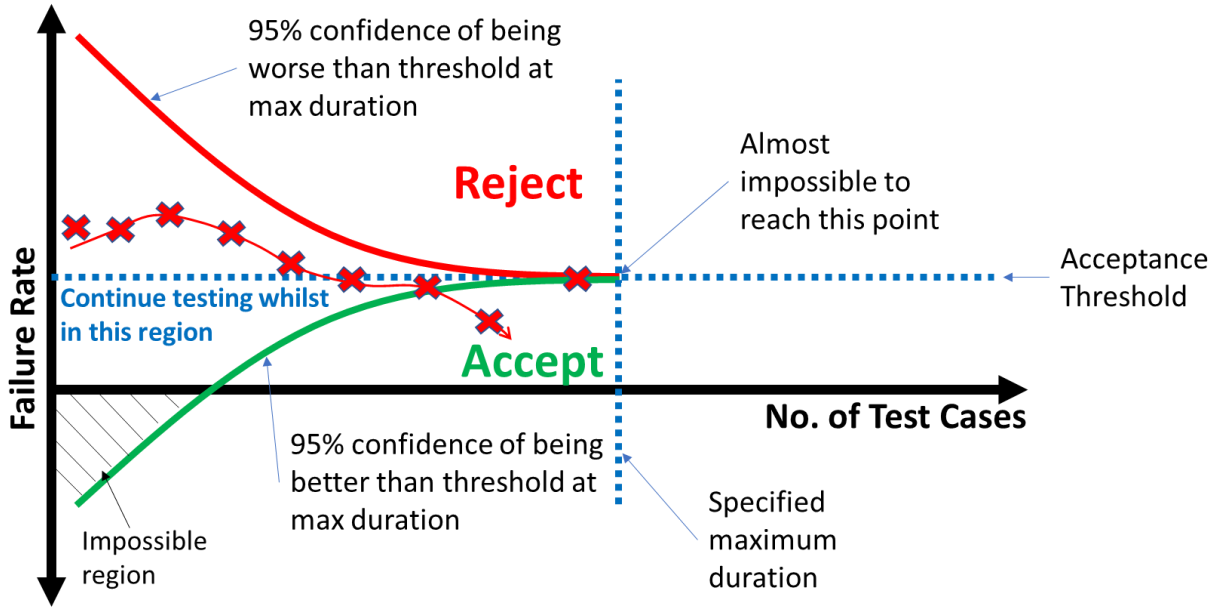


Figure 10: Bounded Sequential Testing option B - early stopping is based upon confidence of the system passing/ failing at the maximum duration, rather than the probability of the true performance being below or above the threshold

Whereas option A could be achieved using the existing approaches to sequential testing (i.e. those outlined in the appendices), with a simple pass/fail decision being made if this approach fails to result in statistical confidence, option B differs from existing sequential testing and would need significant future work to develop a mathematical approach to support the underlying philosophy described in this report.

It may be desirable to adopt a different confidence interval; whereas 80% was used in Figure 8 and Figure 9, Figure 10 suggests a 95% confidence level to allow for the fact that the failure rate at the maximum duration has its own error margin relative to the true failure rate. As is the case for traditional frequentist approaches to accepting or rejecting the null hypothesis within scientific experiments, there is no absolute rule to define what confidence level should be used as the threshold, and therefore subjective judgement is required.

Note that the confidence levels for acceptance and rejection do not have to be identical. It may be decided that asymmetric acceptance levels are required, e.g. that a higher confidence level is required to sign off a requirement as having been met than is required to reject it, due to the consequence of defects being released and causing accidents being greater than the consequences on unnecessary rework. The higher the confidence threshold, the longer it will take before the test program can be terminated early, so the desire for certainty has to be balanced against time and cost.

This report proposes the term 'bounded sequential testing' to differentiate the above approaches from traditional sequential testing, as it is a hybrid approach combining the advantage of the clarity and time-bound nature of traditional acceptance testing with the time and cost savings available due to the stopping rule in sequential testing.

Much of the prior research referenced on this topic is in relation to medical trials or to user trials to compare the effectiveness of web applications in getting users to make a certain selection ('Conversion Rate Optimisation'), the latter methodology being largely derived from the former. It is therefore recommended that further work should be undertaken, involving statisticians with expertise in the field, to develop AV-suitable methods to allow early stopping when comparing failure rates to an acceptance threshold. This should seek a solution that provides strong average savings through early stopping whilst maintaining a reasonable level of accuracy in the confidence levels recorded and a reasonable level of understandability to non-statisticians. Although this would be particularly valuable for AVs due to the volume of testing needed to demonstrate adequate performance, it is anticipated that the findings would also be applicable to acceptance testing for other types of products, both within and outside the automotive industry.

Standard practice when sequential testing is applied within medical trials is for there to be a Data Monitoring Committee who oversee the trial and make the decision about whether it is appropriate to stop after each peek (Pocock, 1992, DeMets and Lan, 1995). As such, the approaches defined above should not be seen as an automated method that provides an ultimate ruling on the early success or failure of a product, but as valuable data to assist a panel of experienced practitioners in making an informed decision, taking into account technical, statistical and ethical considerations that apply to the trial in question. Sequential analysis should not replace intelligent oversight of test programmes, but should enhance it.

3.3 Recommendation for a Test Parameter Repository

The Need for Data Sharing

It is recommended that in order to understand the full range of scenario permutations the vehicle could be exposed to, including rare edge cases, data should be collected from extensive real-world driving, whether under autonomous or manual control. As AVs operating in the same ODD will be subject to the same surrounding hazards, this work could be shared across developers, with data collected centrally either through a specific project or through pooling data collected in the course of separate trials. This would be used to inform regulations or standards with regard to what scenario permutations should be tested and what hazards should be mitigated within the engineering design process.

This is potentially compatible with the concept of a 'Digital Highway Code' (FiveAI, 2018), i.e. a set of traffic rules that are more algorithmic in nature and hence able to be enacted by a computer. In practice it will not be possible for regulators to fully define acceptable behaviour for every traffic permutation; this would be tantamount to regulators defining the control algorithms for AVs themselves, which would be unprecedented and unfeasible given the vast number of permutations possible for even relatively simple use cases. However, it would be plausible to collect and share data on what parameters should be covered by the AV design and the test programme, e.g. possible road marking permutations, sign types and positions, or plausible behaviours of other road users. This would help ensure that all permutations are considered in the design and testing.

Similarly, defining success criteria for each possible test scenario would be an intractable task. For example, pulling out from a side-road would require completely different behaviour depending upon the road layout, presence and behaviour of other road users, speed limit, weather and lighting conditions etc. Personal experience has shown that even when only the speed and position are parameterised, with other variables kept constant and only two vehicles involved, modelling what constitutes 'acceptable' behaviour for every permutation is an extremely complex task.

However, it would be plausible to use a large pool of real-world data to derive a range of thresholds to be used in an acceptance 'oracle', such as minimum allowable gaps, time-to-collision values, required deceleration rates to avoid an accident, recorded jerk levels etc. These could then be used as a guide to acceptability across a wide range of different test cases, although it is envisaged that human oversight would be needed for some scenarios where the oracle is unable to provide sufficient certainty.

The approach of pooling data on possible permutations within an ODD would provide a vital input to scenario-based testing, reducing the probability of key permutations being missed from the test plan. Without such sharing of prior learning, edge cases such as rare weather events, unclear traffic markings or unusual vehicles that were not predicted through a desk-based brainstorming approach would not be included in the test programme and would therefore remain as 'unknown unknowns'.

Such a repository would need to contain a list of known parameters that should be investigated, together with the ranges that these could be expected to take within for the particular ODD. New knowledge coming to light could cause additional parameters to be added to the list, cause the range that the values are known to be able to assume to be expanded (whether continuous ranges or a set of discreet values) or cause the applicability to spread to other ODDs. The ODD would be defined as a function of multiple attributes, e.g. speed limit, road type, junction types, allowable weather conditions, with the parameter list mirroring this.

Late changes carry a significantly higher cost than changes made earlier in the product development lifecycle, and hence one of the major goals of systems engineering is to 'left shift' the development process, i.e. to increase the investment in the early stages to ensure faults are found and corrected sooner, and therefore more quickly and cheaply. However, new safety requirements resulting from gaps found during whole-vehicle validation testing, once the system has been fully integrated (which by definition comes towards the end of the development cycle) will trigger changes very late in the development process, thereby incurring a high financial cost and significant delays. The repository would therefore be expected to result in significant efficiency savings compared to an AV ecosystem where each developer has to identify the possible scenario permutations from scratch.

It is, by nature, difficult to provide a comprehensive list of examples of what parameters the list would capture that wouldn't otherwise be captured by existing methods, as any examples given would be generated in a desk-based activity, and the whole objective is to capture permutations that engineers would be unlikely to think of in such a way. However, one example is the Uber accident in Arizona, 2018 (NTSB, 2019).

Prior to this event, there was research and development surrounding AV and ADAS responses to pedestrians and to cyclists, in a range of relative motions to the vehicle, but the author was not aware of anyone having considered the possibility of a pedestrian pushing a bicycle. This is, of course, a commonplace permutation, and most readers will have at some time been a pedestrian pushing a bicycle, but the need to classify and react appropriately to this category of road user was not, in the author's experience, an aspect of AV behaviour that was considered or tested.

When the Uber vehicle encountered the pedestrian pushing the cyclist, it was aware that an obstacle of some sort was present. However, the system struggled to classify it, changing its classification multiple times, and also struggling to predict its motion (presumably due to the system modelling future motion predications in different ways for different classifications). This culminated in the vehicle ultimately taking no action until it was too late to prevent a fatal collision. However, if large volumes of data on permutations encountered in normal driving had been collected, it seems reasonable to imagine that the permutation of a pedestrian pushing a bicycle would have been identified at some point, and if a repository existed for sharing such data, it could have been incorporated into the safety analysis and test programme for the vehicle.

How the Process would Work

An illustration of the process envisaged to compile and utilise a generic repository for information on test scenario permutations is shown in Figure 11. Field data from other vehicles is collected to form an initial version of the repository before the development of the system commences, and the section that is applicable to the system's ODD provides an input into the system safety requirements deck along with requirements derived from the Functional Safety and SOTIF analyses. These requirements are used to develop the autonomous system, which is then verified through virtual and physical means, in line with normal systems engineering practice.

However, the design of the test programme is not derived solely from information on the system itself, but also from the section of the repository that is applicable to the system's ODD. This is necessary to minimise the chance of the test plan missing a key permutation such that a system flaw is missed.

The data from incidents is analysed for all forms of testing to identify triggers for errors and allow identification of the system changes needed, so that the safety requirements can be updated; this would then lead to a new iteration of the developed system and the test programme. However, it is proposed that the real-world data should also be analysed holistically (i.e. analysing all the data, not just the rare portions where an incident occurred), as this is a source of significant information to further enhance the repository. Whilst a system developer is naturally interested primarily in the incidents that compromised their system, it is important for the benefit of the wider AV ecosystem that permutations beyond this are identified, as these could be the permutations that trigger other systems to make errors.

This would need a regulatory or financial means to incentivise examining and sharing data for the good of other companies, although it should be borne in mind that all developers will benefit if the repository is widely supported. It is also worth noting that analysis of the holistic data can have a direct benefit to the development of the particular system under test, as even if the system didn't make an error in that particular scenario permutation, if a newly discovered parameter is fed back into the scenario-based testing approach, it could be found that it does result in errors when combined with other combinations of parameters.

For example, a previously unforeseen type of road marking may not have caused an incident when first encountered on a sunny day with no traffic around, but could be found to be problematic in adverse weather, or if another road user is alongside etc. Such thorough investigation of the possible permutations is essential to ensure SOTIF is achieved within a complex and uncontrolled operating environment.

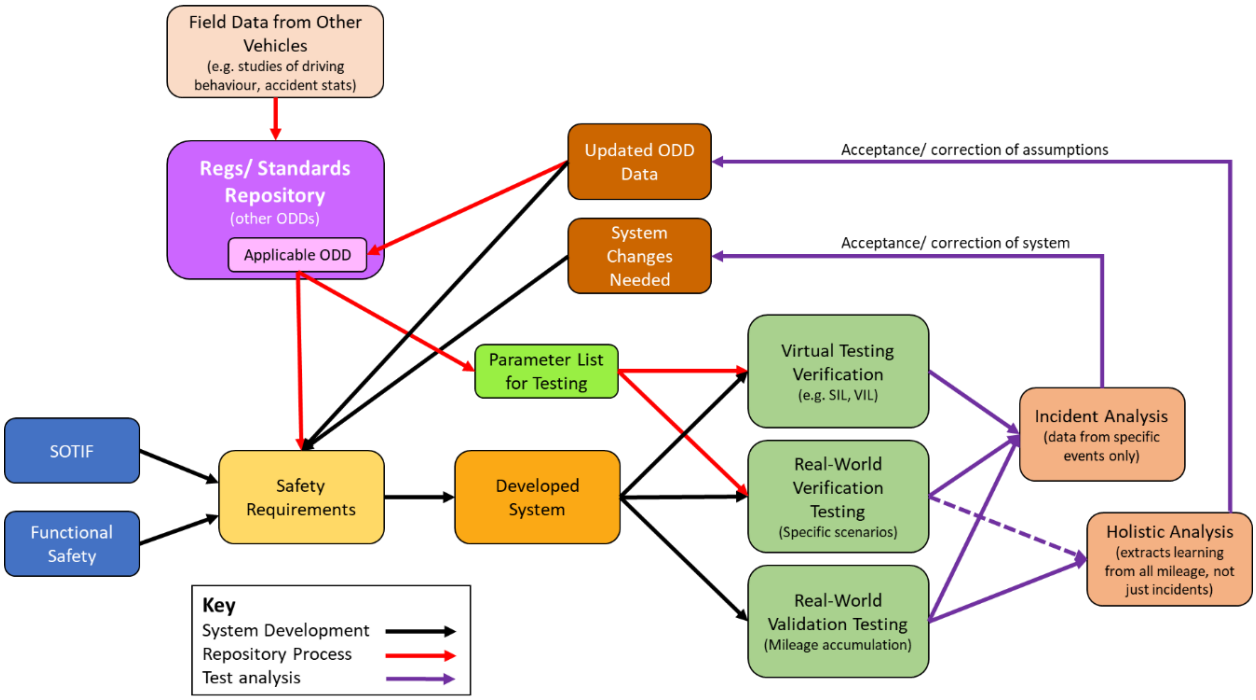


Figure 11: Proposed process for how a repository, run by regulators or a standards body, would be integrated with the system development process to make system design and testing more efficient

The arrow from the Real World Verification Testing is shown as dashed because it may be expected that limited new data will come to light through this route; the scenarios were designed according to the parameter list available prior to testing, so new permutations will only come to light if they are uncontrolled attributes of the test varying at random (e.g. a bird flying close to a sensor).

New data coming to light that expands upon or contradicts the data taken from the repository can then be used to update the safety requirements, again leading to a new iteration of the system. The new data can also influence the data held in the repository itself, or data from other trials that has been acquired in the meantime. It is anticipated that the repository would be updated periodically, perhaps on an annual basis (more frequent updates being precluded by the need to thoroughly review suggested changes with relevant stakeholders).

By comparison, if the information from the repository has not been used, it would be expected that far more errors would be discovered in the analysis phase on the right of the diagram, resulting in updated requirements and a new iteration of the cycle, with the associated time and cost penalty. It is therefore recommended that regulators and/ or standards bodies make such a repository a cornerstone of sharing best practice for safety requirement development and test case development, to enable safer systems to be developed in less time.

4. Physical Test Capability Development

EuroNCAP (European New Car Assessment Programme) is a consumer-group testing regime for vehicle safety that provides star ratings to allow purchasers to compare the relative safety of vehicles (EuroNCAP, 2019), based upon testing of the vehicle's active and passive safety systems. Current EuroNCAP active safety test protocols use robotic control of a manual vehicle (the vehicle under test, or VUT), with either no other actors (e.g. Lane Departure Warning or Lane Keep Assist testing) or one single other soft target actor (e.g. Autonomous Emergency braking for cars/ pedestrians/ cyclists, Emergency Lane Keep Assist).

Whilst HORIBA MIRA have a well-established capability to deliver such scenarios, which are sufficient to test whether an expected response is given to a particular input and which form the state-of-the-art for ADAS testing, AVs will need to respond to complex scenarios with multiple actors moving in their vicinity in different ways, with the scenario evolving and adapting as other vehicles react to the AV's behaviour.

This ability to adapt is particularly important; when testing systems such as Autonomous Emergency Braking, the driving robot system is programmed to precisely control speed and trajectory so the scenario proceeds as specified (e.g. so that two vehicles that are required to be on a collision path meet in the intended location at the same time). However, when the VUT is driven autonomously, there is no means to exert external control over its behaviour, so other actors have to adapt to the speed and trajectory of the VUT in order to ensure the scenario proceeds as planned.

It was therefore determined that there was a need to develop an advanced test capability where multiple actors are able to interact with the VUT, with the motion of these actors able to be triggered in a variety of ways such as by the VUT reaching a particular threshold (e.g. passing a virtual 'line', reaching a certain speed, producing a certain CAN signal). This realism can be further enhanced by incorporating 'cascaded triggering' where other actors do not just react to the VUT, but also to each other, such that the behaviour of the VUT can have knock-on effects as it would in real life.

The HumanDrive project provided the opportunity to use a far wider range of triggers than that used previously for EuroNCAP tests, and to experiment with cascaded triggers between actors. This allowed HORIBA MIRA to successfully enhance our capability to emulate complex real-world scenarios whilst retaining the safety and repeatability provided by the nature of the proving ground and robotic actors. This will be valuable for testing AVs, validating AV simulation, and performing development tests on Advanced Driver Assistance Systems. An image from the testing is shown in Figure 12.

Figure 12: The Vehicle-Under-Test (blue) surrounded by autonomous actors upon the HORIBA MIRA City Circuit - two pedestrian dummies, a GST (Global Soft Target) vehicle (white), and a real vehicle with a driving robot (black)

5. Conclusion

The HumanDrive project, and also the closely related learning from the SAVVY project, allowed HORIBA MIRA to make significant developments within the field of AV test programme creation, bringing together new ideas, ideas from prior AV research and ideas from other sectors. This therefore represents progress towards the ultimate goal of being able to undertake a programme of scenario-based testing that will allow sufficient confidence to be gained in the behaviour of an AV for it to be commercially deployed in public areas.

The research also identified many areas where further work is required, and indeed it will not be possible to fully understand the challenges involved in such a test programme until it is performed at full scale; R&D trials of non-production vehicles with a safety driver always ready to take control are valuable, but cannot replicate the complexity of providing safety assurance for production AVs. As such, this report should be seen as a step towards production level test programmes, rather than an end in itself.

In the meantime, it is hoped that the concepts presented in this report will be developed further within future R&D trials of highly-automated vehicles and within the acceptance testing of ADAS on production vehicles, and as such will provide a contribution to the long push towards being able to verify and validate highly-automated road vehicles as ready for commercial deployment.

6. References

Albers, C. (2019) The problem with unadjusted multiple and sequential statistical testing. Nat Commun 10, 1921. https://www.nature.com/articles/s41467-019-09941-0

Amersbach, C., & Winner, H. (2019). *Functional decomposition - A contribution to overcome the parameter space explosion during validation of highly automated driving*. Traffic injury prevention, 20 sup1, S52-S57 .

Koopman, P (2019) *Ensuring Autonomous Vehicle Perception Safety*, Autonomous Think Tank, Gothenburg, http://users.ece.cmu.edu/~koopman/talks/1904_PerceptionSafety.pdf

Daimler et al (2019) *Safety First for Automated Driving*, https://www.daimler.com/documents/innovation/other/safety-first-for-automated-driving.pdf

DeMets D.L., Lan G. (1995) The alpha spending function approach to interim data analyses. In: Thall P.F. (eds) Recent Advances in Clinical Trial Design and Analysis. Cancer Treatment and Research, vol 75. Springer, Boston, MA

Deutsch, C and Deutsch, J (2009) Latin Hypercube Sampling with Multidimensional Uniformity, http://www.ccgalberta.com/ccgresources/report11/2009-125_improved_multivariate_sampling.pdf

EuroNCAP (2019) *Safety Assist Test Protocols*, https://www.euroncap.com/en/for-engineers/protocols/safety-assist/

FiveAI (2018) *Certification of Highly Automated Vehicles for Use on UK Roads - Creating An Industry-Wide Framework for Safety*, https://five.ai/certificationpaper

Georgiev, G.Z. (2017) Efficient A/B Testing in Conversion Rate Optimization: The AGILE Statistical Method, https://www.analytics-toolkit.com/pdf/Efficient_AB_Testing_in_Conversion_Rate_Optimization_-_The_AGILE_Statistical_Method_2017.pdf

ISO 26262 (2018) *Road vehicles — Functional safety*, https://www.iso.org

ISO PAS 21448 (2019) *Road vehicles — Safety of the Intended Functionality*, https://www.iso.org/standard/70939.html

Kumar, A., & Chakraborty, B. S. (2016). Interim analysis: A rational approach of decision making in clinical trial. Journal of advanced pharmaceutical technology & research, 7(4), 118–122. doi:10.4103/2231-4040.191414, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5052936/

Law Commission (2019) *Automated Vehicles: Analysis of Responses to the Preliminary Consultation Paper*, https://s3-eu-west-2.amazonaws.com/lawcom-prod-storage-11jsxou24uy7q/uploads/2019/06/Automated-Vehicles-Analysis-of-Responses.pdf

NTSB (2019) Preliminary Report Highway HWY18MH010, National Transportation Safety Board, https://www.ntsb.gov/investigations/AccidentReports/Reports/HWY18MH010-prelim.pdf

Pegasus Project (2017) Requirements & Conditions - Stand 4 - Scenario Description, Pegasus Symposium - How Safe Is Safe Enough?, Aachen, Germany, 2017

Pocock, S. J. (1977). Group Sequential Methods in the Design and Analysis of Clinical Trials. Biometrika, 64(2), 191-199. doi:10.2307/2335684

Pocock S. J. (1992). When to stop a clinical trial. BMJ (Clinical research ed.), 305(6847), 235–240. doi:10.1136/bmj.305.6847.235

Pocock, S. J. (2005) When (Not) to Stop a Clinical Trial for Benefit, JAMA 2005—Vol 294 No. 17, https://researchonline.lshtm.ac.uk/id/eprint/12396/1/When%20(Not)%20to%20Stop%20a%20Clinical%20Trial%20for%20Benefit.pdf

RAND (2016) Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?, https://www.rand.org/pubs/research_reports/RR1478.html

Software Testing Help (2019) *Pairwise Testing or All-Pairs Testing Tutorial with Tools and Examples*, last accessed 16/12/2019, https://www.softwaretestinghelp.com/what-is-pairwise-testing/

Wald A. (1992) Sequential Tests of Statistical Hypotheses. In: Kotz S., Johnson N.L. (eds) Breakthroughs in Statistics. Springer Series in Statistics (Perspectives in Statistics). Springer, New York, NY

Appendix 1 – Further Analysis of Sampling Methods

Coverage of n^{th} Order Effects

The complexity of highly automated systems, with many parameters interacting with each other, combined with the need for extremely thorough evidence of system robustness (as appropriate for safety-critical systems), means that it may be necessary to fully investigate 3^{rd}, 4^{th} or even higher order effects (i.e. to sample all possible combinations for groupings of 3, 4 or more parameters). However, this has to be balanced against the time and cost involved in testing; the higher the order of effect covered by an orthogonal array, the more samples will be required. As there is no way to predict the probability of each order of failure mode by analytical means, it will be necessary to collect empirical data from testing of real AVs to understand whether the probability of finding scenario permutations that trigger system errors through examining higher order effects is sufficiently high to justify the investigation.

It is anticipated that for early autonomous vehicles, a conservative approach would need to be taken, meaning that higher order effects would need examination. This will then provide data on the defects uncovered, and if it becomes apparent that higher order effects are suitably rare (or that the exposure to the permutation in service will be acceptably rare to accept the residual risk), it may be deemed acceptable to reduce the number of orders examined and the resulting test burden. However, if the opposite approach were taken and early systems only tested lower order effects, there would be no data available on whether higher order coverage would uncover defects, and hence no empirical evidence would be collected until enough serious incidents had occurred in service.

Note that all orders below the maximum selected will also be covered (i.e. a test plan covering 3^{rd} order effects would by definition also cover 2^{nd} and 1^{st} order effects), and that any higher order errors could be found by random chance depending upon the combinations of values for the other parameters chosen for each test case; a test plan designed to give full coverage of 3^{rd} order effects, and no higher, will also cover some permutations of higher order effects, but it will not provide full coverage. However, if we are accepting that a full factorial approach is impossible, we therefore have to accept that the highest order effects cannot by definition be explored exhaustively, regardless of the method used to select the samples.

Alternatives to Orthogonal Arrays

Orthogonal arrays, including pairwise arrays, are proposed as an effective method to ensure that the test cases are well spread out within the problem space and cover the lower order effects comprehensively. However, there are other approaches that should be considered, a particularly promising one being 'Latin Hypercube Sampling with Multidimensional Uniformity' (Deutsch and Deutsch, 2009).

Latin Hypercube sampling (LHS) is an evolution of Monte Carlo simulation (MCS). Whereas MCS randomises each variable, typically according to the estimated real-world distribution, and can therefore result in some areas of the problem space being over-represented and others under-represented, LHS separates out the distribution of each parameter into strata (directly equivalent to the 'levels' described in Chapter 2.5 for orthogonal arrays), with it being required that a point is selected within each strata for each parameter, albeit with the value chosen within that strata being random.

LHS therefore ensures coverage of every strata with regard to first order effects, i.e. the spread of points for any single axis is in accordance with the intended distribution. In a similar manner to MCS, the strata sizes are typically varied in accordance with the expected real-life probability distribution such that there will be more strata, and thus more samples, towards the middle of a gaussian distribution). However, equal strata sizes can be used to represent a flat probability distribution, and it may be desired to use a solution somewhere between these extremes for AV testing to ensure the most prevalent scenarios have higher coverage than the less prevalent ones but also to ensure that edge cases still get reasonable coverage, rather than the testing being heavily-skewed to repetition of 'ordinary' scenarios.

Latin Hypercube Sampling with Multidimensional Uniformity (LHS-MDU) increases the uniformity in each dimension by initially selecting multiple times the number of samples desired, achieved by dividing each parameter into more strata, then iteratively eliminating the sample point that has the lowest average distance to its two nearest neighbours until the data set is pruned such that the desired number of samples is arrived at. This has the effect of increasing the overall uniformity; whereas LHS can still result in significant under- and over- representation of some areas of the problem space, LHSMDU ensures that the spread of points throughout the problem space is more even, as points that are clumped together will be selected for elimination.

This does not guarantee full coverage of higher order effects, as some gaps can still exist, but for higher density sampling this is arguably of less concern as each strata on each axis would be relatively small, and therefore there would be another point relatively nearby.

Comparison of Fuzzed Orthogonal Arrays and LHS-MDU

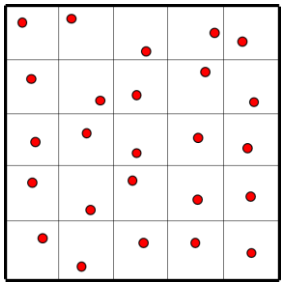


Figure 13: Example of how a 2 x 5 array would look with randomisation within each strata

Figure 13 illustrates the problem with defining the strata that a sample must lie within and then performing the randomisation, the approach taken for fuzzed orthogonal arrays; whilst some samples are relatively well spaced, it is possible for samples to exist in relatively close proximity, either side of the dividing line between strata. Although this will not be the case for most points in the array, the randomisation means that there will be a minority of samples that are in relatively close proximity to another. However, it should be noted that Figure 13 shows a particularly extreme case, as every strata is occupied - this problem would be reduced when using fractional factorial approaches as there would be less strata occupied and therefore less opportunities for close proximity

Figure 14 illustrates how opportunities for nearby points are more limited in a pairwise array. It is viewed from the perspective of the blue box at the front bottom left (i.e. all parameters at their 'low' value), and shows the four strata that contact it. Two of these, the purple and the red ones, share a single edge, whereas the two blue ones (top centre and back right) do not make contact within the three dimensions

of the geometric drawing, but do make 'contact' by virtue of being of the same colour (i.e. they share the same level in the fourth dimension). Perhaps somewhat counter-intuitively, the four adjacent strata are all equally 'close' to the front bottom left one; rearranging the illustration to show one of the fourth dimension (colour) as one of the three geometric dimensions would produce an image where the strata currently sharing a colour would instead share an edge.

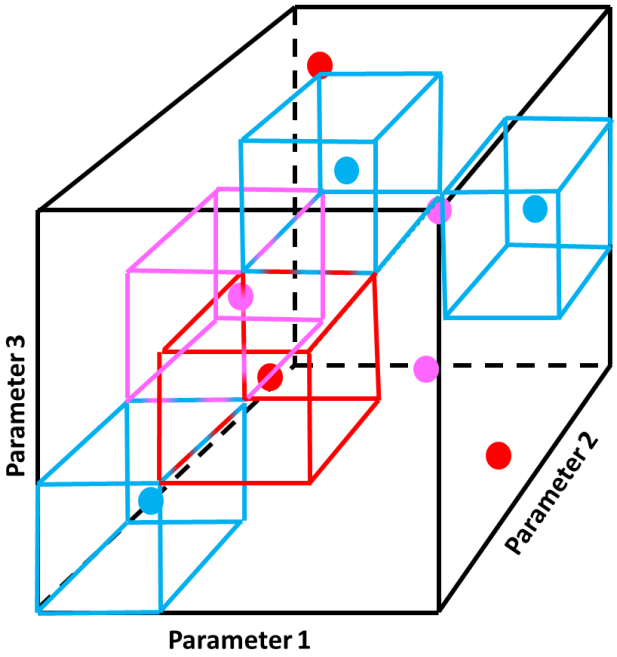


Figure 14: Illustration showing 'contact' between strata that are adjacent to the blue strata on the front bottom left (note that the other blue strata are included as they are 'adjacent' in the 4th dimension of colour)

The sparser sampling in Figure 14 means that there is a much lower chance that a point will fall close to a line that contacts another strata with a sample point, as most sample points falling near an edge will fall near an edge adjacent to an unoccupied strata. Furthermore, even points appearing to be close within the 3 geometric dimensions are a different colour, and therefore have a different value in the 4th dimension; geometrically near points in the purple box will also be near in colour (low, medium and high were defined as blue, purple and red respectively), but when the randomisation is applied within the colour 'dimension', the sample point would be relatively unlikely to assume a shade of purple that is right at the blue end of the spectrum. Therefore, even if a pair of samples did exist close to the blue and purple edge, in the majority of such instances they would still have significantly different values in the 4th dimension. As the blue and red strata lie at opposite ends of the colour dimension, there is no possibility of these sample points being close with respect to the 4th parameter.

It can therefore be seen that making the array sparser or increasing the number of parameters will lower the probability of any particular pair of points being close, and therefore over the whole array will result in a lower number of near-duplicate sample points relative to denser, lower dimension arrays. This arguably supports the use of fuzzed pairwise testing (or equivalent higher-order orthogonal arrays) for less densely sampled problem spaces, where nearby points become less of an issue but where care is needed to ensure the limited samples still cover lower order parameter interactions, but also arguably supports the use of LHSMDU where the problem space is more densely sampled, as theoretically-perfect coverage

of lower order effects becomes less important when the strata become more narrow, but the need to avoid extremely closely-spaced points becomes paramount.

It would be possible to make fuzzed orthogonal arrays less prone to near-duplicate samples by applying a method similar to the multidimensional uniformity described for Latin hypercubes; for example, multiple samples could be randomly selected within each occupied strata, and those closest to other sample points progressively deleted until there is one per strata. However, this doesn't guarantee an even spread (as for LHSMDU, the random samples can still assume similar values meaning the deletion will make little difference) and has the potential downside of reducing the probability of tests taking place near the edges of strata, distorting the spread of values for each individual parameter.

Another possible modification of Fuzzed Orthogonal Arrays, again similar to the approach used to for LHS-MDU, would be for each level to be divided into smaller intervals, with each sample taken from the overall level being assigned to a different one of the smaller intervals, to ensure the same values for that variable are not duplicated, thereby giving a better spread of values to understand 1st order effects of that variable. For example, if parameter 1 is at 'low' in three samples, it could be ensured that one of those samples is biased to the lower end of that 'low' range, another in the middle, while the other is biased towards the upper end.

Orthogonal arrays (i.e. pairwise arrays and their higher order equivalents) should not be seen as a separate concept to Latin hypercubes, as they are similar approaches derived from the same basic principle, with a range of alternative options available that span the divide. Looking at one side of the 4 parameter pairwise array shown on the right of Figure 15, it can be seen that if the depth (i.e. parameter 2) is ignored, the selection of sample points forms a Latin square (shown on the left of Figure 15), i.e. a square where each value appears once and only once in each row and column; indeed, the six faces of the cube represent 3 mutually-orthogonal Latin Squares, with opposing sides providing two alternative views of each.

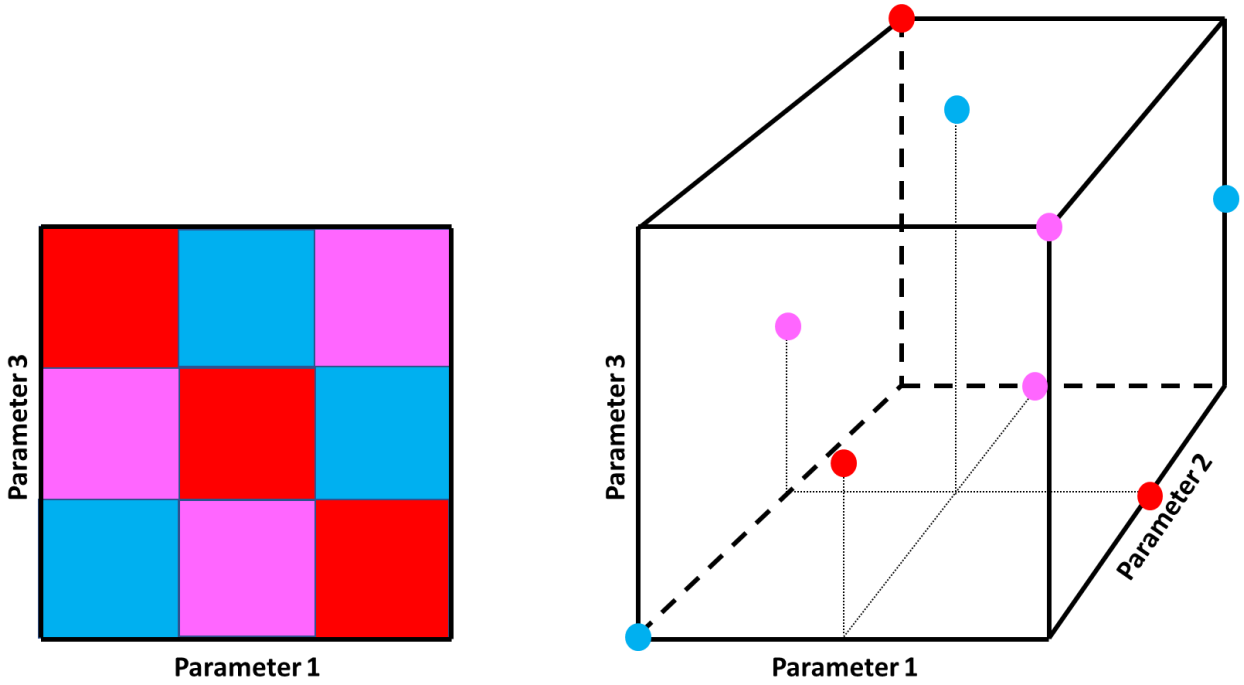


Figure 15: Looking at any one face of the orthogonal array (right), i.e. ignoring the 'depth' provided by the 3rd parameter, produces a Latin square (left)

A Latin Cube is essentially a three dimensional version of this (i.e. 3 spatial dimensions, with the colours or values at each point providing a 4th dimension); the pairwise array in Figure 15 is therefore a possible permutation of a Latin cube, although a Latin cube would not necessarily have to cover 2nd order pairings, so less ordered permutations would also be possible. A Latin Hypercube merely extends this same principle to higher dimensions (i.e. more parameters) compared to a Latin cube, making it difficult to visualise or depict, but conceptually identical.

Although Deutsch and Deutsch present hypercubes that don't ensure every possible pairing of parameters is covered, instead including every level of every parameter once and only once, it is possible to introduce nth order orthogonality within Latin cubes and hypercubes; a pairwise (2nd order) hypercube requires each level of each parameter to be repeated λ times, where λ is equal to the number of levels assigned to each parameter.

The pairwise array shown on the right of Figure 15 is an example of this, with each value of each parameter appearing 3 times, and each parameter having 3 levels. This therefore results in a more dense and more ordered matrix than would be created if 2nd order effects were not considered, or looked at from the opposite perspective, for a given level of resources allowing a given number of tests, removing the 2nd order requirement would allow more levels to be included for the same number of test points, potentially resulting in better coverage of the individual parameters (i.e. 1st order effects).

This latter issue with pairwise testing was partially mitigated in the 'fuzzed orthogonal array' approach developed for HumanDrive by randomising within the bounds of each strata such that the repeats of that level are unlikely to be identical. On the other hand, the lack of coverage of higher order effects is partially mitigated in the approach presented by Deutsch and Deutsch through the pruning of nearby points to ensure a better spread of sample points throughout the multi-dimensional problem space. As such, the approaches appear to have relatively similar merits.

Figure 16 shows how a Latin cube without orthogonality (i.e. not covering 2nd or higher order effects) will have less points for a given number of parameters and levels, when compared to a pairwise array. This saving in test burden is at the expense of very significant gaps where the problem space is not sampled, so a more representative comparison is provided by Figure 17, which shows two alternative permutations that could be produced when 9 levels per parameter are used, in order to have the same number of samples as the pairwise array. It can be seen that very poor coverage is achieved if the points align on an axis (the worst-case permutation of perfect alignment being shown on the left). The image on the right was generated using random assignment of parameter values (i.e. LHS without the MDU), and shows far better coverage than the theoretical worst-case, but some significant clumping of samples offset by some significant caps (circled in red).

Using LHS-MDU (i.e. adding in the 'multi-dimensional uniformity) results in an improvement in the spread of sample points, albeit with some clumping (of the orange, blue, pink and yellow samples), as shown on the left of Figure 18. This was achieved using the method described by Deutsch and Deutsch, i.e. by dividing the parameters into 27 strata to produce an initial 27 points (three times the number in the previous comparisons), then successively deleting the sample with the lowest average distance to the nearest two samples until the desired 9 remained. For comparison, a fuzzed pairwise array with 9 samples is shown on the right of Figure 18, created using the methodology described in Chapter 2.5. Both appear to provide similar coverage in terms of gaps within the array, although the pairwise array in this particular instance shows less clumping of nearby points, indicating slightly more efficient use of resources. As random chance

is a major factor in deciding the sample points, further work would be needed to establish whether this is a consistent trend, and problem spaces with far more dimensions and levels would need to be investigated to identify whether the complexity of the test programme affects the choice of sampling method.

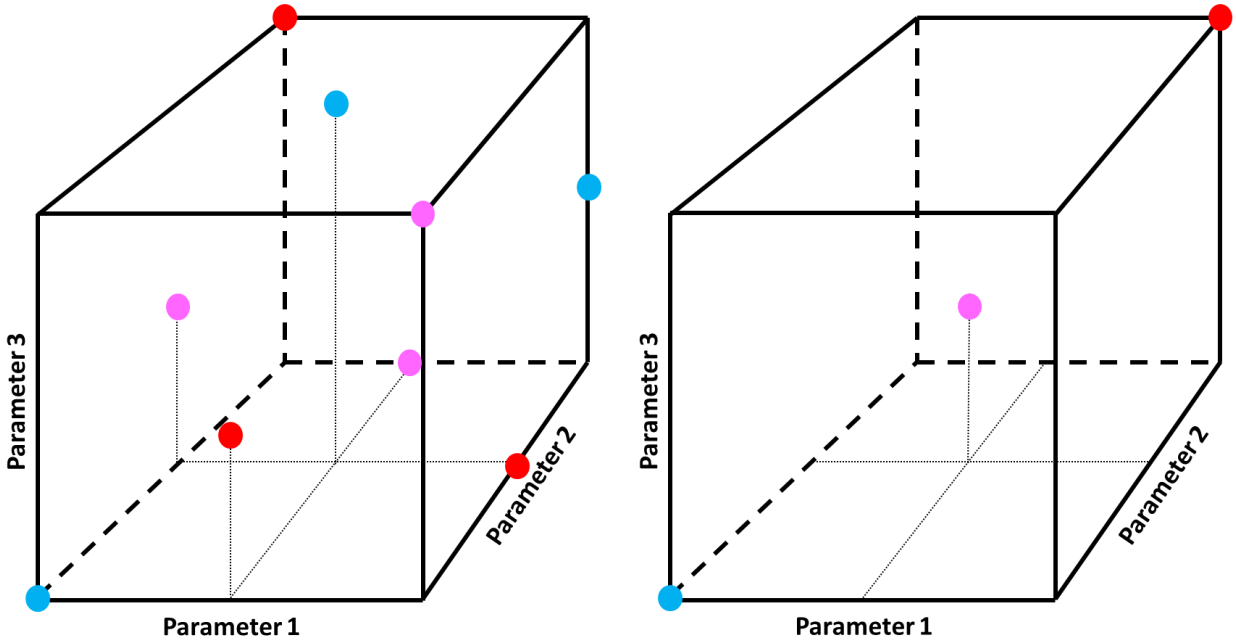


Figure 16: Latin Cube with no orthogonality to cover pairwise effects results in far less samples (right) than a pairwise array (left), but at the expense of poor coverage of much of the problem space

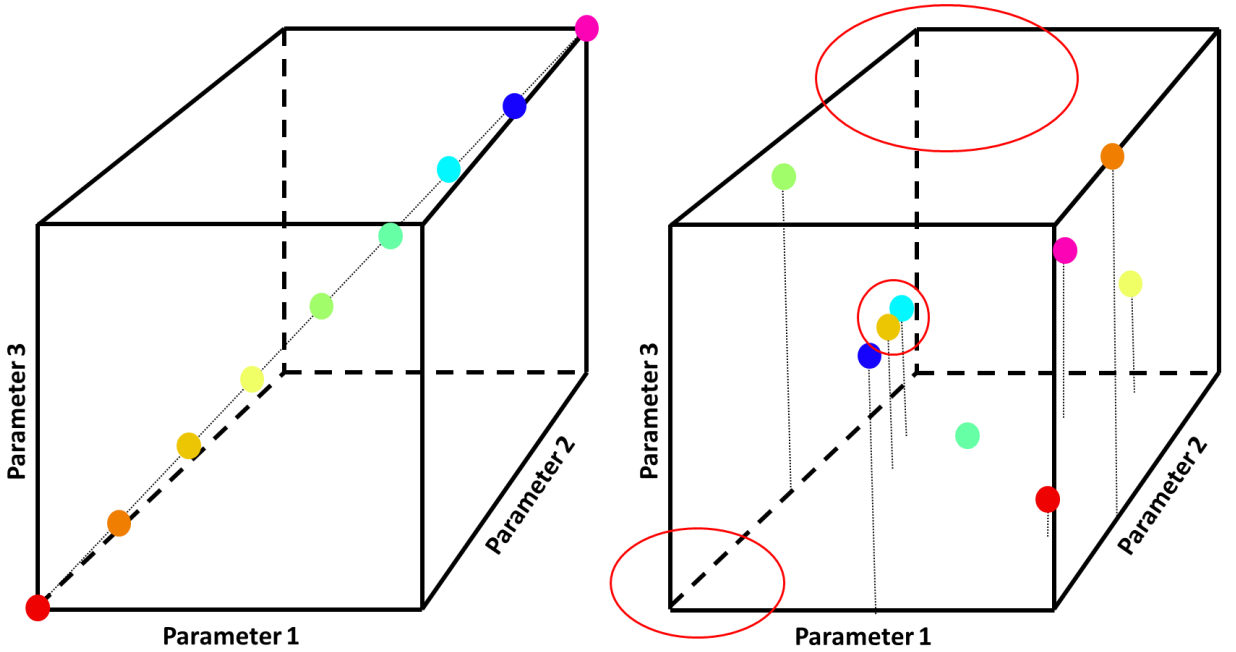


Figure 17: Increasing the number of samples in a Latin cube to match the equivalent pairwise array for fair comparison: depending on how the sample parameters are randomly arranged, it can result in poor coverage (worst-case shown left), or more even distribution (right – red ovals highlight gaps and clusters)

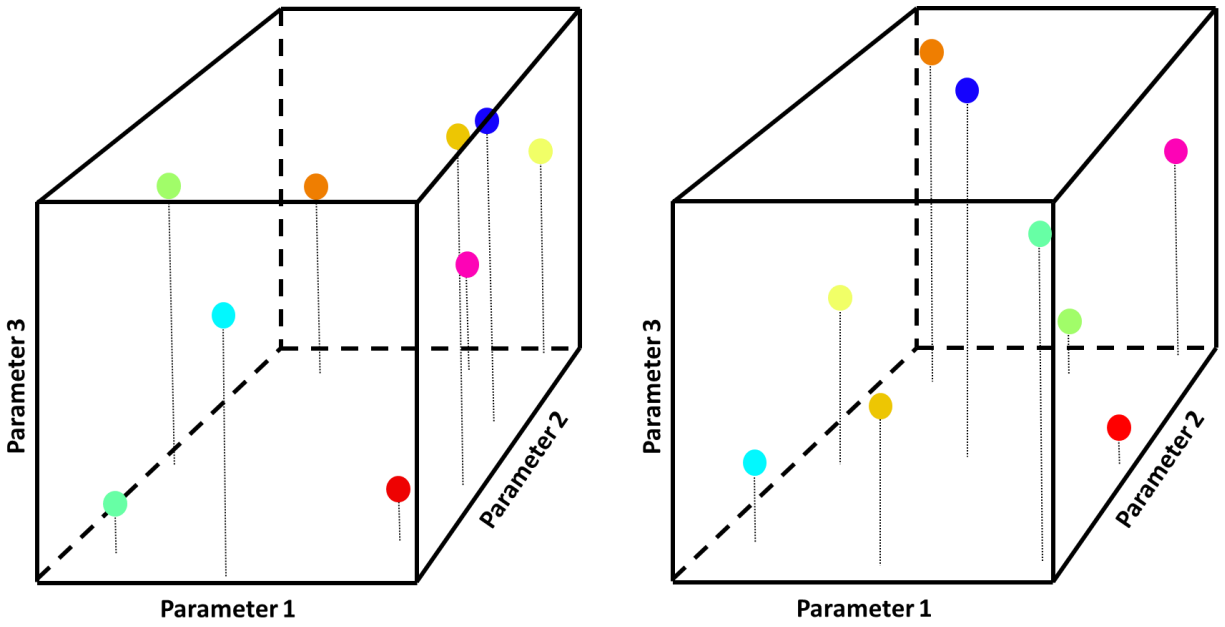


Figure 18: Applying multi-dimensional uniformity to a Latin cube produced the image on the left. By comparison, the image on the right shows a pairwise array where the sample points were then randomised within their strata. Both avoid gaps to a similar extent, although the left image includes a cluster of test points, lowering efficiency

Ultimately, practical experience of such approaches will be needed to form a consensus on the most efficient way to explore the problem space, informed by empirical data on errors captured in the test programme, and errors that are missed by the test programme and observed in service. Fuzzed orthogonal arrays and LHS-MDU appear to be tangibly superior for the purpose when compared to MCS and LHS, but further research would be needed to understand which of the former two is the best approach, or at least to validate the reasonableness of each method if it is not possible to prove one to be demonstrably superior.

It is worth noting that some parameters have a finite number of discrete values; for example, the prevailing speed limit may only be able to assume 3 values within a particular ODD. As hypercubes and pairwise arrays require an equal number of levels on each axis, it would be necessary to repeat the same value on some axes to make up the required number of strata. In some cases, it may be possible to combine two parameters into one axis – for example, two parameters that each have three discrete levels result in nine possible combinations, so if the other axes have at least nine levels, it would be possible to assign the two parameters onto a single axis.

Another area for further research, which may impact upon the sampling method, is 'search space optimisation', i.e. adapting the search to sample areas identified as being of interest to a higher density, with the initial search using the above methods being merely a means to identify areas of interest rather than a means to produce results. This could be achieved by creating 'sub-hypercube' (or equivalent sampling method) to gain higher resolution in a particular area of the problem space (Figure 19, left). However, it may not be desirable to explore a whole hypercube, as the first round of data with the sparse array may be sufficient to zoom in on specific areas of interest without having to explore wider area that would be created by a hypercube (Figure 19, right, showing a line where specific points of interest are concentrated, with sample points specifically clustered around it).

This area of interest may be where there is divergent behaviour (e.g. the crossover point is reached where the vehicle determines that it is better to steer around an object rather than brake for it, resulting in a discontinuity), where the performance fails to meet an acceptance criteria (or comes close to the threshold), or where some form of oracle assessing the quality or safety of driving records a particularly concerning result.

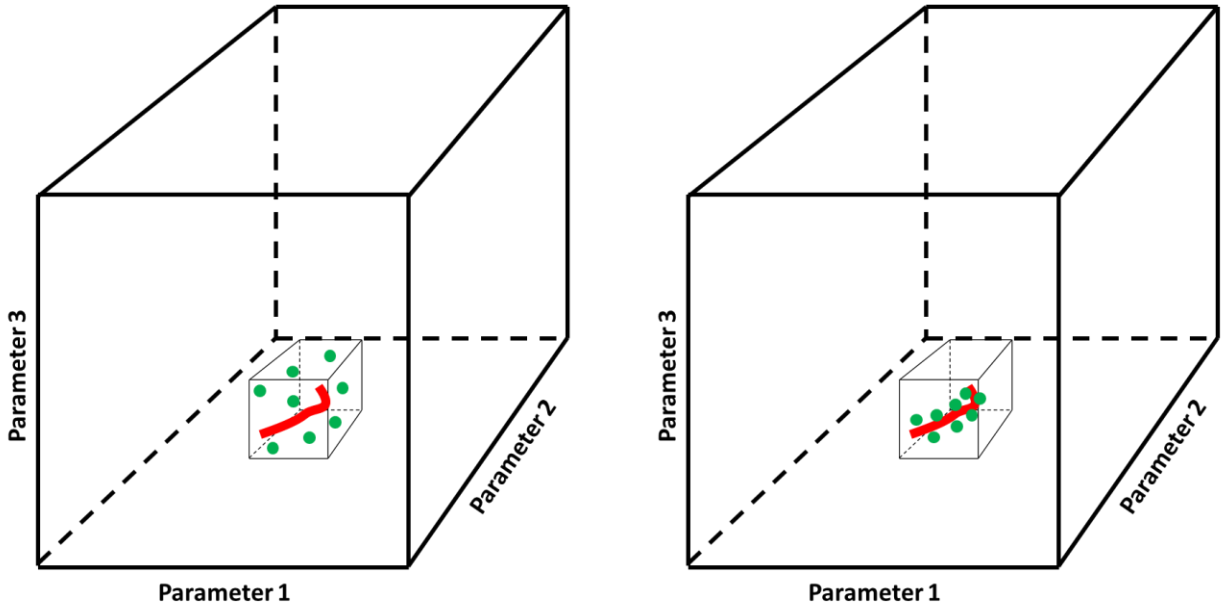


Figure 19: Taking further samples from an area interest - the sub-hypercube approach (left) provides poor coverage of the red line where the interesting threshold occurs compared to targeted sampling of more specific points (right)

However, it must also be borne in mind that a production AV test programme would include simulation and physical testing, with the fuzzed orthogonal array or LHS-MDU approach being suitable for selecting sample points for physical tests to validate the simulation. Therefore, another metric to determine an area of interest could be the delta between simulation and real-world results (or potentially the rate of change in the delta, indicating that the data is not just offset but is showing a different underlying trend).

When new samples are taken in an area of interest, the approach should always seek to re-use data collected from the initial search of the overall problems space, such that new points are not created excessively near to prior ones. This could involve exempting prior samples from deletion within an LHS-MDU pruning process, or developing an algorithm that attempts different mappings of parameters against columns in an orthogonal array to find the permutation that results in the most required samples being already covered by prior ones.

Search space optimisation is an area of future research that HORIBA MIRA are undertaking within the VeriCAV project, another collaborative R&D consortium in receipt of funding from Innovate UK, and will therefore not be further examined here.

Appendix 2 – Mitigation of the Effects of 'Peeking' in Sequential Testing

The mathematical analysis of when to stop sequential tests early is made more complex by the effects of 'peeking'. Traditional experiments using a fixed sample size only examine the data and calculate confidence intervals at the end, meaning that there is only one point within the test programme where acceptance or rejection can occur. It may be tempting to use the same algorithm and the same p-value (i.e. the probability of finding a result at least as extreme as that observed if the null hypothesis were true) repeatedly at each interim analysis within sequential testing. However, this would result in far more opportunities for the test to be accepted or rejected, and hence a crossing of the acceptance or rejection line would in actuality represent a far lower level of confidence than the tester was expecting. This could result in an increased number of false positives and false negatives (Wald, 1992).

Alpers (2019) illustrates this by showing how a computer simulation using randomisation to replicate a clinical trial where there is no difference between the treatment and the placebo can pass below the intended p-value at points within the trial and then rise above the threshold again. If a sequential approach was used, this dips below the threshold would result in the trial being terminated and the treatment being wrongly accepted. Clearly, therefore, there is a higher chance of p-values below the threshold being observed if peeking occurs at every step, relative to if the p-value is only examined at the end. In the example given, only two of the six trials failed to record a p-value lower than 0.05 (the acceptance threshold) when subjected to peeking at every step, despite the randomisation being designed to simulate no difference between the 'treatments'. This means that 67% of the trials recorded a false positive, whereas a p-value of 0.05 means that only 5% of trials would be expected to result in false positives.

A method is therefore needed to calculate a 'stopping p-value' that is different to the overall p-value required for the trial, this p-value making it harder for the trial to stop early (i.e. requiring the result to deviate further from the acceptance threshold), in order to compensate for the effect of peeking. Note that when reporting the p-value at the end of the trial, the overall trial p-value should be used to represent the confidence in the results; the adjusted 'stopping p-value' should not be used as it is not representative of the confidence in the results, and is merely an interim artefact in order to arrive at representative thresholds during the test.

There are multiple approaches that all provide estimations of the effect of peeking but do not provide perfect compensation. The aim should therefore be to select an approach that provides sufficient accuracy in the estimation of statistical confidence to be acceptable, whilst also remaining reasonably understandable to non-statisticians.

There are multiple approaches that are widely used and accepted to be sufficient for the purposes, broadly falling under the category of 'Interim Sequential Analysis' (sometimes referred to as 'Group Sequential Analysis'), where the data is examined and a decision made on whether to continue at certain pre-defined points within the trial, and full sequential analysis where the data can be examined after each sample. Whilst full sequential analysis appeals due to the flexibility provided, interim sequential analysis still allows significant savings due to early stopping, and tends to be more understandable.

Some examples of suitable methods are as follows:

- Pocock boundary is an example of a group sequential analysis where results can be examined at a set number of predetermined points, with the stopping p-value adjusted to compensate for the number of looks, this p-value being the same on each look. The higher the number of looks, the greater the adjustment to the p-value (Pocock, 1977). This method is relatively simple to apply, but using a constant p-value fails to capture the level of uncertainty diminishing as more data is collected throughout the trial.
- Haybittle–Peto boundary also uses an adjusted p-value at intermediate peeks. However, the p value for these intermediate peeks is always 0.001, regardless of the number of peeks. For the final peek at the end of the trial (if not stopped beforehand), the stopping p-value is equal to the desired overall trial p value. This makes the method relatively easily understood, although it is criticised for being less likely to stop early - essentially it just mitigates the increase in opportunities to observe extreme results by setting an extremely high threshold for early stopping. This means that the p-value remains a reasonable approximation of the significance of the results when the final sample is taken, but also means that savings due to early stopping are less likely. Additional peeks can be added as the trial continues, as neither the interim p-values nor the final p value are affected by the number of peeks, making this method relatively flexible (Pocock, 1992). Care must be taken that the fist peek is not done too early as although 0.001 is generally a conservative level for an adjusted p-value, for very low sample sizes it is possible that it would make the likelihood errors unacceptably high (Pocock, 2005).
- O'Brien–Fleming boundary is also similar to the Pocock boundary but features stopping p-values that are different at each pre-determined peek as the experiment progresses, with lower p-values on earlier peeks requiring more extreme values for the experiment to stop, and the p-values increasing on later peeks to reflect the larger sample size (Pocock, 1992). This could be seen as analogous to Figure 8, which also requires less extreme results to cause stopping as the experiment progresses, and therefore can be seen to overcome the limitation in the Pocock boundary approach where the constant stopping p-value fails to reflect the continuing increase in the number of samples collected (Kumar and Chakraborty, 2016).
- Error-spending functions allow peeking at any point during the test, without this having to be predetermined (DeMets and Lan, 1995). This can be used to decide whether to stop after every test run (full sequential analysis), or at points within the trial that may or may not be equally spaced (group sequential analysis, as per the previous methods). This approach therefore allows the maximum flexibility, with the negative aspect being that the stopping rule is not easily understood, being reliant upon highly complex mathematics. Software is available for error spending functions, albeit less readily available than for tradition fixed-sample methods of statistical analysis (Albers, 2019). There are two types of error spending functions: Alpha spending functions control the null hypothesis rejection, and therefore relate to type 1 errors (false acceptance), whereas beta spending functions relate to type 2 errors (false rejection).

The first three methods above require a maximum sample size to be pre-determined such that the trial will terminate without any definite conclusion being reached if the targeted confidence is not reached – while this is a valid outcome for scientific studies, it is not appropriate for engineering acceptance testing, where a definite conclusion to accept or reject will be needed. Error spending functions allow the test to be extended until a definite result is obtained, but with no guarantee that this point will ever be reached. Both of these limitations support the need for a time-bound means to arrive at an acceptance decision, as provided by the 'Bounded Sequential Testing' approach proposed within this report.

HORIBA-MIRA.com
HumanDrive.co.uk
Twitter: @HumanDriveCav