University of Warwick
Doctor of Engineering Programme

"Executive Summary"

Ian Kendall, Jaguar Cars Ltd

R

26 July, 1999

Ian Kendall
Jaguar Cars

Tenth Portfolio Submission

26 July, 1999

"Executive Summary": Key Challenges in the Development of Automotive Software-based Electronic Control Systems

Contents:

Ian Kendall
Jaguar Cars

ABSTRACT

This document is the executive summary of the author's Engineering Doctorate portfolio. It is entitled "Key Challenges in the Development of Automotive Software-based Electronic Control Systems", and presents the main results from two distinct but interrelated projects, each of which addresses a different challenge. It begins with an introduction to automotive software-based electronic control systems, and emphasises the problems associated with the use of software. The first project was concerned with using software in a safety-related control system, and addressed the issue of how to demonstrate that it was sufficiently safe, i.e. that there was sufficient confidence that the software would behave as expected, even under fault conditions. The second project was concerned with the investigation and introduction of a new simulation technology for developing electronic control systems. It sought to address the challenge presented by the need to develop ever more complex systems, whilst at the same time reducing dependence on prototype testing, and reducing the time available for development. The projects ran concurrently throughout, and both were performed within the Electrical/Electronic Engineering department at Jaguar Cars, who are responsible for the development of all the control systems within Jaguar's products. This summary concludes with a section which hypothesises on the future direction for automotive software-based electronic control system technology, against a context of the business environment in which the industry operates.

Ian Kendall
Jaguar Cars

1. Background and Introduction

Vehicle manufacturers are continually challenged by ever more aggressive moves towards better quality, quicker time to market, more features and lower costs. Other key business pressures are shown in Figure 1, which also illustrates how these often pull in different directions. For example, how can higher quality be achieved when there is less time to do it, and how can more features be added for less cost?

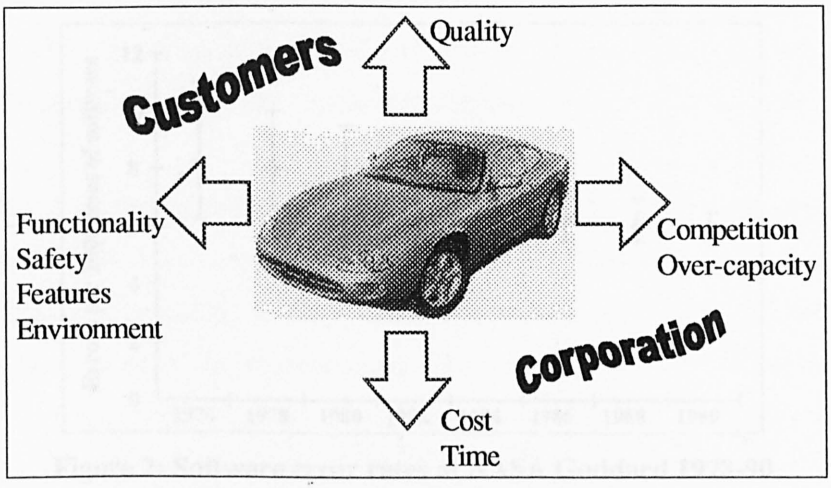

Figure 1: Automotive business drivers

To help meet this challenge sophisticated software-based electronic control systems content in vehicles has shown rapid growth over the last 10 years, and industry predictions are for this to continue. Software provides a very special challenge in that it exhibits very different characteristics to the mechanically based system with which the motor industry is traditionally familiar. This situation is exacerbated by the migration of software into areas where vehicle safety is involved, usually as an enhancement to safety, but dependent on the software working correctly.

Software now controls many aspects of a vehicle's function (e.g. engine management, comfort and convenience features, braking systems, navigation, adaptive cruise control etc.) and, in value terms, over 30% of the total cost of a luxury car is in the electrical system and its associated software. Also, customers are heavily exposed to software, often unknowingly, and quality measures can be significantly affected by defects - on every car! Furthermore, legislation in the USA, and soon in Europe, mandates comprehensive on-board diagnostics for monitoring exhaust emissions control systems, which are mostly implemented in software. Defects in this software, causing malfunctions in the diagnostic system, will result in a failure to get certification, and the authorities can use draconian

powers to prevent a company from selling cars, or issue fines, which can run into tens of millions of dollars.

It is generally accepted that software is not perfect. Even the best software will still contain defects. Figure 2 shows some data gathered at NASA Goddard [1] over a number of years. It can be seen that despite the dramatic progress in software technologies over that period, there has been little improvement in underlying defect rates, although there is some improvement in reducing variability. (i.e. closing the gap between the best and the worst).

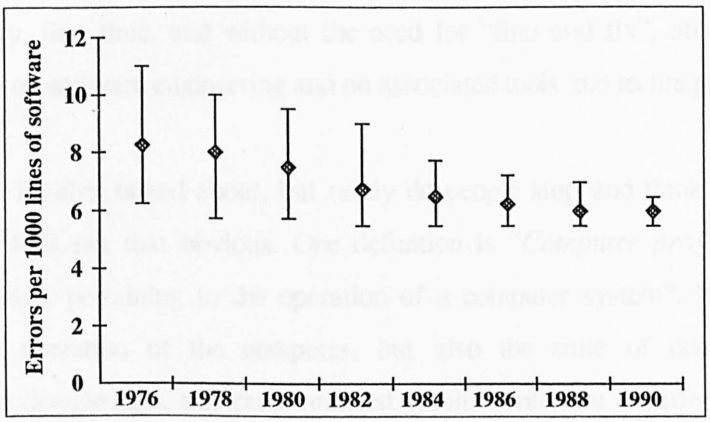

Figure 2: Software error rates at NASA Goddard 1978-90

The main issue is whether these inherent defects prevent the software from functioning correctly within its envelope of operation. It must be remembered that software cannot be fully tested to demonstrate correctness. The consequences when it goes wrong can be wide ranging and serious, with adverse costs running into millions and posing a major threat to profits. Some of the possible consequences of visible software defects finding their way onto the market include:

- Stopped production (e.g. failure to get certification), resulting in workforce lay-offs and loss of sales/resales;
- Recalls. 100% of vehicles would be necessary, as any defect would be identical in all copies of the software;
- Reworks in plant and at port of entry. Reprogramming using FLASH technology, rechipping using EPROM technology, or even complete replacement if "masked" ROM is involved;
- Customer dissatisfaction, resulting in high warranty costs in the short term, and loss of reputation and market share in the longer term;
- Product liability and litigation risk, if defects in safety-related software result in damage or injury;

- Drain on engineering resources to "fire-fight" problems, with knock-on effects to future model years, making it more likely that further problems will occur again.

Jaguar has had its fair share of concerns with software based control systems, often with director level visibility and large cost implications. However, despite this, it is still the case that many people in the organisation still have little knowledge, understanding or interest in the root causes of defects in software. Since software was introduced, the culture in the organisation has aligned to fixing problems quickly when they occur, viewing the of creation of software as something which is too technical, and best left to the "boffins". However, as the organisation shifts to understanding how to deliver world-class levels of quality, first time, and without the need for "find and fix", an opportunity exists to introduce more focus on software engineering and on associated tools and techniques.

What is software? It is often talked about, but rarely do people stop and think carefully about what exactly software is. It is not that obvious. One definition is "*Computer programs* and *associated documentation* and *data* pertaining to the operation of a computer system". It is not just a set of instructions for the operation of the computer, but also the suite of documentation, such as specifications, design descriptions, test cases and test results. Software is different to other forms of engineering, e.g. mechanics, but also has some similarities. For example some of the differences are:

- Its intangibility, as born out by the definition above. Software cannot be regarded a single entity, but is different things, to different people, at different stages of its development;
- It is usually very, very complex. Most of the problems with software are associated with its complexity, and the ability of any individual to visualise all the possible behaviour. There are millions of possible paths through even a simple piece of software, which is why it is impractical to test it completely;
- Software is a pure design. There is no concept of manufacturing, except making identical copies of it.
- It does not wear out, corrode or age. Although problems do occur when a mechanical system, being controlled by software, does.
- All copies are the same, there is no variability. Although again problems do occur with hardware variation.
- Software failures are not random, but systematic. Given an identical set of circumstances, a failure will occur every time. Therefore, traditional reliability methods are not easily applied, or even impossible to apply, to software. For example, what is a mean time between failure (MTBF) when it is known the failure will always occur? And how do you approach an FMEA on software? See

Figure 3 for an interpretation of how stress-strength interference, a common way of thinking of mechanical failures, could be interpreted for software.

- It is perceived as easy to change. Many people see it as a simple manipulation of a text file. The reality is very different if a quality change is to be made – all documentation must be revised, a detailed analysis of what else might be affected needs to be performed, test cases need to be generated, tests re-run and reports written, etc. Every time a change is made there is risk. Ed Adams from IBM is attributed with the statement, "Every change has a 15% chance of causing a problem at least as large".

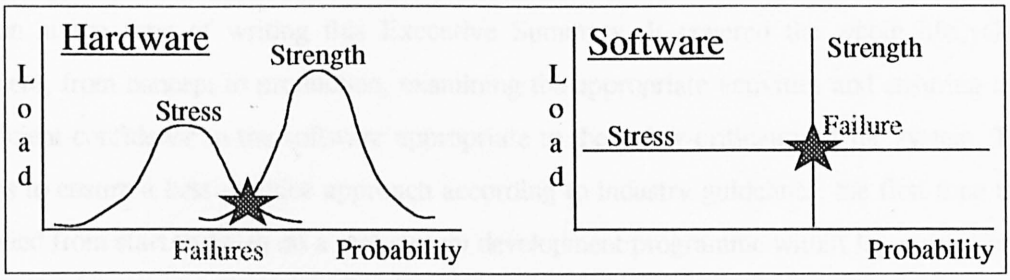

Figure 3: Stress-strength interference - an interpretation for software

Conversely there are similarities between software and other forms of engineering, although, perhaps because of the intangibility of software, these are often neglected:

- It is valid to talk about software reliability, but it must be understood that software failures are not easy to define. Software "failures" can be counted and properties like MTBF and reliability growth can be calculated. Although it must be remembered that it is not the statistical properties of the software defect itself that is being measured (it is always present), but rather the statistical properties of the set of circumstances which cause it to surface. Commercial software often exhibits very poor reliability, which would be unacceptable to the automotive sector, e.g. most people's experience of word processor software would support a claim that it typically has an MTBF of only a few hours!
- Software should have an equivalent level of change control associated with it as exists for mechanical components. For example, no quality engineering organisation would allow a dimension change to a mechanical component, without a design change to the drawing. However, it is not unknown to allow modifications to software code, without the discipline to cascade the change into all the associated documentation, if it exists.
- Load bearing mechanical components are unlikely to be committed to production without first being subjected to finite element modelling and stress analysis. However, it is commonplace to have software which has had little or no modelling and analysis performed on it.

- Vehicle manufacturers rarely use a mechanical component supplied by a third party without performing full testing themselves, or asking for evidence that the supplier has done it. However, it is commonplace to trust suppliers to deliver correct and complete software, merely as an "executable" file or already installed in an electronic control module, without asking for any detailed information.

This background provides the basis for two distinct but interrelated Engineering Doctorate projects. The first project looked at the issue of how to approach software for a new safety-related electronic throttle control system, fitted for the first time to the Jaguar S-Type, which had just gone in to production at the time of writing this Executive Summary. It covered the whole lifecycle of the development, from concept to production, examining the appropriate activities and ensuring that there was sufficient confidence in the software appropriate to the safety-criticality of the system. The main focus was to ensure a best practice approach according to industry guidelines, the first time these had been applied from start to finish on a real system development programme within Jaguar or Ford, or, to the author's knowledge, anywhere to such a level of detail. The object was to ensure that as many of the software defects as possible were removed, by attention to detail, by the application of best practice methods to all development phases, and by providing evidence that this has been achieved.

The second project looked at a new and innovative method for specifying, analysing and testing software based electronic control systems. It aimed to demonstrate how state-of-the-art modelling and simulation techniques can be used, to bring a significant proportion of verification and validation effort for automotive software-based control systems into the laboratory, instead of relying on costly and resource intensive prototype vehicle testing. This also seeks to find ways of reducing defects in software, but this time in the context of compressed development times and reduced development costs. It has resulted in a 7 figure investment programme, with support at the highest levels of management.

It was the author's own experience with the first generation of electronic throttle systems in Jaguar which led to both projects. The S-Type electronic throttle was the first full-authority system in Jaguar or Ford to be engineered for a production vehicle. Ensuring that there was sufficient confidence in the safety of the software was one of the major issues surrounding the development, and the author's role as a second-party independent safety assessor was a key factor in delivering the required confidence in the system. Although the technology for electronic throttle has been around for many years, it has been concerns about how to handle the safety issues associated with the software which has prevented it from reaching the market before now. The author was involved in MISRA (the Motor Industry Software Reliability Association) from its inception, and the work within the MISRA team, with

representation from across the UK motor industry, resulted in the production of the MISRA Guidelines for the Development of Vehicle Based Software. This was the first time software engineering best practice suitable for safety-related systems had been defined in a form appropriate for automotive application. These Guidelines were therefore the key to overcoming the concerns about software safety on the S-Type electronic throttle. The author's previous experience overseeing a third-party safety assessment, albeit on a non-full authority electronic throttle system (for the 1997 XK8 sports car), coupled with a detailed knowledge and understanding of the MISRA Guidelines, came together to provide a sound basis from which to approach the development for the most critical software for the new full-authority system for S-Type.

The potential for HILST, or Hardware-in-the-Loop Simulation Testing, first came to the author's attention after the experience of performing extensive safety validation tests on the first electronic throttle system. This consisted of working in cramped conditions in an XK8, surrounded by an electrical "breakout box", laptop computer, and various other test equipment, laboriously inducing faults, in many different driving conditions, to evaluate the effect on the vehicle. Trying to induce a fault and capture the results, whilst doing 100mph around a banked corner on a test track is not a pleasant experience, especially as the work took several weeks to complete. HILST seemed to offer the potential for using simulations to test the system, tests which could be performed in a controlled environment, would be repeatable, and that could even be automated. Furthermore, it would also be possible to carry out some test cases which were impossible to do in the real car. (For example, the effect of mechanical failures, which could not be induced without access to the throttle body under-bonnet, were impossible to evaluate whilst the vehicle was moving). From this entry point, a whole new development approach, not just for testing, but also for control system specification, design and analysis, was revealed.

Both projects are very much at the forefront of current thinking in automotive engineering technology, and are two of the most significant challenges related to the use of ever more complex software based control systems. They involve a significant amount of innovation, the first in order to bring a specific system safely to the market in a timely manner, and the second to fundamentally change the way in which electronic control systems in general are developed (i.e. one is a "product" innovation, the other a "process" innovation).

This document is structured first to provide a brief summary of each of the nine portfolio submissions, followed by chapters on some of the detail in each of the two projects in turn. It concludes with a section of overall conclusions, including some speculation on the future for automotive electronic

control systems. Also, in Appendix A, is a review of the "Project Features" required for the Engineering Doctorate, which outlines how these have been achieved via the portfolio submissions.

2. Structure and Summaries of Portfolio Submissions

As mentioned in the introduction, there are two projects, both of which cover different aspects of automotive electronic control systems engineering. These two projects, and the portfolio submissions which support them, are shown in Figure 4. Note that the 9 submissions are numbered according to the order in which they were written.

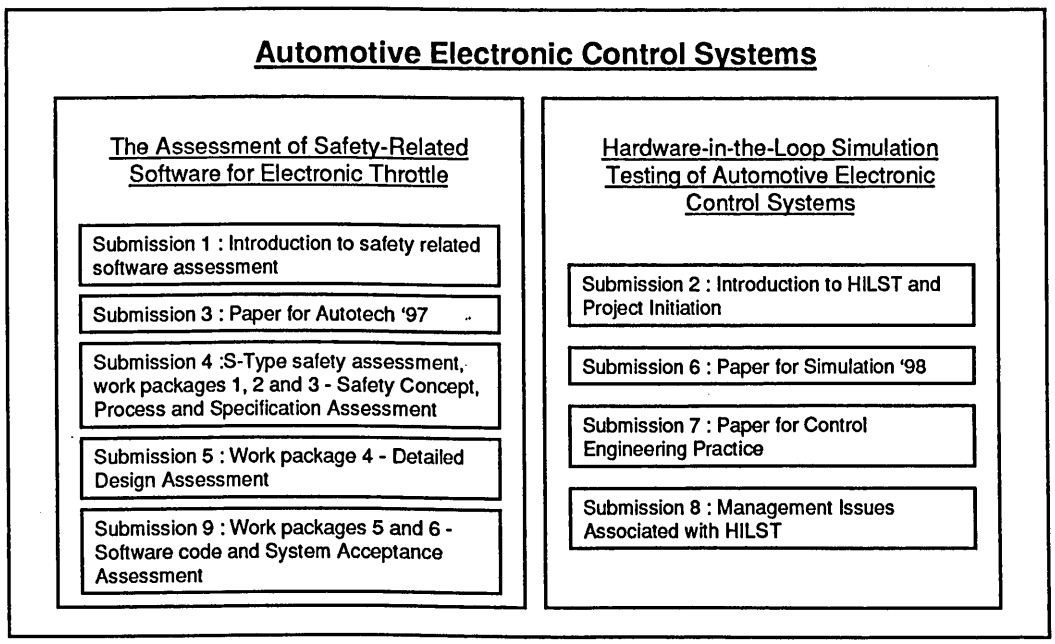

Figure 4: Projects and associated submissions

The projects and associated submissions are briefly summarised in the rest of this section, and more detail on the results of the two individual projects can be found in sections 3 and 4 respectively.

2.1 The Assessment of Safety-Related Software for Electronic Throttle

In November 1994, the Motor Industry Software Reliability Association (a consortium of UK automotive companies, suppliers and consultants) published "Development Guidelines for Vehicle-Based Software" [2].

Ian Kendall
Jaguar Cars

The new 1999¼ model year Jaguar S-Type, code-named X200, includes a full-authority electronic throttle for the first time on a Jaguar, or a Ford, vehicle. Electronic throttle removes direct control of engine power from the driver (traditionally done via a mechanical throttle cable) and places it under software-based computer control. Safety is of paramount importance, and is essentially achieved through the identification of safety-related software, which must then be developed to have sufficient confidence that it will not cause the vehicle to accelerate when not required. MISRA provides guidance on how to approach the development of safety-related software for an automotive application. This project used the MISRA Guidelines, in detail for the first time, to ensure there was sufficient confidence in the safety of the S-Type/X200 electronic throttle monitor software, before it went into production. It essentially involved the interpretation and application of the MISRA Guidelines to a specific project, and, by definition, required careful and meticulous reporting of the findings.

This project, described in more detail in section 3, was covered by 5 portfolio submissions, which are summarised as follows:

2.1.1 Portfolio Submission 1: Introduction to Safety-Related Software Assessment (13th Mar 1997)

The purpose of this submission was to discuss the issues for software in safety-related applications, particularly in the current automotive market, and to explore the role of standards and guidelines in this area. The historical context for the work on safety-related software for electronic throttle applications within Jaguar was described. It sought to provide all the necessary background information required to understand the position of safety-related software, and its assessment, within the automotive industry, and to relate this to specific Jaguar projects. Included is a review of other industry sector approaches; an analysis of the UK automotive industry's approach (MISRA), in which the author played a major role; an introduction to Jaguar's current electronic throttle applications; and details of a previous similar project from which useful lessons were learned. The framework for follow-on work on the new electronic throttle projects (including S-Type/X200) was established. Two of the most significant contributions to this framework were a new Jaguar Engineering Standard for a risk-based approach, and a plan for the assessment activities on the S-Type/X200 electronic throttle safety-related software, which were to follow. The basis for the assessment plan was the division of the work into 6 work packages, which together covered all of the different sections of the MISRA Guidelines. The 6 work packages are :

1. Preliminary Safety Analysis and Integrity Assessment. } reported in Submission 4

2. Software Quality Assessment. } reported in Submission 4
3. Requirements Assessment. } reported in Submission 4
4. Design Assessment and Detailed Safety Analysis. } reported in Submission 5
5. Safety-Related Code Assessment. } reported in Submission 9
6. System Acceptance Assessment. } reported in Submission 9

2.1.2 Portfolio Submission 3: Conference Paper for Autotech '97 (2nd Feb 1998)

This is a reprint of a paper titled "A Safety Analysis Methodology and its Automotive Application", presented by the author on the 5th November 1997, at the "Autotech '97" conference held at the National Exhibition Centre, Birmingham. It was jointly authored by Jaguar Cars and Leeds University, one of the main partners in the PASSPORT project [3] [4]. PASSPORT was an EC funded research project to propose a methodology for the analysis of safety related transport telematic systems. Jaguar were the first automotive manufacturer known to have applied the techniques to a production system, and this paper briefly describes the work done using the PASSPORT methodology for analysing electronic throttle systems. Although the paper does not include product specific information (for confidentiality reasons), it is based on the work performed for the S-Type/X200 full-authority electronic throttle system. This paper was written in the context of another EC project (sponsored under Framework IV) called COMPASS, in which both Jaguar and Leeds University are partners. COMPASS has the objective of developing a computer based tool to implement the PASSPORT safety analysis methodology.

2.1.3 Portfolio Submission 4: Safety Concept, Process and Specification Assessment (2nd Feb 1998)

This submission is a report of the first 3 of the 6 work packages identified in the assessment plan presented in Submission 1 – Preliminary Safety Analysis (PSA), Software Quality Assessment, and Requirements Assessment.

The development of the S-Type/X200 electronic throttle system was the responsibility of Ford's Powertrain Control Systems Engineering department in the USA, who are acting as a supplier to Jaguar. The main objective for the author was therefore to support the designers, giving independent assurance that the recommendations of the MISRA "Development Guidelines for Vehicle Based Software" [2] were suitably followed throughout, and providing documentary evidence to support this claim.

The first work package begins with an explanation of the PASSPORT Preliminary Safety Analysis methodology [3], and goes on to show how it was applied for the first time on a real industrial project. This was the basis for the published Autotech paper mentioned in the previous section. The main outputs from the safety analysis were an understanding of the hazards, of which the top-level one was confirmed to be "unintended vehicle acceleration"; the safety requirements; and the worst-case integrity level, determined to be MISRA level 3. The main achievement was to influence the decision to secure provision for a separate monitoring subsystem, known as the ETM (Electronic Throttle Monitor), with its own dedicated development team.

The software quality assessment, in the second work package, covered a detailed examination and audit of the safety-related software development process within Ford. A clause-by-clause review against the MISRA guidelines was performed, appropriate to deliver integrity level 3 software. Several deficiencies and corrective actions were identified.

Finally in this submission there was an assessment of the specification documentation hierarchy, and the traceability for the safety requirements, which formed the activities for work package 3. This included a piece of work using formal mathematical methods for the first time on a production automotive system.

The conclusion of the first half of the safety-related software assessment of the electronic throttle, could claim a number of successes, and this was only possible through involvement of the assessor from a very early stage. The ETM team in particular, became well attuned to the needs of delivering safety-related software with sufficient integrity. Although several deficiencies and concerns were raised and corrected along the way, the software in the ETM, was deemed to be specified to provide Jaguar customers with an appropriate level of protection against the known level 3 hazards. This formed the basis for the team in the USA to go ahead and finalise a design.

2.1.4 Portfolio Submission 5: Detailed Design Assessment (31st Aug 1998)

This is the fourth work package of 6, and looked at the design of the S-Type/X200 electronic throttle monitor (ETM) subsystem software. Two points of view were considered. Firstly, the ETM was considered in the context of the overall system using PASSPORT Detailed Safety Analysis as a framework to perform Failure Mode and Effects and Fault Tree analyses. The second point of view considered the ETM software. The ETM is an independent monitor processor, with its own software, which places an "envelope" of safety around all the other electronic throttle components, and can force

the system to a safe state if it detects an error. It was therefore important to ensure a robust software design, to integrity level 3, was achieved for the ETM, using the MISRA guidelines as a reference.

As in the first 3 work packages, a handful of minor issues were uncovered. However, recommendations were made to the design team to address them, thus ensuring that a robust software safety argument was maintained.

2.1.5 Portfolio Submission 9: Software Code and System Acceptance Assessment (1st Mar 1999)

This submission is a report of the final two work packages, associated with the coding phase and system acceptance, for the safety-related software of the electronic throttle control system.

The fifth work package, safety-related code assessment, looked in detail at the programming phase for the Electronic Throttle Monitor microprocessor (ETM). As mentioned above, the ETM is designed to place an envelope of safety around all other components in the system to prevent "unintended vehicle acceleration", and therefore is assumed to be the only software which needs to be classed as safety-critical to integrity level 3. As well as the MISRA clause-by-clause review, this work package also included a software complexity analysis, followed by some extended unit tests on a selection of the most complex software modules.

The final work package was the acceptance assessment. This pulled together all the safety-related testing activities performed on the software as part of the system, plus a few loose ends from elsewhere within the MISRA Guidelines (recommendations on off-board diagnostics, software maintenance and process metrics). It also included a significant phase of work to perform some final, independent safety-validation tests on the whole system, as installed in a vehicle, to confirm safe operation under many different failure conditions.

Included at the end of this submission was a brief summary of all 6 work packages. The overall conclusion was that there was sufficient evidence that the S-Type/X200 electronic throttle monitor subsystem software had been appropriately engineered, according to the recommendations of MISRA, to achieve safety integrity level 3. In conjunction with the overall system validation (outside the scope of this project) ,the system could therefore be considered, within the bounds of what was known, sufficiently protected against "unintended vehicle acceleration". Hence, there was sufficient confidence,

and evidence of that confidence, for the S-Type to go into full production. This represented a successful result for the project.

2.2 Hardware-in-the-Loop Simulation Testing of Automotive Electronic Control Systems

Strategically, the direction of Jaguar's (and Ford's) future product development will require a greater dependence on computer modelling and simulation, rather than the testing of expensive prototype vehicles, and this project has moved the company towards this goal. The project aimed to establish the introduction of computer based hardware-in-the-loop simulation testing (HILST), and to demonstrate its feasibility. HILST is a technique which can greatly assist in the development of electronic control systems. Although the Corporation was heavily investing globally in many forms of CAD, CAM and CAE tools, there was nothing specifically targeted at control systems. Hence it was necessary to initiate a local activity within Jaguar, to develop and introduce HILST technology.

This project is described in Section 0, and was covered by 4 portfolio submissions, here summarised as follows:

2.2.1 Portfolio Submission 2: Introduction to HILST and Project Initiation (14th Jul 1997)

This submission aimed to give the reader an appreciation of Hardware-in-the-Loop Simulation Testing (HILST), and to provide an understanding of why the author believes it is an important new development in the automotive electronics and software engineering design process. It includes a literature review, which was used to assist in the identification and selection of suitable tools for simulation and HILST, and to seek to establish the level of activity in some of Jaguar's competitors. It was concluded that Jaguar should adopt the MATLAB®/SIMULINK®/STATEFLOW® toolset from Mathworks Inc. as the simulation environment, and dSPACE as the real-time HILST platform. The strategy for the introduction of these in to Jaguar's electrical engineering process was described, to explain how control system simulation and HILST could fit into the wider CAE context. It concluded with a plan for a small pilot project, based on the XK8 driver's door control system, the objective of which was to demonstrate the technical feasibility of HILST, using the aforementioned tools.

2.2.2 Portfolio Submission 6: Conference Paper for Simulation 98 (2nd Oct 1998)

This submission covers the first phase of work for the HILST pilot project, and describes the technical work involved in developing an off-line model of the XK8 driver's door control system. The off-line model was intended to enable "plant" models to be developed, suitable for testing a real driver's door control module (DDM) against. The "plant" is everything in the system which is to be controlled by the electronics and software. In the case of the driver's door there were three main sub-systems and functions – central locking, the electrically positioned wing-mirror, and the electric window. The model is "off-line" because all the work was performed within the desktop computer. Real-time execution was not considered, only the behavioural aspects of the sub-systems.

Because the pilot project was based on a system which was already in production, there were no product confidentiality concerns, and it was decided to seek to publish and present the technical work externally. This submission is a reprint of a paper entitled "Simulation as a Means of Achieving "The Impossible": An Investigation Into The Use Of Simulation in the Development of Electronic Control Systems at Jaguar Cars". It was presented by the author on 30th September 1998, at the "Simulation '98, Innovation Through Simulation" international conference, held at the University of York, on 30 September to 2 October 1998, and organised by the IEE.

Although publishing for a conference meant that much of the technical detail had to be excluded (due to the need to keep the length of the paper down), the salient points are covered. The most significant innovations were the selection of the driver's door system itself, which had not been modelled in this way before; the development of an SCP communications board for dSPACE and software for SIMULINK® (SCP is a proprietary communications protocol used by Ford – including the Driver's Door Module – for which no off-the-shelf solution was available); and the development of a hybrid model, which included both continuous and event-driven modelling paradigms in a single environment.

2.2.3 Portfolio Submission 7: Journal Publication on Implementation of HILST (15th Jan 1999)

Following the successful completion of the off-line model for the driver's door system, the next phase of the pilot project was to go on and develop an on-line model, capable of real-time HILST application with a real control module. Again is was decided to publish externally.

This submission contains the text of a paper, titled "An Investigation into the Use of Hardware-In-The-Loop Simulation Testing at Jaguar Cars", submitted for publication in the Journal of Control

Engineering Practice. It is based on the earlier paper covered in the previous submission, but extends the work to describe the real-time realisation of the simulation. Control Engineering Practice is a journal of the International Federation of Automatic Control, and is published by Pergamon Press Ltd, on behalf of the IFAC.

The pilot project was successfully concluded when it was demonstrated that a real driver's door module (DDM) from the XK8 sports car could be put "in-the-loop" with a simulation of the locks, mirrors and windows, such that the DDM functions as it would when installed in a car. Two of the most notable achievements in reaching this point were the development of a suitable interfacing system for the DDM to the HILST equipment, and the corrolation and tuning of the model to match real measured data. The project culminated in the demonstration of test scripts which could automate the functional testing process.

2.2.4 Portfolio Submission 8: Management Issues associated with HILST (18th Jan 1999)

This final submission covered the management issues involved in the exploitation of Hardware in the Loop Simulation Testing (HILST). The activities described are the direct result of the work carried out during the pilot project, and as part of the Engineering Doctorate, and show how Jaguar is pursuing HILST to make it part of the mainstream development process for electronic control systems for new vehicles. It includes a review of the current situation within the author's department, Electrical/Electronic Engineering, who are responsible for all Jaguar's control systems; a description of how the idea of HILST was "sold" to senior management and others; an explanation of the resource plans and training; and a description of the budgets and new facilities put in place for the large scale future implementation of HILST technology. It therefore has in its title "from innovation to exploitation".

2.3 Relevant IDGS Post Module Work

Two of the additional four IGDS modules which were required for this EngD degree provided an opportunity to use material from the project work, but looking at aspects not covered by the main portfolio submissions.

2.3.1 The Management of Change (12th Mar 1998)

This post module work examined the Hardware in the Loop Simulation Testing (HILST) project as an organisational learning and change project, rather than as a purely technical one. This proved a significant watershed, as it shifted emphasis from technical research to implementation and management influence. Such a shift was a key factor in guiding the author to lead the subsequent development of an implementation team, and the continuing investment in HILST at Jaguar.

2.3.2 Collaboration and Control Management (16th Nov 1998)

One of the key aspects of HILST is the ability to interface the "real world" with the simulated environment. No off-the-shelf solution could be found, so Jaguar placed a contract with a small local company to design and build a generic, re-configurable interfacing system to the author's requirements. This module provided the catalyst for an in-depth examination of the commercial arrangements in place between Jaguar and its supplier, and led to a new written contract which clarified the agreement for both parties. The interfacing system, which resulted from the contract, is an excellent solution to the problem identified during the Engineering Doctorate, and now is to be marketed by the company concerned as an innovative new product.

3. The Electronic Throttle Monitor Software Assessment Project

This project was concerned with the use of the MISRA Guidelines (see section 3.1.2) as the basis for assessing the safety integrity of the software for the monitor subsystem for the electronic throttle on the new Jaguar S-type compact saloon, code-named X200, which went into production in January 1999.

The project began with a review of automotive safety-related software and its assessment. After considering the approach in other industry sectors, and the experience from a previous Jaguar electronic throttle project, a framework was established for the S-Type/X200 monitor subsystem software assessment. This framework consisted of a new Jaguar risk based approach, together with a plan for the assessment split into 6 work packages. Each of these work packages was then carried out using the MISRA Guidelines as the main reference, and the activities and results were carefully documented, before the system went into production.

Ian Kendall
Jaguar Cars

3.1 The Background to Automotive Safety-Related Software

A safety-critical, or safety-related control system is one where the operation, or rather the mal-operation, of the system has the potential to lead to harm, either to people, property or the environment. Such systems have been around in the automotive industry since the beginning, braking systems, steering and engines for example. However, in recent years there has been a big increase in the use of electronic systems and programmable electronic systems containing software. If these types of control systems are applied in safety-critical areas, then the software embedded within them is known as safety-critical, or safety-related, software.

The concern with software-based electronic control systems in safety-related or safety-critical applications is complexity. Assurance that there is acceptable freedom from risk due to a system usually comes from confidence in the correctness of the design, and the reliability of the components. Complexity impedes confidence. Electronic systems, on a micro-scale, are immensely complex, and software, by its very nature, performs control functions and tasks requiring millions of calculations. It is virtually impossible, in all but the simplest programs, to guarantee the absence of errors, and if such errors exist, their manifestation and effects may be diffiçult to predict with certainty.

Safety-critical and safety-related software is also sometimes referred to as high-integrity software. The principle of integrity is related to the confidence that the software will behave correctly and reliably in its environment. The more safety-critical a system is, i.e. the greater the risk of harm that could be caused by a failure, the higher the confidence that is required in the design, and hence the higher the integrity. The amount of integrity inherent in a piece of software, or its integrity level, and how to "measure" it, is an important concept that lies at the heart of the current thinking in safety-critical systems.

Jaguar does not perform the detailed design, nor write the software, for any of its electronic systems. Instead complete systems are bought-in from suppliers who have particular expertise in given application areas. This has always been the situation and has worked well, allowing Jaguar to leverage high levels of feature and functionality by utilising the skills of its suppliers. Strategically, the view has been that detailed design is entirely a matter for the suppliers. Jaguar does not need to know details, and should treat a bought-in system simply as a "black box", trusting the supplier to get it right. However, it should not be assumed that, because systems are bought-in, the responsibility for software functionality and its integrity should rest entirely with the suppliers.

Like most automotive manufacturers, Jaguar has many procedures and methods for most aspects of supplier quality assurance, such as the very comprehensive QS9000 standard[1] [5], but until relatively recently, nothing much for software. It is believed that the correct approach is for Jaguar to work more closely with its suppliers, taking an interest in their safety-related software development processes, assessing their capabilities and work products, and suggesting improvements and solutions. When a vehicle is signed-off for production, there must be a "measurable" level of confidence, i.e. integrity, in the safety-related software.

3.1.1 Legal Issues

Manufacturers of all consumer goods are subject to laws across the world which govern their fitness for purpose and safety. If an accident occurs due to a contravention of any of these laws, a company, its directors, and, in theory although highly unlikely, individual employees can be criminally prosecuted under such laws. A successful criminal prosecution by the State normally results in a fine, but in some cases prison sentences.

Often a criminal trial will precede a civil action. In civil law an individual (or group of individuals), or organisation can take action to prove liability against another individual or organisation, resulting in the payment of compensation or damages. In the UK, the Consumer Protection Act has introduced the principle of strict liability - an American concept that has now also become part of a European Directive enacted into UK law. All that is required under strict liability is to show that a defect in a product exists, and that the defect caused the damage or injury, and not, as before, that it was there as a result of negligence. This means that, despite showing a duty of care towards the product's user, if there is a defect, then the defendant is liable. This perhaps represents a bigger threat to a manufacturing company than criminal proceedings, as the damages awarded can be enormous, particularly in North America. Juries have been known to award hundreds of millions of dollars for individual claims, in what is known as punitive damages, where the objective is to punish the defendant, for example by destroying profits, rather than just to compensate the plaintiff. Civil litigation often results in out of court settlements, where the lawyers agree financial terms behind closed doors. This has the advantage of not only avoiding the massive punitive damages a jury may award, but also avoids the bad publicity.

There are very few defences under strict liability. Once a defect is shown to exist, and shown to have caused the damage or injury, then the case is lost. However, the size of the damages is still to be

[1] QS9000 is the automotive industry's interpretation of ISO9000, but extends and interprets the requirements to match them to fit the industry's methods of operation. It was written jointly by the "big three", Ford, General Motors and Chrysler, and has been widely adopted by them, their subsidiaries and their suppliers across the world.

decided. In such a situation, a manufacturer as a defendant can argue that, at the time the product was designed, it was not possible to foresee the eventual consequences, that all the latest state-of-the-art techniques were applied, and that all reasonable precautions were taken in accordance with industry best practice. If successful, the damages awarded, which although compensating the plaintiff, will not seek to punish the manufacturer of the product. Even out-of-court, the size of the settlement will be determined by the strength of the defence. Cost, ignorance or incompetence are not defences, and such revelations in court are likely to result in large punitive damages [6].

It is phrases such as "state-of-the-art", "industry best practice", "reasonable precautions" and "unforeseeable consequences", that lead manufacturers to refer to standards (as a minimum), and emerging standards, for safety-critical software [7]. Defects in software are inevitable. Every effort needs to be taken to find and remove any defects, and to ensure those that remain do not compromise safety in any way. It is essential to be able to demonstrate state-of-the art techniques, reasonable precautions, industry best-practice and a serious attempt to foresee all consequences throughout all stages of the process of creating safety-critical software. Manufacturers must endeavour not just to do a good job, but also to have evidence that they have done so.

3.1.2 The MISRA Guidelines

MISRA stands for the Motor Industry Software Reliability Association, and was formed around 1991, with the objective of defining the UK automotive industry's approach to safety-related software. It consisted of vehicle manufacturers and suppliers, organisations expert in software engineering, and the motor industry's own research organisation, MIRA, as shown in Table 1. The author was Jaguar's representative throughout, and was still involved in MISRA activity at the time of writing this Executive Summary.

Table 1: The members of the MISRA Consortium

Vehicle manufacturers:	Automotive suppliers:	Consultants/software experts:
Ford (UK)	Lucas Electronics	Leeds University
Lotus	AB Automotive	Centre for Software Engineering
Jaguar	Delco (now Delphi) Electronics	Rolls Royce and Associates
Rover	Automotive Products	MIRA

In November 1994, MISRA published its "Development Guidelines for Vehicle Based Software" [2], which was officially launched at the Society of Motor Manufacturers and Traders in London. It has been well received, and is now recognised by experts both inside and outside the industry, as the UK

automotive sector's recommended approach to safety-related software. Note that it is a set of guidelines, rather than a standard, as MISRA itself is not a standards body, but simply a consortium of parties who were interested in agreeing some common principles. With the existence of the MISRA Guidelines, it could be concluded that any new safety-related project, such as the electronic throttle monitor, should seek to ensure that the guidelines are followed. This is essentially what this project is about – interpreting the MISRA Guidelines in a practical way, and independently assessing progress against each of the recommendations, recording that they were adequately followed by the supplier, and supplementing the work where this was not the case.

During the time that the MISRA Guidelines were being defined, and throughout the whole of the S-Type development programme, the significant international standard IEC61508 [8] was also under construction[2]. IEC61508, which was still not fully published at the time of writing this Executive Summary, attempts to establish a generic framework for safety-related systems and software. As it is generic, i.e. tries to cover all industry sectors equally, from nuclear power stations to consumer products, one of its key stated objectives is to drive interpretation within each separate industry sector, and result in sector specific standards. The MISRA Guidelines certainly cover the software aspects of this, but do not seek to provide much guidance on the approach to system architectures and hardware requirements. However, some work was done within the MISRA working groups and is still the subject of further work within the consortium.

For this project, no attempt was to be made to claim compliance to IEC61508, but its existence was acknowledged, and early drafts were used as additional reference material to assist with some of the assessment work. As the only automotive specific guidance available on the subject was the MISRA Guidelines, the focus for the work was their application to the safety-related software development for the electronic throttle monitor subsystem.

3.1.3 Independent Safety Assessment

One of the currently accepted best-practices in the development of safety-related software is the use of independent assessment. The assessor, an independent expert to oversee and advise whilst not being involved in the day-to-day detail, helps to ensure that the development team are kept on course in order to deliver the required integrity level. The MISRA Guidelines, therefore, also recognised the importance of assessment for software. MISRA describes an independent assessor as "an advocate for the level of

[2] Originally in two parts, the work of two separate working groups within the International Electrotechnical Commission, known at the time as IEC (Secretariat 122) SC65A/WG9 and IEC (Secretariat 123) SC65A/WG10, one covering software aspects and one covering system aspects. Since the first drafts were issued in 1991, it became known first as IEC 1508, and finally IEC61508. There are now 7 parts of which only 3 are currently published through BSI as issued standards.

confidence in the safety delivered to the end customer", whose role is to "demonstrate that the risks associated with the final system are at an acceptable level".

Not only does the independence from the detailed work provide for a "fresh eyes" opportunity to see the situation strategically, it is also important that the assessor can, according to MISRA, "ensure that there is no bias from the development team, nor misplaced pressure from management". In other words, the assessor must be empowered to make, or influence, decisions from a safety perspective and not from a pure commercial or project management perspective. Therefore, some organisational independence of the assessor is essential. Furthermore, MISRA recommends that assessment should not be considered as a checking activity taking place at the end of a project, but rather as a continuous process, in which the assessor is involved from start to finish.

There are two basic types of assessment, process assessment and product assessment. The first of these refers to an assessment of the development process, ensuring that all the appropriate steps are adequately planned and carried out, without any regard to what is actually being developed. The second type concentrates on the product itself, e.g. the software, to ensure that it has the properties and features that would be expected, given the integrity level. A comprehensive independent assessment should include both types.

3.1.4 Risk Background and Legacy systems

The automotive industry has been using electronic control systems in safety related applications for nearly 20 years. It is clear that, despite the absence until relatively recently of standards such as IEC61508 and MISRA, the industry has a good track record of implementing and marketing these systems safely and reliably. For example, electronic fuel injection for diesel engines, found in most modern diesel cars, if it malfunctions, has the potential for increasing engine power output when not required[3]. Also, antilock braking systems depend inherently on software control to achieve their function, and some have potential failure modes which may result in a complete loss of braking. Suppliers of these systems have built up their competence over many years and through the field experience of many thousands, if not millions, of units. Through a process of evolutionary development and continuous improvement, this competence and experience has resulted in very reliable and safe systems, which even though they may have potential hazards associated with them, have contributed to the modern car offering a greater level of safety to the driver and occupants than ever before.

[3] Diesel engines regulate power output by varying the air-fuel ratio, i.e. injecting more fuel will result in more power, even when the inducted air does not change. This is in contrast to a gasoline engine which operates on a nominally fixed air-fuel ratio, and engine output must therefore be regulated by controlling the air induction to the engine, via the throttle. The fuel is then injected accordingly to keep the air-fuel ratio constant.

With the emergence of a more structured approach to assessing hazard risk and functional safety, many of the systems that are already in existence, and have been for some time, could be considered as non-compliant against emerging standards such as IEC61508 [8] and MISRA [2]. Indeed, one could say that if the car itself was considered using modern assessment techniques, then the inherent risks associated with it would not be acceptable, and the car would be a proposition carrying too much risk. However, the road transport system has evolved as a vital part of all developed economies, and the risks involved in driving are accepted and part of everyday life for many, if not most, people all over the world. Whilst the safety of cars is something which all manufacturers continually seek to improve, the risk background of the environment in which motor vehicles operate must be kept in perspective when assessing risk associated with new technologies. Any incidence rate of accidents caused by vehicle design faults is too small to appear in the published Government statistics [9]. By far the biggest causal factors are associated with driver behaviour and poor vehicle maintenance. This, however, is not cause for complacency, or a justification to ignore best-practice for the design and development of new systems. It is simply an example of the importance of keeping things in perspective, and is illustrative of the automotive risk environment.

Many of the techniques used in safety-related systems analysis, such as Failure Mode and Effects Analysis, only focus on the potential for things going wrong. Currently, within the automotive industry, there is no generally accepted way to balance the risks of undesirable failure modes against the safety benefit gained when the system works as intended. Hence, it would be easy to focus only on the negative risks and constrain the introduction of new systems which may have an overall safety benefit, because of their potential failure modes. Air bags are a good example of this. They do have the potential to go off when not required, and not when required to in the event of a collision. However, despite the fact that there is a finite probability of this, and incidents have occurred, overall they are still considered a key safety feature. In addition, in recognition of dangers associated with air bags and small children, the industry is beginning to offer "intelligent" airbags which can adjust their firing to the size of occupant. Such "intelligence" adds complexity and places more demands on the electronics and software, which of course, have yet more failure modes, but the benefits are still believed to outweigh these risks.

Systems or software which have been in existence for some time are known as legacy systems or legacy software. Whilst they may not have had the benefit of today's best-practice at the time they were developed, in general, they have been shown to perform to an acceptable level of safety in the field. Whilst it is a good idea to stay abreast of new standards and guidelines, such as MISRA and IEC61508, as they emerge to see whether they contain any new insights on safety, there is no benefit to

Ian Kendall
Jaguar Cars

be gained from trying to apply them retrospectively. These standards and guidelines are mainly concerned with attempting to minimise risk before systems are commissioned, however if they are already operating acceptably in the field, then there is little justification to place emphasis on them. Indeed, it would be difficult to do this without fundamentally redesigning many of the systems or rewriting all the software, and this may carry more risk of introducing an unwanted error than leaving alone what is already operating safely in the field. It is much more reasonable to seek to concentrate on new systems and the new hazards which they may introduce.

The automotive industry has a significant amount of regulation of the safety requirements for many systems, for example the need for dual circuit hydraulics for braking systems, which are policed to varying degrees throughout the world through type approval and homologation. This policing is usually in the form of tests, measurement and inspections performed on the finished product, and is only truly effective for mechanical and low-complexity electrical systems. At present, there is no agreement on how to regulate for the development of complex automotive safety-related software-based systems, such as those intended to be covered by IEC61508 and MISRA. Therefore the industry must be self-regulating, and is empowered to perform the development in its own way, focusing its effort in the areas which provide the most benefit and added-value in terms of managing risks. Other industries, such as aerospace, have rigorous external regulation, enforced by the Civil Aviation Authority and the Federal Aviation Authority, which seek to provide a level playing field for all as to how the work must be performed according to standards (DO-178B) and set the priorities in terms of risk. These authorities have the power to refuse certification of a system for use, and therefore must be satisfied before any commercial considerations. In the absence of a world-wide regulator for the development of automotive safety-related software-based systems, each company must consider its own strengths and weaknesses and, in a competitive environment, optimise added-value and risk according to what it perceives the customer is prepared to pay for and accept. In IEC61508 this concept is formalised in the term ALARP, "(risk) as low as reasonably practicable". This is further supported by the common practice of re-using legacy systems developed for earlier vehicles on a new vehicle to keep costs down ("carry-over" systems). Very few new cars launched only contain new systems designed from scratch especially for them.

In the case where a system is not new, but the use of electronic control within it is new, it is reasonable to claim that the design intention should be to make the electronic system at least as safe as the mechanical system it replaces. This can be interpreted as not to increase the overall likelihood of the existing hazards occurring, even if there now may be more potential causes associated with the electronics, therefore ensuring the risk is the same. In reality, each new system should seek to reduce

the occurrence rate of hazards (i.e. improve reliability), and the risk. This is a business reality and not just a safety issue as, put another way, what would be the point of designing a new (electronic) system with worse reliability than the previous (mechanical) one, when the motor industry knows its customers place so much importance on reliability? (Electronic engine management systems, for example, have played a key part in making modern cars more reliable than their predecessors with carburettor fuelling and contact-breaker ignition.)

3.2 · Electronic Throttle Control at Jaguar

3.2.1 What is "Electronic Throttle" ?

Electronic throttle control is sometimes called "drive-by-wire", because it is a system in which a computer controls the throttle butterfly (and thus engine power), rather than directly by a mechanical cable connected to the driver's accelerator pedal. The concept of electronic throttle is not new, but until recently has never found its way into production, even though it has been technically feasible for many years. However, some of Jaguar's competitors have now brought electronic throttle to the market. Electronic throttle control has several inherent benefits, such as the integration of functions like idle speed control, cruise control and traction control, and greater flexibility for achieving good driveability. However, there is much greater potential in the future as a "springboard" technology, enabling new advanced safety features, such as autonomous cruise control, collision avoidance and advanced telematic applications. Jaguar could therefore not afford to be left behind the competition, and electronic throttle has become an accepted part of the company's product plans. A diagram of a simple electronic throttle control system is shown in Figure 5.

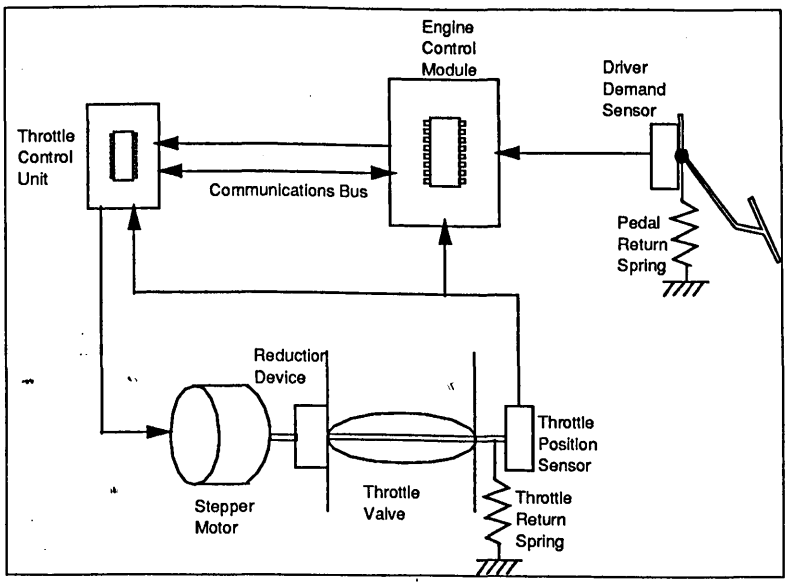

Figure 5: A simple electronic throttle control system

3.2.2 The Denso Electronic Throttle for the XK8

The first application of electronic throttle in Jaguar was on the AJV8 engine for the XK8 sports car, which went into production in August 1996. It was supplied to Jaguar as part of the engine management system by the Denso Corporation of Japan, and has a different architecture that proposed for S-Type/X200. It was novel in that it had a conventional mechanical cable that limits the authority of the electronic control, thus minimising the safety impact of the electronic parts. Around 1992, the time the XK8 project was started, the International Electrotechnical Commission had just published the first draft of what is now IEC61508 [8], recommending independent assessment as a key technique in assuring safety of electronic control systems. Hence, some form of independent assessment of the Denso system needed to be carried out for the XK8. However, Jaguar did not have the necessary expertise, and after some evaluation of likely sources, Lloyd's Register of Shipping were selected as third-party safety assessors. The MISRA Guidelines were still under development during most of this project, hence it was possible to feed the practical experience of dealing with safety-related applications into the MISRA activity, as well as some of the MISRA ideas into the Lloyd's Register work for XK8.

Lloyd's performed a full and comprehensive third-party independent system safety assessment on the XK8 electronic throttle [10]. One of the key lessons learned from this work was the need to carefully scope and focus the assessment on the areas which can benefit most from it. For Jaguar and its suppliers this is mainly the software. Lloyd's did perform a comprehensive review of Jaguar's approach to hardware design, and it was concluded that this was appropriate for ensuring high quality components suitable for safety-critical applications. Hence, it was reasonable to focus any future assessment activities mainly on the safety-related software, as of the hardware safety aspects are well covered by existing engineering processes.

3.2.3 The Ford Full-Authority Electronic Throttle for the S-Type (X200)

The Jaguar S-Type compact saloon, code-named X200, went into production in January 1999, and is powered by V8 and V6 engines. Both engine variants have the Ford PTEC (Power-Train Electronic Controller) incorporating electronic throttle control (ETC). Unlike the previous XK8 system, this is "full-authority", as there is no mechanical limiter, and the electronics alone determine the throttle position, and therefore engine power. Hence, all possible assurance is needed that the new hazards associated with the system are adequately dealt with. This implies a high level of confidence in the integrity of the software that lies at the heart of the system.

The concept of the electronic throttle system under consideration here for S-Type is what is known as a "pedal follower". This is the most basic of throttle strategies, which seeks to control the position of the throttle blade in direct relation to the position of the driver's accelerator pedal. Alternative, more advanced strategies have less direct relation between pedal and throttle position, for example using the pedal as a torque demand, and control the throttle, fuel and ignition at their optimum to produce the required torque depending on prevailing engine conditions. Although ultimately torque control is directionally the future for electronic throttle, as a first step pedal follower is a simpler implementation. In addition, because there is a more direct relationship between pedal and throttle positions, there is greater determinism of throttle position (i.e. it does not depend on other prevailing engine conditions), and therefore opportunity for monitoring correct operation. The system design concept put forward by the supplier for the Jaguar S-Type included a separate monitoring subsystem, with its own separate processor (known as the Electronic Throttle Monitor, or ETM). This was intended to check the position of the throttle blade sensors against an expected position, and would be capable of taking action to force the system to a safer state if a violation was detected. A violation was defined as a situation which would be likely to result in "power greater than demand". It was therefore possible to scope the assessment work on the software for the monitoring subsystem, as sufficient confidence in it would translate to sufficient confidence that "power greater than demand" would not occur. Furthermore, the presence of the separate monitoring subsystem would allow the segregation of the higher-integrity software from the rest of the electronic throttle system.

The "supplier" for the S-Type/X200 electronic throttle is Ford USA. Quotation marks are used because they are an internal supplier. However, during the project, as part of a huge corporate re-structuring, Ford created an organisation called Visteon, who are responsible for the component parts, and are now treated as an external supplier; although overall system engineering responsibility remained with the same Ford group in the USA as before. Ford was keen to gain from Jaguar's experience with XK8 and Denso, to bring full-authority electronic throttle to the market, for future planned application on Ford's global product range. This included learning about independent safety assessment, such as that previously performed for Jaguar by Lloyd's Register [10].

3.2.4 A Risk Based Approach

One important development, which was implemented as part of the preparation for the S-Type/X200 electronic throttle monitor software safety assessment, was a new Jaguar Engineering Procedure that defined a risk policy for electronic throttle, and set targets for the levels of safety required.

Risk is a probabilistic quantity, defined as a combination of how severe a situation is and how often it is likely to happen. For example, when a system fails, if it has the potential to injure 1000 people, but is believed only to be likely to happen once in a million years, then it could be considered low risk. Conversely, if there was something which could injure only 10 people, but that could occur once a year, then it could be considered high risk, as there is a greater overall chance of injury. This is how claims are made, that, for example, living near a nuclear power station may be safer than driving to work. This simple approach, however, is further complicated by underlying social factors which can lead to perceptions of risks being unacceptable, no matter how unlikely they can be shown to be, as the consequences (for society) are so severe.

A risk based approach seeks to put a framework in place for assessing severity and likelihood, and a method of combining them to determine risk. Jaguar's approach was to use a concept taken from IEC61508 [8], which assumes there are four "risk classes":

I. Intolerable risk
II. Undesirable risk
III. Tolerable risk
IV. Negligible risk.

Built into such an approach is the principle of ALARP. ALARP [8] stands for "as low as reasonably practicable", and recognises that there is a point where risk is not as low as it might be, but is generally accepted and acceptable; to reduce to further would incur disproportionate and unreasonable costs.

3.2.5 The Overall Assessment Plan

With the previous experience gained with Lloyd's Register on the XK8 project [10], it was felt that Jaguar could take on the task as a second-party assessor (Jaguar, the customer, assesses Ford, the supplier.) This was the author's role on the project, as the independent safety assessor for the software. Second-party assessment obviously does not have the same level of independence as third-party assessment, but this is not a problem as long as there is sufficient organisational independence. However, it does have some advantages. For example, communication is much better between the assessor and the development team, so it is much harder for the team to hide any deficiencies from the assessor, and, of course, it is more cost effective. Also, as Ford are Jaguar's parent company, there were no problems with confidentiality. This is a very important factor, as full access to all levels of documentation and information is essential to carry out an effective assessment.

As mentioned earlier, one of the key lessons from the Lloyd's assessment work was the conclusion that the integrity of the software was the main issue, as hardware development processes were already well addressed. On this basis, the assessment for S-Type/X200 electronic throttle was scoped to concentrate on the high-integrity software (i.e. the software in the ETM). The author's objectives, as second-party assessor for the S-Type/X200 Electronic Throttle monitoring software, were therefore:

- To oversee the software development to ensure compliance with the MISRA Guidelines;
- To determine the required integrity level, based on the possible hazards;
- To advise on tools and techniques, appropriate for achieving that integrity level;
- To support decision making, by providing reasoned safety arguments;
- To perform independent review, analysis and testing throughout the entire development life-cycle;
- To produce documentation forming evidence of the safety of the electronic throttle monitor software;
- To formally accept the ETM subsystem as fit for purpose on behalf of Jaguar at sign-off.

This was the first time the MISRA Guidelines had been used in such a way, the first time Jaguar had attempted such a detailed assessment of a supplier's processes and work products, and the first time Ford had been subjected to such an activity. Success would depend on striking a balance between what was required and what was achievable, establishing a culture of mutual learning, and on reliance on the author's expertise and credibility as the assessor.

A plan was constructed that proposed that the assessment should consist of 6 work packages, which combined together to cover the MISRA Guidelines [2] as shown in Table 2, and each of which was to be documented by the production of a report. The work packages and respective reports essentially aimed to gather evidence of the design and development of the electronic throttle monitor subsystem software, against the background of the applicable standards and guidelines. Having such a plan was not only good project management discipline, but was also important so that the parts of Ford responsible for delivering the system were made aware of the activities involved, and the impact these would have.

Table 2: How the Assessment Plan relates to the MISRA Guidelines

Assessment Work Package	Relevant MISRA Guidelines Section(s)
1. Preliminary Safety Analysis and Integrity Assessment	3.2.1 Integrity – Introduction 3.2.2 Safety Analysis 3.2.3 Human Factors in Safety Analysis
2. Software Quality Assessment	3.1 Project Planning 3.2.4 Development Approaches 4. Software Quality Planning
3. Requirements Assessment	3.3 Requirements Specification
4. Design Assessment and Detailed Safety Analysis	3.4 Design 5. Emerging Technologies
5. Safety Related Code Assessment	3.5 Programming
6. System Acceptance Assessment	3.6 Testing 3.7 Product Support

The plan attempted to ensure that the assessment was seen as a continuous activity, which allowed for mutual learning by both parties, and which added value to the overall design and development process. The results of the 6 work packages will now be described more fully.

3.3 WP1: Preliminary Safety Analysis and Integrity Assessment

The Preliminary Safety Analysis was conducted using the PASSPORT methodology [3] for the first time. This methodology was originally devised for use on large safety-related transport telematic projects, such as traffic management schemes, which typically involve many distributed computers, communications channels, and megabytes of software. Here, however, it was used on a small, in-vehicle control system, but one which is of key importance to Jaguar (and Ford) to get right, because of its safety implications.

The automotive industry has a long history of using Failure Mode and Effects Analysis (FMEA) [11]. Trying to apply FMEA in the traditional way has some problems; firstly, because it often leads to too much detail (and of course there is uncertainty about that detail at the start of a project); and secondly, there is a difficulty in categorising failures, road/weather condition effects etc. (all outside the influence of the designer), as there is a tendency to look at "worst-case" scenarios and force all failures to the highest level (i.e. FMEA severity 10). This was not helpful. PASSPORT appeared to offer a solution to

these difficulties, firstly through the use of "controllability" categories, and secondly by splitting the analysis into two clear phases, one of which is designed specifically to work in the early, uncertain, concept stage of a project - Preliminary Safety Analysis (PSA). (The other is known as Detailed Safety Analysis (DSA), and is covered later in section 3.6).

A Preliminary Safety Analysis (PSA) is performed as part of the feasibility study when the system concept is being proposed. The objective is to discover whether there are any safety hazards associated with the system, and if so, to identify the top-level safety requirements and the safety integrity levels associated with them. The aim is to identify how the proposed system, known in PSA as the "Target of Evaluation" (TOE), interacts with its environment, and then to discover whether any of these interactions could result in a hazardous situation in the case of a failure of one or more parts of the TOE. The first task is therefore to produce a model that clearly shows the relationship between the TOE and its environment. This model is known as a PASSPORT Diagram, and the one for the S-Type/X200 electronic throttle is shown in Figure 6.

Once the PASSPORT Diagram has been shown to be complete and consistent then the only way that the TOE can effect its environment, for good or ill, is contained within it. Each element of the PASSPORT Diagram is then systematically analysed, and the question "what if ....?" asked (e.g. what if the actuator failed to operate; operated with no command, etc?, or what if the information was corrupted; failed to arrive, etc?). This is a process similar to FMEA, but simplifies it such that it can be used when little detail is known. The effectiveness of this task can be increased by the use of checklists and guidewords, to remind the analysis team of all the modes of operation that the system may undertake. By this means a preliminary hazard list can be built up for the system.

The next step is to perform a "what causes" analysis. This is like a simplified Fault Tree, and aims to discover how each of the hazards in the hazard list might occur, by building up a tree of preliminary events that could lead to the final undesirable event. By analysing the leaves of each tree it is then possible to identify the top-level safety requirements necessary to reduce the risk of the hazard. This is done by examining the areas in the "what causes" tree(s) where a single cause can give rise to a hazard, and by then asking questions such as "what can be done to eliminate this cause?", "what can be added to prevent this cause from causing the hazard?", or "what can be done to reduce the likelihood of this cause?". The answers to these questions can be worded to construct requirements that can be placed on the design, and which are directly related to the control or avoidance of hazards. These are the safety requirements.

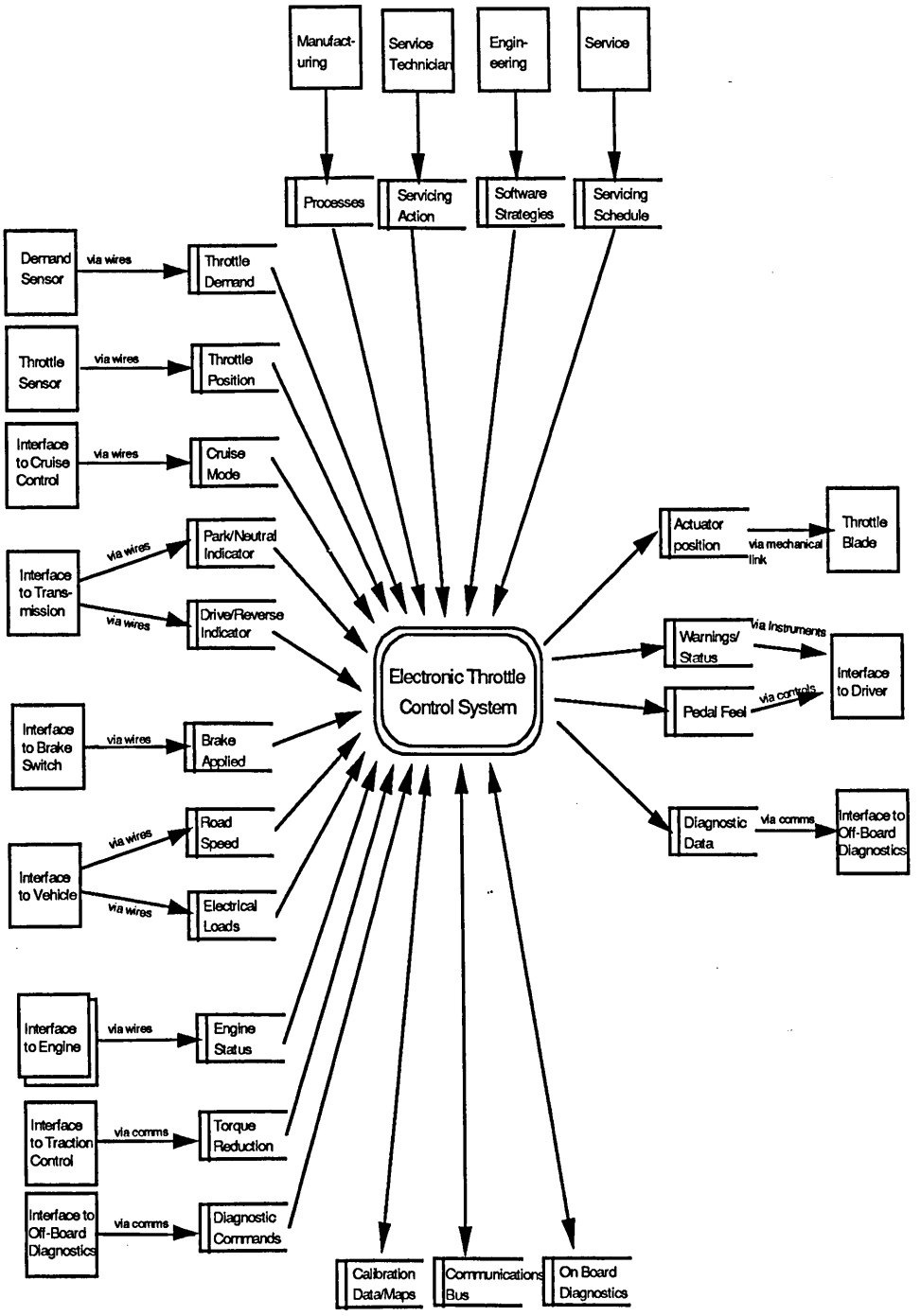

Figure 6: The PASSPORT Diagram for the S-Type/X200 electronic throttle

The PASSPORT concept of "Controllability" provides a way of categorising the severity of the hazards. There are 5 categories: "uncontrollable", "difficult-to-control", "debilitating", "distracting" and "nuisance". They are based on the ability of those involved to control the outcome of a given situation, e.g. they make provision for the ability of the driver to do something sensible to avert an accident. Further explanation can be found in [2] and [12].

Although assigning controllability categories to hazards is still a subjective process, it has been shown to work, by avoiding the difficulties of being too confused about things which cannot be influenced by the design. Even though it is not a true telematic system, the electronic throttle still contains all the same elements, as does any system relying on computer control to perform a function, so it seemed reasonable to apply PASSPORT to electronic throttle. This proved successful, and was highly significant in the early part of the project, when there was some evidence of not knowing how to get started on the detailed specification of safety requirements and design work, both at Jaguar and at Ford.

Working through the PSA using the process described earlier, a set of safety requirements were produced which were much more clearly defined than before, and existed prior to any design work being done. These were incorporated in the system specification issued by Jaguar to the supplier organisation at Ford. With the worst-case hazard described as "unintended vehicle acceleration", categorised as "difficult-to-control", the highest integrity level for electronic throttle was set at 3 (MISRA levels are 0 to 4).

The most significant safety requirement identified was the one which confirmed the need for a separate sub-system for monitoring safe throttle operation. The monitoring subsystem, which although it added a significant cost to the system and required its own dedicated engineering resources, was a major step forward for providing the customer with protection against "unintended vehicle acceleration". Having a documented Preliminary Safety Analysis was a key factor in the justification for this extra cost and effort in order to achieve a safety benefit. The result was a separate 16-bit microcontroller, known as the ETM (Electronic Throttle Monitor), whose only function is to monitor the operation of the rest of the electronic throttle system, enveloping all other system elements. It specifically covers the "unintended vehicle acceleration" hazard, and can take action to mitigate the effects of such a condition if it is detected. This has proved to be a major success, as the ability to segregate all the highest integrity software into the ETM, was a good way of ensuring that a software development process, suitable for level 3, was followed. This also facilitated the assertion that all other lower-level hazards were containable within the rest of the system. Furthermore, the rest of the system, as it is subject to the standard quality control measures, could be assumed to be developed to an appropriate integrity level, and need not be considered further during this assessment project.

Finally in this work package, a clause-by-clause check was done against the recommendations in the appropriate sections of the MISRA Guidelines (as identified in the assessment plan). This is summarised in Table 3.

Table 3: MISRA compliance summary table for Work Package 1

MISRA Guidelines Topics	Section	Compliance Judgement
Preliminary Safety Analysis		
Integrity – Introduction	3.2.1	Full
Integrity - Safety analysis	3.2.2	Full
Integrity – Human Factors	3.2.3	Not applicable, no special human-machine interfacing issues for the ETC system (or ETM).

3.4 WP2: Software Quality Assessment

As a result of the PSA, a small team was created at Ford USA, separate from the rest of the organisation doing software for the other parts of the engine management system, and given the sole responsibility for the ETM. The team's first task was to establish a process to deliver the required integrity in the software of the ETM. It would not have been possible to use the existing powertrain software development process at Ford, as it was not set up for handling software which could be considered capable of more than integrity level 1.

The team defined and documented a process they believed to be capable of achieving integrity level 3, specifically for the ETM. Using the MISRA Guidelines [2], a detailed clause-by-clause, assessment was performed both of the process they had documented, and of how the team implemented it in practice. To assess the working practices of the team, the software quality management principles of ISO 9000-3 [13] and TickIT [14] were applied. The main issues were:

- Was the project control adequate?
- Did the process contain the right steps in accordance with MISRA Integrity Level 3?
- Was the working practice in-line with expectations against ISO9000-3/TickIT?
- Was there a safety plan?

The first issue was mainly concerned with the wider project issues and the whole programme management process, within which the ETC and ETM formed small parts. It proved useful to step back up to a higher level, to check that the influences on the ETM coming from higher authority were still compatible with the integrity required. A key document was the Product Manual, which is produced by Jaguar's Programme Office. They are responsible for the definition of the project, and provide the authority for the work within the individual engineering departments. The Product Manual should

reflect the overall concept. It was therefore important to check on what it said with respect to electronic throttle, to ensure the emphasis on safety was correctly represented.

The second question is at the heart of this piece of work, and the definitive reference is "Table 3" from the MISRA Guidelines [2], which summarises on one page the type of process elements required for each integrity level. The first time this was reviewed, it became apparent that one key item was missing - "formal analysis of the safety functions". This is interpreted as the application of a recognised formal mathematical method to those parts of the software which provide the safety protection - in this case the ETM software. A plan was therefore required to address this situation, and given the lack of expertise available in both Jaguar and Ford, it was necessary to look to a third-party to buy-in this expertise. Furthermore, in order to avoid unnecessary delay and disruption to the rest of the ETM development, it was decided to make the application of formal methods part of the assessment, under the next work package looking at requirements specifications. This proved a wise and successful strategy, enabling the ETM team to continue relatively unaffected, whilst at the same time gaining the additional assurance from the formal analysis, thus filling the gap in the process.

Another, less important, deficiency was the lack of the use of process metrics, although it was discovered that plenty of data is routinely gathered during development. Rather than attempt to "track" live data, which would be burdensome and of limited use, the tactic was to strengthen the work performed in the last work package of the assessment (system acceptance), towards the end of the development, when sufficient mature data specific to this project was available.

The third issue required the use of audit techniques, such as those use in ISO 9000-3 [13] and TickIT [14], to uncover the actual working arrangements of the team. A detailed walkthrough of the change control and configuration management mechanisms was performed. A major issue uncovered here was the fact that the whole Ford organisation was in the process of moving all its configuration management from the old CMS tool on a VAX, to a new tool, called ClearCASE, under UNIX. Whilst the end result was a vast improvement, cutting down on the complex file transfer and conversion steps currently necessary, any change part way through a project was considered to represent a risk. It was therefore necessary to monitor the situation until the transition was complete. Another issue worthy of mention was the timing plan of the process deliverables. There was some concern that, unless a formal plan was produced, committing named individuals to produce each of the deliverables mentioned in the ETM team's documented process, then there was a risk that some items would be forgotten or left too late. A plan was eventually issued, and was reviewed periodically jointly with Jaguar, to ensure all the key deliverables were achieved.

It was also noteworthy that auditing against the MISRA requirements for independence in the verification and validation activities, led to Ford making a new appointment, allocating the additional resource uniquely to these tasks for the ETM. This provided further evidence of the success of the assessment, in guaranteeing the required level of commitment from the project team and its management. Also, not unusually for any audit, some tightening of the review and record keeping was advised. In all, 11 "Observation Records" were raised as a result of audit, each of which forced an improvement to the process.

Finally, the safety plan. Safety plans are one of the major requirements of the generic safety standard IEC61508 [8]. However, they are rarely mentioned in the MISRA Guidelines [2], and there is little clue as to what such a plan might contain. It was therefore necessary to go back to IEC61508 [8], which includes a contents list for a safety plan. There was no provision in the ETC process for the production of a separate safety plan, so the assessment task became one to ensure that the items on the contents list were covered somewhere in the project documentation.

Overall, this work package was successful in ensuring that the ETM team defined a series of tasks for themselves which were consistent with MISRA integrity level 3. Careful attention to detail, and a willingness to continue to press for issues raised during audit to be resolved, in a firm but co-operative way, were key factors which contributed to the success, and ensured that the team followed their plan in practice.

A summary of compliance against the appropriate MISRA sections is shown in Table 4.

Table 4: MISRA compliance summary table for Work Package 2

MISRA Guidelines Topics	Section	Compliance Judgement
Software Quality Planning		
Project definition	3.1.1	Full
Lifecycle plans	3.1.2	Sufficient, after deficiencies corrected
Planning for verification and validation	3.1.3	Sufficient, after deficiencies corrected
Assessment	3.1.4	Full
Re-use	3.1.5	Not applicable, the ETM is all-new.
Development approaches	3.2.4	Sufficient, after deficiencies corrected
Management Responsibilities	4.1	Full
Education and Experience	4.2	Appropriate, part of Human Resources
Human factors in software development	4.3	Not assessed, inherent in organisation
Standards and accreditation	4.4.1	Sufficient, after deficiencies corrected
Checklists	4.4.2	Checklists not used explicitly

MISRA Guidelines Topics	Section	Compliance Judgement
Assessment of compliance	4.4.3	The purpose of this assessment project
Changes in production	4.4.4	Covered by supplier quality control
Software process metrics	4.4.5	No, deferred to work package 6
Documentation requirements	4.5	Sufficiently met
Subcontracting	4.6	Partial, due to corporate organisation

3.5 WP3: Requirements Assessment

The purpose of the requirements assessment was to examine the specifications hierarchy for the system, looking not just for omissions, errors and ambiguities, but also for traceability. One of the most important aspects here was firstly to identify the sources of requirements at the different levels, and to ensure that the safety requirements in particular, such as the need for the ETM, were traceable throughout. As in the previous work package, it was necessary, by following the MISRA Guidelines, to step back somewhat to a higher level, in this case to the specification which captured the whole of the S-Type/X200 vehicle electrical system. This was complicated by the fact that S-Type/X200 is a shared platform with a Ford vehicle programme, DEW98 (the new Lincoln LS), requiring a check that no conflicts existed. In fact electronic throttle control (ETC) is a unique feature for S-Type/X200, and therefore there were no conflicting requirements on it coming from the DEW98 programme. It was also confirmed that the specifications included the requirements recommended by MISRA for assisting robustness against noise and electrical interference (EMC/RFI). In addition, in reviewing the ETC system specification, some of the mechanical aspects were appraised although no specific guidance as to what was expected was available in MISRA, so any judgement made was in the context of normal Jaguar design practice.

Perhaps the most important of the requirements specification documents for software safety was the one known as the Electronic Throttle Monitor Subsystem Design Specification, or "ETM SDS". Despite its name, it actually was the software requirements document for the ETM. It therefore seemed reasonable, having taken account of the other documents, to focus in on this one. It had, at the time the original requirements assessment was performed, been subjected to two very thorough formal reviews conducted by a cross-functional team (with independent members), and subsequently continued to undergo review until the project was completed.

As was mentioned earlier, it was identified that it was necessary to apply some formal mathematical methods to the software safety functions, in order to gain initial assurance. After an invitation to tender

was issued to six potential suppliers, York Software Engineering (YSE) Ltd. were commissioned to work on a formal specification for the ETM, and to perform some analysis on it. This was the first time anyone in Jaguar, or Ford, had proposed the use of such techniques on a production project. YSE Ltd. used the ETM SDS as a basis, and converted it into formal notation using the Z language [15]. In doing so, they raised 100 new issues, which had not at the time been found in reviews. These issues were either ambiguities, points of clarification or straightforward errors. They were passed on to the ETM team to correct the ETM SDS, which remained as their main reference for the design. Doing this ensured that the benefit of using formal methods could be claimed and demonstrated, whilst at the same time not disrupting the team with something they did not have the capability for at that time. YSE Ltd went on to use the formal specification of the ETM in two ways. Firstly, to perform some analysis, using their CADiZ tool to discharge some basic proofs, and secondly to create an executable animation of the specification in ADA. Both of these exercises were of limited success. The proof analysis was judged only partially successful, mainly because it was very hard to decide what to prove and what not to prove (clearly not everything can be done, given limited time and resources); and also because the proofs became large and difficult, which resulted in CADiZ memory constraint failures. The original plan for the animation was to generate the ADA code automatically from Z using another prototype YSE tool (StZ). However, this proved unworkable due to the immaturity of the tool, and the ADA had to be manually coded, but working directly from the Z specification. Once coded, the animation was somewhat more successful than the proof exercise, showing up what was believed to be a previously undetected error in the cruise control portion of the ETM specification. However, when raised, it transpired that this error had already been found by the team in the USA using conventional testing techniques.

It was therefore possible to conclude that overall it had been useful to apply formal methods to this project, but not as definitively useful as advocates of the methods claim. The most value came from the creation of the formal specification itself. However, it was a commonly held view within the development team that there was no basis for asserting that any of the 100 issues, uncovered by the formal method, were not all detectable in other ways, such as by review or prototype testing. The analysis showed the main weakness of the technique, that it was relatively immature, and tool support was inadequate. At the time of writing, although improvements are being made, this is still the situation. The animation was a partial success, but again there was strong suspicion that it offered no additional value than conventional prototype testing.

Throughout, in assessing against the MISRA Guidelines, it was found that the Guidelines themselves are far from perfectly suited to such use. Many of the clauses are not recommendations at all, but are

simply information, whilst it is not always clear whether others are mandatory or "nice if's". Furthermore, neither is it obvious as to how the individual clauses relate to the integrity level. Clearly there is a relationship, but it is not specified. These issues were overcome by using a engineering judgement and expertise, a lot of which was gained from being involved in writing the Guidelines in the first place.

A summary of compliance against the appropriate MISRA sections is shown in Table 5.

Table 5: MISRA compliance summary table for Work Package 3

MISRA Guidelines Topics	Section	Compliance Judgement
Requirements		
Whole vehicle architecture	3.3.1	Sufficient
Vehicle control systems	3.3.2	Not applicable
Noise and EMC	3.3.3	Full
Verification and validation	3.3.4	Full
Tools and techniques	3.3.5	Full

3.6 WP4: Detailed Safety Analysis and Design Assessment

The purpose of this piece of work was to assess the design aspects of the S-Type/X200 Electronic Throttle Monitor (ETM) against the requirements of the MISRA Guidelines. It followed directly on from the previous assessment of the requirements and specifications. The design was looked at in two ways. Firstly, a Detailed Safety Analysis (DSA) examined the ETM in the context of the complete electronic throttle system design using the PASSPORT methodology; and secondly, a clause-by-clause assessment of the ETM software against the "Design" section of MISRA. A schematic showing the main components of the hardware design is shown in Figure 7.

Although the main focus of the assessment is the ETM software, it was important to consider the whole system concept to understand where the ETM fitted in and that it's requirements covered all the necessary failure modes of the hardware. This was the purpose of the Detailed Safety Analysis (DSA), carried out using the PASSPORT methodology [4]. PASSPORT DSA uses a concept known as the PASSPORT Cross, which requires two models of the system, a functional model and a physical model, to be related together. The idea is that the functional model is predominantly software, consisting of processes (or Functional Elements - FEs) and data (or Information Sets - ISs); and the physical model is predominantly hardware, consisting of components (or Physical Elements - PEs) and interconnections (or Communications Facilities - CFs).

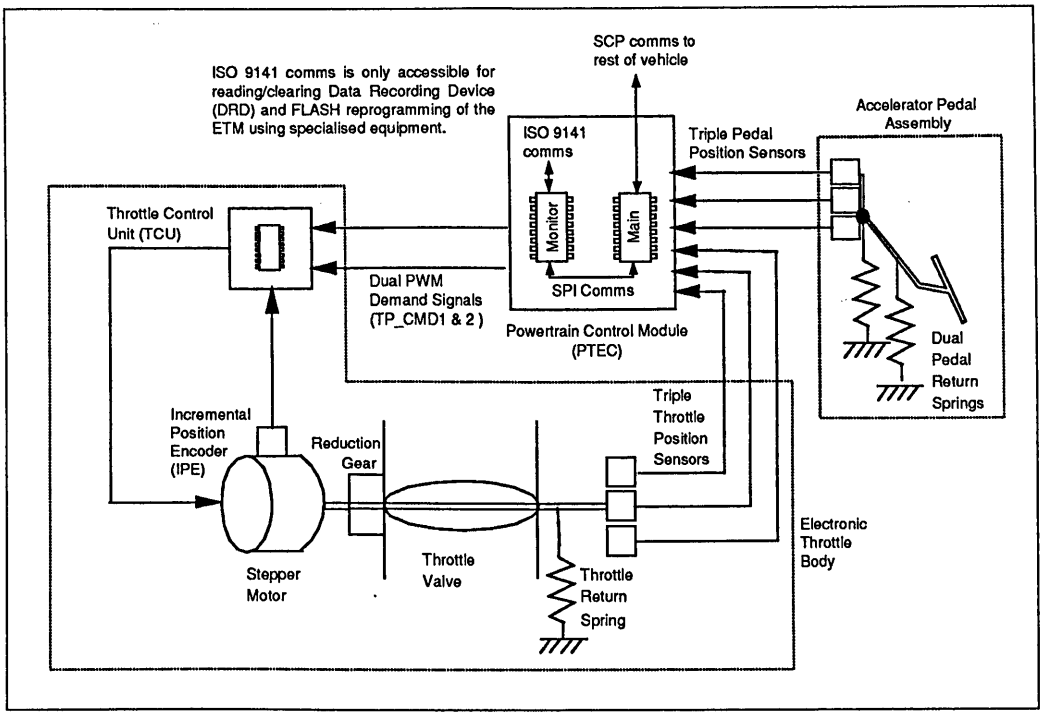

Figure 7: A schematic of the main S-Type/X200 Electronic Throttle Components

For the electronic throttle system, the physical model was constructed as a series of sketches, similar to circuit diagrams, starting at a high level of abstraction and decomposing down towards individual electronic and mechanical components. To develop the physical model it was necessary to study the system design in considerable detail, including the hardware design for the mechanical, electrical and electronic components, plus the module and vehicle circuit diagrams. During the course of the work, it became apparent that it would be necessary to extend the PASSPORT method as described in [4], to handle the mechanical components and "mechanical communication facilities" (a mechanical joint or connection which could transmit a force). This knowledge was used to feed into a European Commission Framework IV supported project in which the author is a partner, called COMPASS, which is developing a computer based tool to implement the PASSPORT methodology [16].

The functional model was captured with a conventional CASE (Computer Aided Software Engineering) tool, using the Yourdon structured analysis methodology [17] as a guide. It started at a level of abstraction which defined the functions of the electronic throttle system as a whole, and worked down towards the functionality specific to the ETM.

Whilst it proved valuable to go through the physical and functional modelling steps, as this greatly assisted in the author's understanding of the system design, the next step of the PASSPORT DSA, the construction of the PASSPORT Cross, proved less successful. The PASSPORT Cross is made up of two matrices which represent the connectivity of the physical (PEs and CFs) and functional models

(FEs and ISs), and two further matrices that project the two models onto one another (PEs to FEs, and CFs to ISs). The matrices were built manually on a conventional spreadsheet, and it soon became apparent that it was too error prone and tedious to be of any useful value. This was due to the size and complexity of the matrices, such that it became virtually impossible to visualise and manipulate them manually. Therefore, it was shown that full implementation of DSA, even on moderate sized systems, is impractical without computer tool support. This was the main reason for the author's interest in the COMPASS project [16], which, if successful in computerisation, would make the PASSPORT Cross more automatic and hide the detail, whilst retaining the usefulness of the method in mapping the physical model to the functional model, and vice versa, enabling detailed completeness and consistency checking of the design. For the electronic throttle, although an attempt at the PASSPORT Cross was made, it was decided not to use it for the next stages of the PASSPORT DSA methodology, FMEA and FTA.

The FMEA was based instead on the physical model, taking each physical element, and then each communication facility, in turn and going through a traditional FMEA process [11]. The functional model therefore had no direct input into the analysis, however the very fact that it had been constructed gave increased knowledge and understanding of system operation, which was used indirectly. The FMEA showed that failure of any the components led to a safe default mode of operation, although one new, previously unconsidered failure mode was uncovered. This was associated with the reverse switch, used on manual gearboxes to allow the system to select an appropriate throttle progression (implemented as a pair of look-up tables against pedal position, one for reverse and one for all other cases). A new safety requirement was placed on the design and development team, to ensure that the failure mode of the reverse switch was suitably dealt with. As a result a single look-up table was defined, thus completely removing the undesired effect associated with the failure mode.

The Fault Tree Analysis (FTA) [18] was constructed using the "What-If" analysis from the Preliminary Safety Analysis part of PASSPORT performed earlier, together with the knowledge gained from the physical and functional modelling work. The "top" event was defined as "unintended vehicle acceleration", and a series of AND and OR gates were used to break this down to combinations of individual component failures. Using a specialised fault tree software package, it was possible to calculate the minimal "cut sets" of the system (the minimum combinations of "bottom" event that are necessary to cause the "top" event. This showed that there were two single point failures (the pedal pivot and sensor wiper), both giving the same effect, resulting in erroneous input to the system. This was due to the fact that there can be only one accelerator pedal, which if it jams (can be the pivot or the wiper) will result in the system believing the driver has his/her foot on the pedal. Such a failure mode is

present and accepted even in a conventional mechanical throttle and cannot easily be eliminated. Therefore, it can be considered outside the scope of the electronic throttle system. Other than this special case, the FTA confirmed that there were no other single points of failure.

It is theoretically possible to quantify fault trees with failure rates for all the components, and calculate the probability of the top event. However, in this case this was not possible, because such calculations are sensitive to the failure rates used, and, as the S-Type/X200 throttle system is new, very little component reliability data was available. However, as an exercise in order to follow Jaguar's risk based approach, an attempt was made at calculating the top event probability in the fault tree using the following assumptions. Making the assumption that the MISRA integrity levels map directly onto the IEC61508 levels (which is not necessarily true), software failure rates were added using the reliability targets suggested by IEC61508 [8]. Furthermore, it was asserted that the software could detect most of the hardware failures, and certainly all the critical ones. All other failure rates were set to zero. The probability of the top event ("unintended vehicle acceleration") was calculated and was in the right "ball park" for claiming that the risk associated with electronic throttle system was "Negligible" (Risk Class IV). Thus, although it was recognised that the calculations and assumptions are easily challenged, because of the absence of any suitable data, it was still considered better to make an attempt than to ignore it altogether.

The DSA served to confirm that the overall integrity level was still 3, because no new hazards were uncovered, and that the ETM could provide protection against all the failure modes, except the special case of the accelerator pedal mentioned above. For the rest of the design assessment, the ETM software was assessed against each clause of the MISRA guidelines. This led to several areas of more detailed assessment, namely the SPI (Serial Peripheral Interface) inter-IC communications between the ETM processor and the main PTEC processor, the "task manager" which schedules all the ETM software functions, and an assessment of the techniques for the robust use of interrupts. The design of the ETM software was captured in two main documents produced by the design team. These were the ETM Software Requirements Specification ("SRS"), which despite its name was the high-level software design in Hately/Pirbhai notation [19], and the ETM Software Detailed Design Specification ("SDDS"), which provided the low level design information for each of the ETM software modules, organised in a series of structure charts [20]. These documents were traced back to the specifications covered in the requirements assessment, whilst the traceability downwards was recorded and managed by the design team.

In conclusion, the DSA and design assessment did uncover some areas which required attention, such as the need to add a further error state to the SPI handler software (to account for the situation if the PTEC sent too many bytes), and the application of more defensive programming techniques to the task manager. However, overall, the design of the ETM software, in the context of the whole system, was deemed to be sound and appropriate for MISRA integrity level 3.

A summary of compliance against the appropriate MISRA sections is shown in Table 6.

Table 6: MISRA compliance summary table for Work Package 4

MISRA Guidelines Topics	Section	Compliance Judgement
Design		
Real-time implications	3.4.1	Sufficient. ETM software design accommodates measures which ensure effective real time performance.
Floating point arithmetic	3.4.2	Not applicable. ETM is fixed point only
Modelling	3.4.3	Not applicable. Although an animation was produced as part of the formal methods work.
Optimisation and adaptive control	3.4.4	Not applicable. The ETM is not a control system and requires no optimisation.
Communications and multiplexing	3.4.5	Full
On-board diagnostics	3.4.6	Full
System security	3.4.7	Full
Fault management	3.4.8	Full
Design for verification and validation	3.4.9	Full
Tools and techniques for design	3.4.10	Full
Emerging Technologies	5	Not applicable, only conventional design methods used.

3.7 WP5: Safety-Related Code Assessment

The Electronic Throttle Monitor (ETM) acts to place an envelope of safety around all the other components of the system. Malfunctions of any software elsewhere in the system, which result in the throttle opening and causing an unsafe situation, will be detected and acted upon by the ETM. Therefore it can be assumed that all the safety integrity level 3 code in the system is in the ETM, enabling the appropriate level of attention to be paid to achieving the required integrity for this software. As the ETM is a relatively small component within the system, with an appropriately small development team (3 full time engineers - one ETM system engineer, one ETM software engineer and

one ETM test engineer), it was a manageable task to seek to employ the appropriate software development techniques.

One of the key factors during the coding phase was the choice of the C language. Some experts have been critical of C for having many weaknesses fundamental to the syntax and semantics of the language itself [21]. (These experts are usually advocates that ADA is the only choice for safety critical software). However, there are also many reasons for C being a good choice, particularly for automotive systems. For example, firstly, despite concerns about it, it is widely used and well understood - meaning that there is a pool of experienced C programmers. A good and competent C programmer is better, and safer, than a inexpert and inexperienced ADA programmer, no matter how good the language. The ETM software engineer, who was responsible for the coding, was suitably experienced and highly competent. Secondly, there is excellent tool support for C, which is not the case for many other languages. Tools not only help productivity, but also greatly assist in raising the confidence in the correctness of the software. Two key tools used for the first time within Ford on the ETM were Flexelint from Gimpel Software Inc, a static analysis tool, and CANTATA, a dynamic (and static) test automation tool from Information Processing Ltd. A third, and very compelling, reason for using C is the availability of compilers for the target processor (an Intel 80C196 for the ETM). Often C is the only available compiler for the smaller, embedded, microprocessors. (For the 80C196, Intel only list three compiler vendors, each of which only offer C.). C produces very efficient target code once compiled, and its instructions are well suited to low level manipulation such as that required to drive on-chip I/O peripheral circuitry (e.g. timers and counters). Finally, although C is known to have weaknesses, there is an argument to be made that, provided these weaknesses are known, then they can be avoided, or at least carefully inspected to check that no problems have arisen because of them. This is the role of what the MISRA Guidelines [2] calls "codes of practice".

The safety-related code assessment therefore focussed on the C source code of the software in the ETM, using the MISRA Guidelines [2] as a checklist of what to expect of the coding phase. The three areas covered by the guidelines are recommendations on the need for "codes of practice" for programming, verification and validation of code, and appropriate tools and techniques. All three areas were well addressed by the development team, and they showed a willingness throughout to ensure that full compliance was achieved. "Codes of practice" is particularly relevant when using C, for the reasons mentioned above, and it was important to confirm that appropriate measures were taken to address the known weaknesses of the C language. MISRA published its "Guidelines for the Use of the C Language in Vehicle Based Software" in April 1998 [22], which was too late to seek formal compliance for the ETM code. However, they were reviewed by the ETM team, and it was confirmed

that there were no concerns arising with the recommendations of MISRA C. Also, special attention was paid to the compiler used. Whilst it was the one recommended by Intel, the ETM microprocessor supplier, and is recognised as industrial strength, no formal certificate of compliance existed for it. Therefore, Ford were proactive in exercising the compiler with Plum-Hall validation suites to confirm that, whilst there were issues, none of these directly affected the software constructs used in the ETM.

Throughout the programming phase, reviews were carried out on all the code produced. These reviews were used to compare the code implementing each function to the set of documentation aligned with it, and were completed before the code was submitted to final, formal unit testing. In addition to manual reviews, a static analysis tool was used to analyse all software source code for anomalies and errors which humans often find difficult to spot (e.g. variables declared but not used; variables used without being defined or initialised; data type mismatches (automatic conversion); and unreachable code branches). The approach taken with static analysis was to have all the checking options enabled, and to carefully examine situations where warnings occurred. Code was then either modified to correct the statically detected anomalies, or the anomalies were inspected to ensure that they were deliberate and justifiable. By performing static analysis prior to code review, the human inspection effort was able to focus on those aspects which humans are good at, e.g. appropriate variable naming, understanding of comments and code function, etc.

Furthermore, as a means of probing deeper into the code, a McCabe complexity analysis [23] was performed, followed by some additional dynamic unit testing using the CANTATA test tool. Both of these activities were performed by the team in the USA, but in order to provide additional confidence, it was appropriate for the assessor to check the results independently. However, the intention was not to repeat the work done in the USA, but rather to add to it. The purpose of the McCabe complexity analysis was firstly to confirm that no C code modules had a complexity greater than 10. The MISRA Metrics Report [24] recommends that the maximum McCabe cyclomatic complexity should be 15, however it was agreed that 10 would be maximum for this project (the higher the number the higher the complexity). The second objective of the complexity analysis was to select a reasonable number of modules on which to focus additional dynamic testing effort. It is generally accepted that the more complex a software module is, the more likely it is to contain errors. Hence, it is logical to use McCabe to select a sample of the more complex modules to submit to repeated and extended unit testing using the CANTATA automated test tool.

It was decided to select the nine most complex modules, which was all those with a McCabe cyclomatic complexity number of 8, 9 or 10. The unit testing using the same test cases as used by the US team was fist repeated to confirm the results. Even this provided some additional confidence as the US team used a UNIX platform, whereas the repeated work performed at Jaguar used a Microsoft NT platform, thus reducing the likelihood of any operating system or compiler defects leading to inconsistent results. The test cases were then extended to achieve a higher level of coverage of the source code. The original test cases were designed to achieve 100% decision coverage, i.e. every branch in the code is executed. However, for the nine selected modules, where necessary, the test cases were added to achieve 100% condition, or Boolean, coverage. This is a tighter coverage metric than decision coverage, as it requires all true-false combinations of Boolean expressions to be executed. Only five of the nine modules had any Boolean expressions in them, and therefore could be extended, and for two of the modules it was not possible to achieve greater than 87% Boolean coverage. This was because of mutually exclusive Boolean expressions in the code. Although this may not seem ideal, it was decided that attempting to manipulate the logic within these modules to remove the mutually exclusive parts would be likely to result in code which was more difficult to understand than the original. Ease of understanding is a more important attribute than a somewhat esoteric measure such as Boolean coverage.

Overall, the programming phase for the safety-related, i.e. ETM, software of the ETC system was judged to be in accordance with the recommendations of MISRA, and in-line with expectations for a safety integrity level 3 development.

A summary of compliance against the appropriate MISRA sections is shown in Table 7.

Table 7: MISRA compliance summary table for Work Package 5

MISRA Guidelines Topics	Section	Compliance Judgement
Safety-Related Code		
Codes of Practice	3.5.1	Full
Verification and validation of code	3.5.2	Full
Programming tools and techniques	3.5.3	Full

3.8 WP6: System Acceptance

This final work package aimed to pull together all the testing activities performed on the system which were pertinent to the Electronic Throttle Monitor (ETM) and its software, and to tie up a few of the

remaining "loose ends" from the MISRA clause-by-clause review, namely recommendations on "off-board diagnostics" and "software maintenance". This led to two fairly major pieces of work, a set of safety validation tests and an analysis of some process metrics (which was originally deferred from work package 2 – "Software Quality Assessment"). The MISRA recommendations on testing were adequately met for integrity level 3 by the efforts of the team through a series of test phases. From the "bottom-up", these were:

- Unit testing of C source modules. This was performed on a UNIX host using the CANTATA automated test tool from IPL. The decision by the Ford development team to use CANTATA was reached following Jaguar's initial experiences with it. 100% decision coverage was achieved (supplemented, where possible, by 100% Boolean coverage for the nine most complex modules, as part of the safety-related code assessment work package). Test cases were derived from the ETM module specs in Software Detailed Design Specification (SDDS) and Software Requirements Specification (SRS). It used a "black box" testing approach, followed by "white box" to achieve 100% decision coverage. Boundary value test cases were included.

- ETM Functional Testing. This was performed on the complete PTEC Powertrain Control Module, with simulated sensor signals on the bench. 100% requirements coverage of the ETM software functions was achieved, based on test cases derived from the ETM System Design Specification ("SDS") and interface specifications. It also included stress testing, by pushing input signal frequencies to their limits.

- ETC FMEM (Failure Mode and Effects Management) Testing. This was performed on a vehicle, to confirm all the FMEM functions operate as expected at the system level. The test cases were based on the ETC system FMEA, and confirmed the correct flagging of fault codes and warning lights.

- ETC Safety Validation Tests. This phase was planned independently by Jaguar, and carried out jointly by the US team and Jaguar personnel at Jaguar's local test track at MIRA. The main objective was to validate the safety requirements, confirming that "unintended acceleration" under all failure conditions was less than a predefined figure (although some further refinement and interpretation of this simple criterion was necessary). Ultimately, it was the judgement of the test team to confirm that the vehicle behaved in a safe manner and that the hazard mitigation assumptions were valid. There were 3 key parts to the tests:

1. To evaluate single points of failure, somewhat similar to the FMEM tests mentioned above, except that all tests were performed dynamically in 3 driving conditions (idle in park, drive at 60kph and in cruise control at 60 kph).
2. To evaluate multiple points of failure, which are combinations of single points of failure occurring together. Not all combinations were tested as this was not possible due to the number of permutations.
3. To confirm the operation of the ETM software, and assess vehicle behaviour, in particular to quantify "unintended acceleration", when an induced "Main" processor "failure" leads to an ETM intervention. "Failures" were induced by manipulating the Main processor calibration on-line to open the throttle more than normal.

The results were written up in detail as part of the "System Acceptance" work package (as well as in an official Jaguar technical report), as this phase of testing represented the final comprehensive evaluation of safe system operation in the presence of failure before the system was committed to volume production.

All safety-related test phases were matched with the appropriate level of specification or design documentation, which can be visualised on the familiar V-model, as shown in Figure 8.

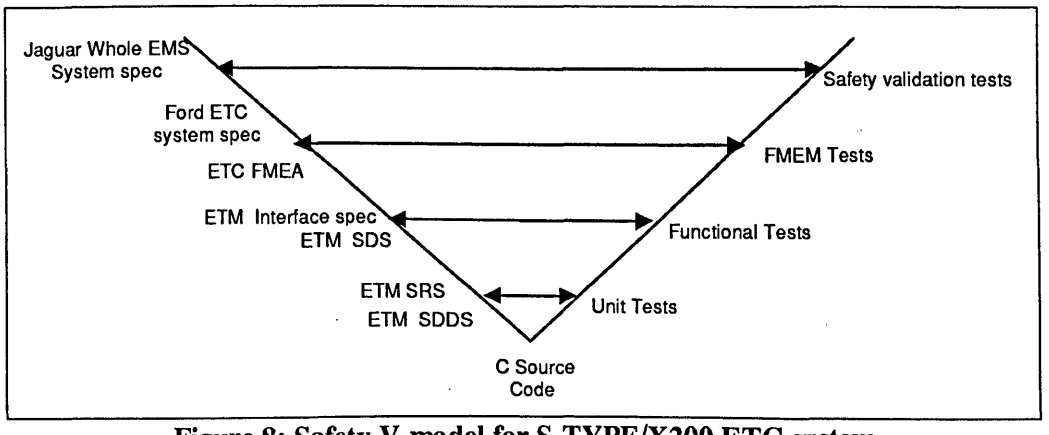

Figure 8: Safety V-model for S-TYPE/X200 ETC system

Finally, an analysis of some of the process metrics was performed. The data used was available as a consequence of the development process itself, but it did take some effort to organise access to it in a form which could be presented graphically. Metrics were analysed in the following categories:

- Software changes for the "Main" processor;
- Software changes for the "ETM" processor;
- ETM software product metrics history, e.g. McCabe complexity over time;
- Major hardware changes for ETB, TPS, Pedal ass'y and TCU (includes TCU software);

- Jaguar concerns tracking for prototype and pre-production vehicles;
- ETC related documentation;
- Number of X200 prototype vehicles built.

Overall, the metrics analysis showed that the safety-critical ETM process was more responsive than the "standard" Ford powertrain software process used for the "main". In particular the "Software Concern Report" (SCR) procedure, adopted locally by the ETM team following an earlier assessment observation, proved an efficient and effective way of tracking all ETM related software issues raised during review and testing. The effort put into ETM documentation, formal methods analysis and code reviews was born out by the small number of specification related defects which were uncovered during the later stages of development. It was also possible to see the shift in focus during ETM development from adding new functionality, to refinement, through until final "tidy up" before sign-off. It was also good to see a remarkable degree of stability in the ETM, as measured by code size and number of modules, for at least the last year development when most of the verification and validation activities were carried out. The concerns tracking data from the Jaguar prototype fleet did show a steady increase in the number of concerns being raised, although this has to be taken in conjunction with the number of prototype vehicles in the fleet which also increased over time. It is reasonable to suggest that the incident rate of concerns per prototype-hour was decreasing over time, although it was not possible to collect data to confirm such a suggestion.

A summary of compliance against the appropriate MISRA sections is shown in Table 8.

Table 8: MISRA compliance summary table for Work Package 6

MISRA Guidelines Topics	Section	Compliance Judgement
<u>System Acceptance</u>		
General testing	3.6.1	Full
Dynamic test	3.6.2	Sufficient
Integration test	3.6.3	Not directly applicable. No separate integration test phase – part of "functional testing".
System test	3.6.4	Full
Tools and techniques for testing	3.6.5	Sufficient
Off-board diagnostics	3.7.1	Full
Software maintenance	3.7.2	Full
Software process metrics	4.4.5	Deferred from work package 2. Sufficient

Ian Kendall
Jaguar Cars

3.9 Overall Safety Justification Statement

The independent assessment of software for the S-Type/X200 electronic throttle monitor (ETM) was carried out across 6 work packages, over the whole project lifecycle from concept through until final sign-off for volume production. The role of the assessor was to act as an advocate for prospective customers, to ensure that the system exhibited a level of safety that they are entitled to expect. The 6 work packages and their headline conclusions were:

1. ***Preliminary Safety Analysis.*** This used the PASSPORT PSA methodology to look at the system concept and determined the overall safety integrity level (SIL) to be equivalent to MISRA level 3, i.e. the worst case hazard "unintended acceleration" was classified as "difficult-to-control", according to MISRA's controllability categories. The key safety requirements were derived, including the need for a separate monitor processor, which became known as the ETM. All other lower-level hazards were asserted to be containable within the rest of the system which was developed to an appropriate integrity level.
2. ***Software Quality Assessment.*** This used the MISRA Guidelines [2] as the main reference, supplemented by IEC61508 [8] in certain specific areas, to define and confirm (using ISO9000-3 [13] and TickIT [14] audit techniques) a software development process for the ETM which would meet Safety Integrity Level 3.
3. ***Requirements Assessment.*** This reviewed all the specification documents pertinent to the ETM software, to confirm that all were well defined. It was supplemented by the application of Formal Methods (Z), which analysed the main ETM specification for correctness and completeness, including an attempt at some proof of safety properties.
4. ***Design Assessment and Detailed Safety Analysis.*** This confirmed the MISRA design recommendations were sufficiently followed, and applied the PASSPORT DSA methodology to look at the system design in detail, to warrant the validity of the assumption that the ETM could indeed place an "envelope of safety" for level 3 hazards around all other components.
5. ***Safety-Related Code Assessment.*** This confirmed a suitable approach to the use of the C language for the ETM software. It included an analysis of module complexities using McCabe, followed by supplementary unit tests on a selection of the most complex modules, extending the coverage requirement from 100% decision to 100% Boolean coverage.
6. ***System Acceptance Assessment.*** This reviewed all the testing activity performed on the ETM software, throttle control system and whole vehicle relating to the ETM, and covered the remaining MISRA clauses not covered elsewhere. It included some extensive safety validation tests,

4th Submission

5th Submission

9th Submission

performed as final confirmation of system safety performance, and an analysis of development process metrics.

The overall conclusion, taking all this into account, was that there was sufficient evidence that the S-Type/X200 electronic throttle monitor subsystem software has been engineered according to the recommendations of MISRA, to achieve safety integrity level 3. In conjunction with overall system validation (outside the scope of this project), the electronic throttle can therefore be considered, within the bounds of what is known, sufficiently protected against "unintended vehicle acceleration". The electronic throttle system was duly approved for production in the S-Type in January 1999.

4. The Hardware-in-the-Loop Simulation Testing (HILST) Project

This project was concerned with the introduction of a new technology for the development of electronic control systems generally. The author had long been of the opinion that the way Jaguar approached control system development, particularly for software-based control systems, was in need of some fundamental change. Initially this belief was due to the need for improvement. However Jaguar, along with Ford globally, is now facing a serious challenge, with a top management led drive to cut significant time and cost from its product development processes, in order to remain competitive in the market place.

The project began with a literature survey of the use of computer based simulation techniques for the development of automotive control systems. This led the author to conclude that Jaguar needed to move rapidly from a position of little or zero knowledge, to one where the tools and techniques advocated in the literature were widely used within the Jaguar control systems community. This community includes not just the Electrical/Electronic engineers, but also the mechanical engineers responsible for the hardware which is controlled. A pilot project was carried out, which resulted in a successful demonstration of the capabilities of the tools and techniques. This enabled the idea to be sold to top management, and a major investment initiative was begun, to meet the business challenge mentioned above.

4.1 Background

Electronic control systems are a major part of modern vehicles. They take the form of application specific computers, embedded in the vehicle, that are programmed to perform the functions required. These special computers are known in the business as ECUs, which is short for Electronic Control Units. The complexity and number of these ECUs is increasing all the time, and this presents a challenge for the engineers who design and develop them. The design and development of all the ECUs in a Jaguar car is performed in conjunction with external suppliers. These suppliers are responsible for the detailed design of the ECU, as well as, in most cases, for delivering a complete system, including sensors, actuators, ECU and software. Jaguar's role in control system development is mainly concerned with the specification of requirements, the validation of the suppliers output, integration of all the various systems onto the complete car, and ultimately the formal sign-off for production. It is important that Jaguar is in partnership with its suppliers, and that there is a close working relationship, which includes both Jaguar having sight of the detailed design activities at the suppliers, and the suppliers having a stake in Jaguar's role. Prior to this project, the primary method of specifying, integrating and validating control systems was a highly iterative and evolutionary approach, by building multiple phases of prototype vehicle.

As well as greater complexity, which causes its own problems for this development approach, there is also now a further drive towards greater competitiveness. Ford with its Ford 2000 globalisation strategy, which Jaguar now forms an integral part of, is re-engineering all aspects of its business. For design and development, this has resulted in a new process known as the Ford Product Development System (FPDS) which aims to set out the sequence of events necessary to achieve:

- product excellence
- customer satisfaction
- high reliability
- reduced time-to-market (down to a total of approx 3-4 years).
- total cost efficiency (in particular reduced dependence on expensive prototype vehicles)
- resource utilisation
- employee pride

FPDS is structured around a series of milestones, known as "gateways", which take place at set times leading up to production. Most are concerned with more traditional automotive engineering disciplines, but still form the framework around which all other activities must be based. The first key milestone is

"Strategic Intent", or <SI>. This requires that all the programme direction is set, and all the new technologies are identified, understood and suitably planned to minimise risk.

To deliver a vehicle programme to the often conflicting requirements of FPDS, the use of advanced design and testing computer tools must be maximised, to reduce the reliance on expensive prototype parts and vehicles, and to facilitate a "predictive" system engineering environment. One such technique is Hardware-in-the-Loop Simulation Testing (HILST). HILST is the term given to the technique for testing systems whereby only part of the system is real, and the rest is a computer simulation, so that expensive prototype vehicles are not necessary for performing all of the verification and validation tests on ECUs. Normally, to be able to fully test an ECU, it is necessary to have it installed in a vehicle, see Figure 9, because the functions it performs are heavily related to the dynamics of the mechanical systems it controls. That is, inputs from sensors reflect some real physical state of the vehicle, and outputs to actuators cause dynamic changes in other states. In addition, the large amount of real-time interaction between ECUs makes it increasing difficult to test one in isolation from the others.

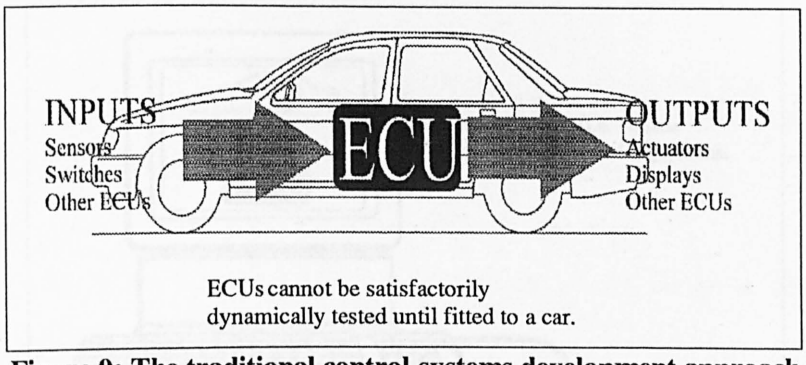

Figure 9: The traditional control systems development approach

In HILST, an ECU or ECUs are not installed into a vehicle, but rather all the sensory inputs and actuator outputs are wired via the appropriate electrical interfaces into a computer, see Figure 10. The computer is programmed to interpret actuator commands from the ECU(s) to determine the change of state that should result, and to provide realistic sensory inputs back to ECU(s) which reflect that change of state.

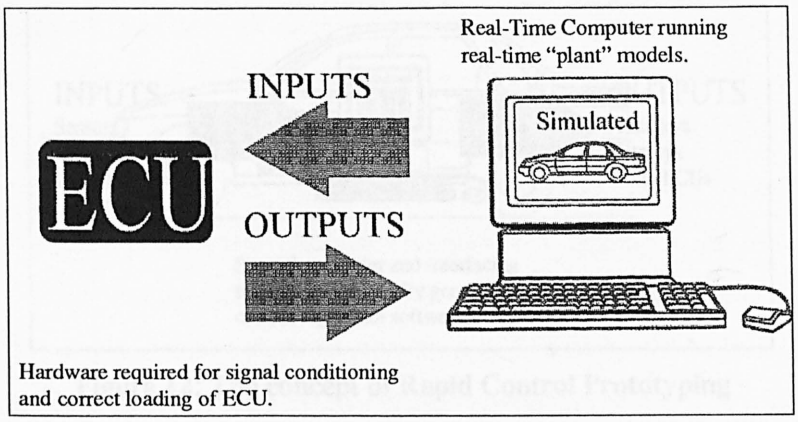

Figure 10: The concept of Hardware-in-the-Loop Simulation Testing

HILST is not a technique that can be considered in isolation, there are also some other tools and techniques which should form part of an overall computer-aided-engineering (CAE) strategy for electronic control systems. Firstly, there is "pure", or "off-line", simulation, which is a natural precursor to HILST, where both the "plant" (e.g. the vehicle), and the controller (e.g. the ECU) can be modelled and simulated, before the supplier commits to a hardware and software design, see Figure 11.

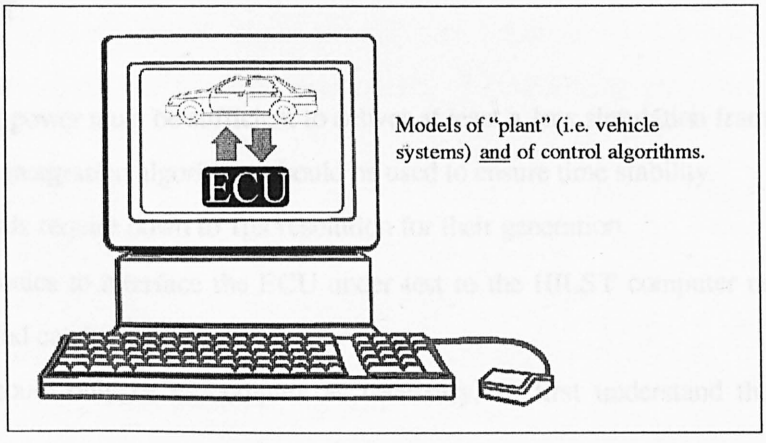

Figure 11: "Pure", or off-line, simulation

Secondly, there is rapid control prototyping, or RPC, a technique which can be used to analyse and explore control system requirements, to increase the effectiveness of capturing and specifying those requirements to suppliers. See Figure 12.

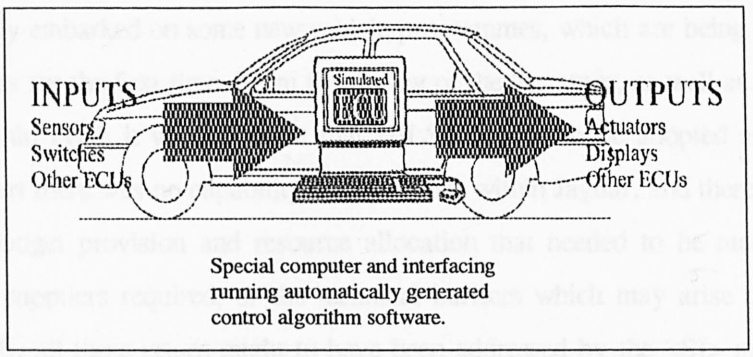

Figure 12: The concept of Rapid Control Prototyping

HILST and RPC, however, are somewhat different to "pure" simulation, in that they require special hardware platforms, and significant capital investment, rather than just being a software tool.

There are many published papers on control system simulation, HILST and RPC, which provide serious evidence that there is a significant amount of work going on across the world, not least by many of Jaguar's competitors. In reviewing a selection of these papers the following main conclusions can be drawn about HILST:

- Processing power must be sufficient to deliver at least a 1ms simulation frame time.
- Fixed step integration algorithms should be used to ensure time stability.
- Some signals require down to 1μs resolution for their generation.
- The electronics to interface the ECU under test to the HILST computer need to be specially designed and can be non-trivial.
- Models should only be as complex as necessary, so first understand the objectives of the simulation.
- dSPACE GmbH from Germany offer an excellent set of hardware and software tools suitable for HILST, which integrate very well with (one of) the "industry standard" MATLAB®/SIMULINK® modelling environment(s).
- The same (or very similar) dSPACE tools used for HILST, are also suitable for RPC.
- Graphical user interfaces for modelling and for interaction with the real-time simulation offer vastly superior, user-friendly, methods for building and running HILST projects.
- Many of the main modelling challenges for road vehicles can be based on other published work.
- Model validation may or may not be important, depending on what the model is to be used for (see Section 4.3.4).

Jaguar have recently embarked on some new vehicle programmes, which are being executed according to the FPDS process for the first time. From the review of the literature, as well as knowledge of other work going on within Ford, it was apparent that HILST needed to be adopted as soon as possible. However, at the start there was no capability or experience within Jaguar, and therefore no information on the levels of budget provision and resource allocation that needed to be made, or the types of relationships with suppliers required, or the technical barriers which may arise in adopting HILST. According to FPDS, all these issues ought to have been addressed by the <SI> milestone, which had already passed. Hence, as things stood, the risk in recommending HILST as a major step forward on the FPDS programmes which were already underway, was too high. In order to address this situation, a small pilot project was instigated quickly, in order to investigate the feasibility of HILST, and to attempt to identify and quantify the pre-requisites for its use on the new programmes.

4.2 The Pilot Project to Demonstrate the Feasibility of HILST

Given that Jaguar had, at the start of this project, zero knowledge and experience of HILST, the objectives of the pilot study were to answer some, or all, of the following questions, as soon as possible:

- How practical a proposition really is Hardware-in-the-Loop Simulation Testing?
- Can simulation and HILST ever truly replace the testing of real cars?
- If so, what are the realistic capabilities and limitations?
- Are there any additional benefits not currently known about?
- Which tools are needed, and are the tools selected suitable?
- How can existing processes be adapted to include simulation and HILST?
- How long does a typical HILST application take to set up?
- Are there any key technical issues which need to be understood and solved?
- What are the set-up costs and training needs?

A key factor in the choice of subject matter for the pilot project was that it should not be distracted by the HILST target, that is, a system should be selected which is both simple and already fully validated. This is to enable the attention to be kept on the feasibility of HILST process itself.

A suitable candidate was the driver's door control system from Jaguar's 1997 Model Year XK8 sports car, which has a small electronic module, known as the Driver's Door Module, or DDM, embedded in each door. This was a novel application of simulation tools, because this type of system has

traditionally not been thought of in control engineering terms. Most published examples are applications such as engine control, e.g. [25], and vehicle dynamics, e.g. [26]. A plan was duly constructed for a HILST project based the on the DDM, which, at the time, had already been in volume production for about a year.

From the conclusions of literature survey, it was decided to base the pilot on the MATLAB®/SIMULINK®/STATEFLOW® toolset from The Mathworks Inc., and HILST equipment from dSPACE GmbH. As time was so tight, the plan also made use of expert consultancy support resources from the UK vendors of the tools, Cambridge Control Ltd., to speed up the learning curve and to provide a direct link to dSPACE (some customisation of hardware was necessary to support the Ford proprietary SCP serial communication capability of the XK8 DDM). The technical details of the HILST pilot project for the XK8 DDM were the main topic for the rest of this EngD project.

As stated earlier, the main purpose of the pilot study was to understand the process for using simulation and HILST to assist the development of electronic control systems and decrease dependence on prototype vehicle tests. This project therefore needed to include both the capability to model the system entirely as an "off-line" or "pure" simulation, as well as the development of a real-time, "on-line" Hardware-in-the-Loop simulation. It was sensible to approach the off-line simulation first, and move onto the "on-line", HILST part later. This being the case, it was important, when constructing the off-line model, to keep in mind the underlying objective of developing it later into an on-line, HILST application. In practice, this meant not getting into too much detail with the off-line model, and only modelling the behaviour which was pertinent to the DDM, i.e. if the DDM did not have an input which could detect, or output which could influence, a behaviour then there was no need to model it.

4.2.1 The XK8 Body Control System and the DDM

The Driver's Door Module, or DDM, on the XK8 is one of 7 electronic control modules that are multiplexed together on a serial data communications bus, together forming what Jaguar calls the "Body Control System" [27]. The full Body Control System consists of a module in each door (one of which is the DDM), a module under each seat, the instrument cluster and two further modules at the front and rear of the vehicle, see Figure 13. Functions are distributed between these 7 modules, using an SCP bus to pass information between them. SCP is a multi-master multiplex protocol that transmits data at a rate of 41.7kbps, using pulse width modulation, over a twisted pair bus [28]. The partitioning of the functionality between the 7-body control system modules depends on several factors, such as the provision of suitable I/O, reduction of wiring runs, processing capacity and failure-mode management.

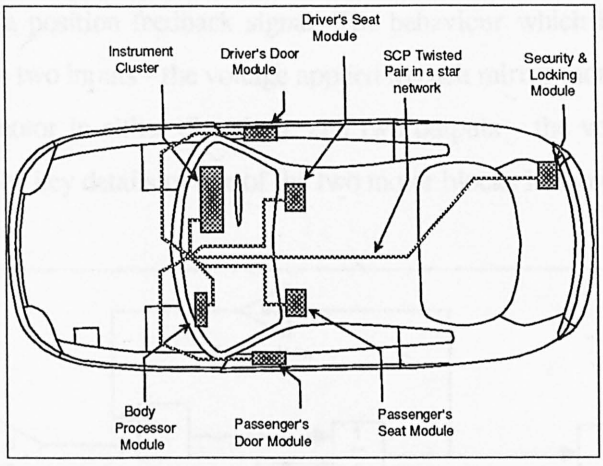

Figure 13: The XK8 Body Control System Architecture

The DDM essentially controls three main functions in the driver's door: central locking, with superlocking (or double-locking); electric window operation, with one-shot and anti-trap functions; and electric wing mirror movement, with memory positioning. It contains a small, low-cost 8-bit microcontroller, interfaced to 11 wired outputs and 22 wired inputs, and it transmits 51 SCP messages and receives 146 SCP messages (including diagnostic messages and requests) to and from the other 6 modules.

4.3 Off-line Model of the Driver's Door Control System

It seemed a sensible approach initially to treat each of the 3 sub-systems, i.e. locking, windows and mirrors, separately as 3 independent pieces of "plant" and "control", and then to integrate them together into a single door system model.

As was stated earlier, throughout the off-line phase, it was important to keep in mind that the purpose is to demonstrate on-line simulation, with an actual DDM "in-the-loop" with real-time simulations of the "plant", i.e. locks, mirrors and windows. This being the case, it was decided not to attempt to produce a full model of all the control logic of the DDM software. Instead, it was only necessary to model the DDM at a very simple level, only sufficient to be able to interact with "plant". This allowed more attention to be paid to the plant models, which were needed for testing against the real DDM in-the-loop later in the project.

4.3.1 The Mirror Subsystem

The mirror "plant" needed to capture the behaviour of an electric wing mirror assembly, the key component of which is the mirror actuator. This consists of two small permanent magnet DC motors, one for the horizontal motion of the mirror and one for the vertical motion, each of which has a

potentiometer to provide a position feedback signal. The behaviour which is important to the DDM required the model to have two inputs - the voltage applied to each mirror motor (and which can change polarity for driving the motor in either direction), and two outputs - the voltages from the feedback potentiometers. Some of the key details of one of the two motor blocks is shown in Figure 14.

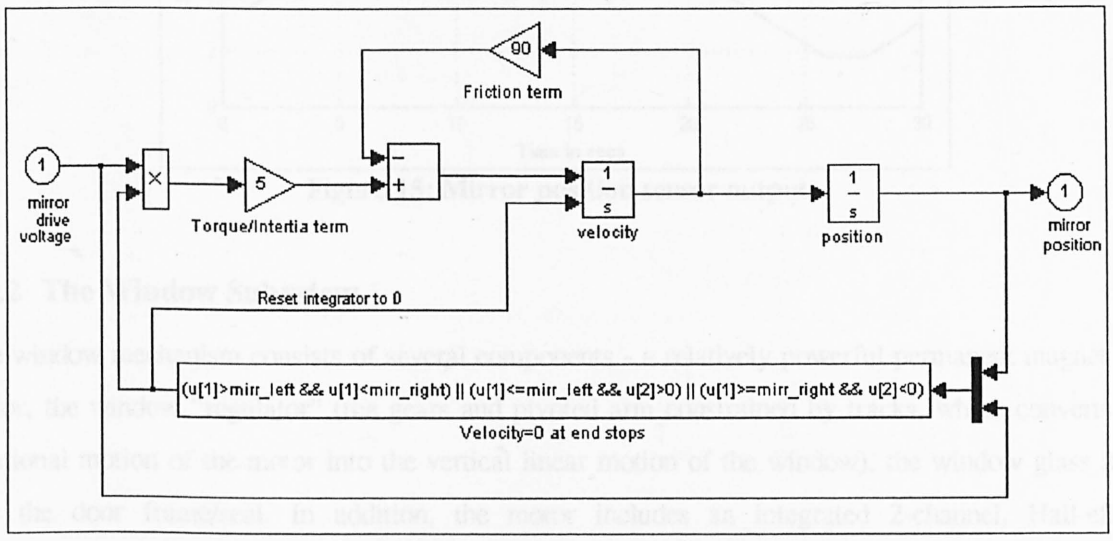

Figure 14: The mirror dynamics model

This simple model assumed that the drive voltage generates a proportional torque, which can be integrated twice to get position. Motor transients, current and back-EMF are not important functionally, therefore were not modelled. The only complexity was the need to have the velocity at zero when the mechanical limit is reached, even if a voltage is still applied, which was achieved by a reset function for the "velocity" integrator.

The "controller" model for the mirror sub-system was kept to a simple piece of logic which acts like four switches, one each for mirror up, down, left and right. The real DDM software includes much more functionality for memory positioning etc. It was not necessary to model this in order to demonstrate the behaviour of the plant.

The only important factors for HILST are the speed at which the mirrors move, as measured by the sensors, and the output voltage range over which the sensors operate. These were obtained, firstly by checking the design specifications for the system, and secondly by taking data from a real car. The parameters of the model (e.g. friction) were then adjusted manually to achieve a representative signal. Figure 15 shows the model output of one of the feedback position sensors in response to the mirror being commanded one way then the other.

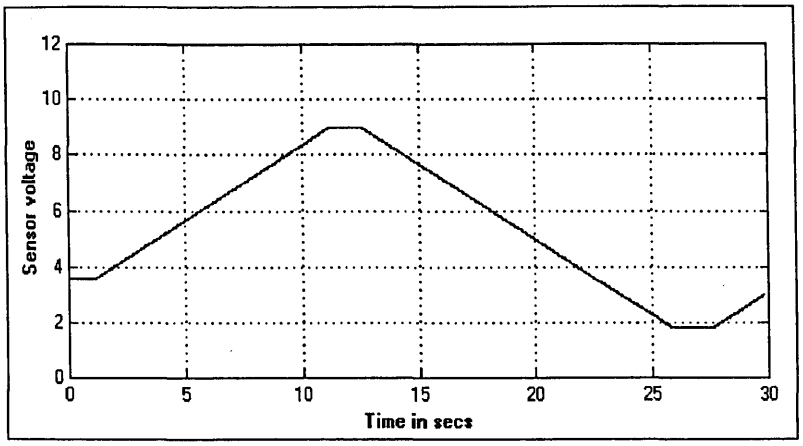

Figure 15: Mirror position sensor output

4.3.2 The Window Subystem

The window mechanism consists of several components - a relatively powerful permanent magnet DC motor, the window "regulator" (the gears and pivoted arm constrained by tracks, which converts the rotational motion of the motor into the vertical linear motion of the window), the window glass itself and the door frame/seal. In addition, the motor includes an integrated 2-channel, Hall-effect, incremental position encoder. One of the key pieces of behaviour which is important for the window system was the current in the motor, which the real DDM measures for the purpose of detecting when something is trapped in the window. This time, therefore, it was necessary to model the effects of back EMF, and to include terms for motor coil winding resistance and inductance in order to get a reasonable representation of current flow. A key difficulty was the non-linear way in which the friction on the window varies with its position. These non-linearities arise because of the shape of the window glass, which govern how much of it is in contact with the seal at the bottom, and the effects of the seal as the window reaches the top. This was modelled with a look up table which approximates the friction into four regions. The key details of the SIMULINK® model are shown in Figure 16.

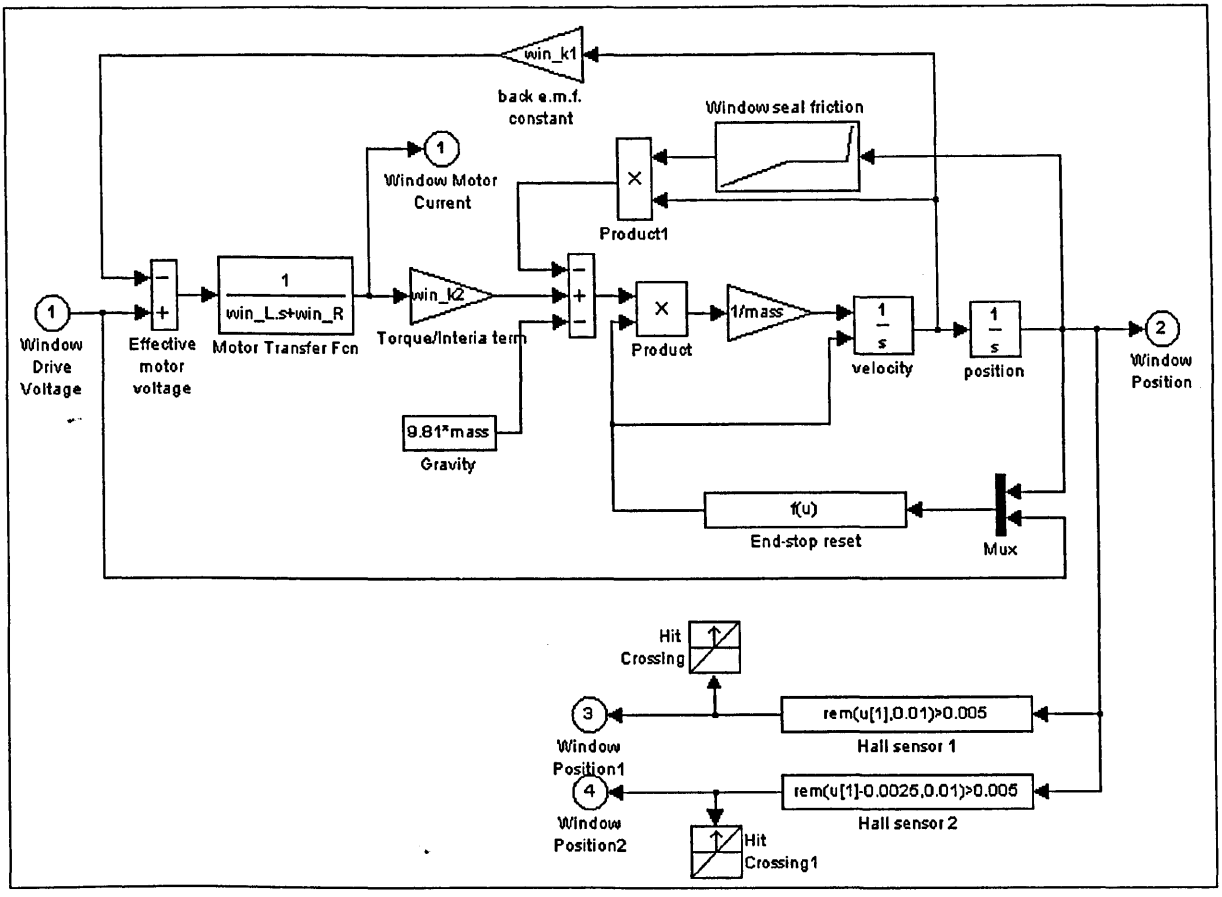

Figure 16: The window dynamics model

Again as the purpose of the controller in this model was purely to allow the plant model to be exercised, it was kept exceedingly simple. All that is necessary is the application of a voltage, which changes polarity to change direction of the motor. In the real DDM the window control is much more complex, including a current and position measurement to allow detection of anything trapped in the window. The real algorithm also has to "learn" the end stop positions of the window by counting pulses from the Hall-effect sensors. Figure 17 and Figure 18 show the key characteristics of the model, these are window position and window motor current. They show the window moving from fully closed to fully open and back again, and how the motor current varies during this cycle. Figure 19 shows the output from the Hall-sensors, for a short period in each direction. The parameters for this model were initially estimated to give approximate results, however much more representative results were obtained by using measured data from a car, and optimising the parameters automatically using non-linear identification software from Cambridge Control Ltd., which operates on the data and the model from within the MATLAB® environment.

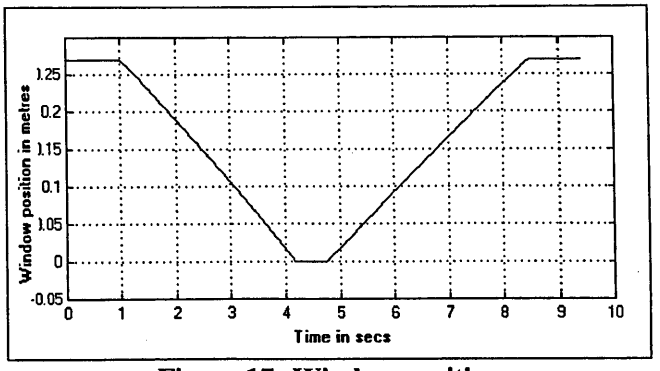

Figure 17: Window position

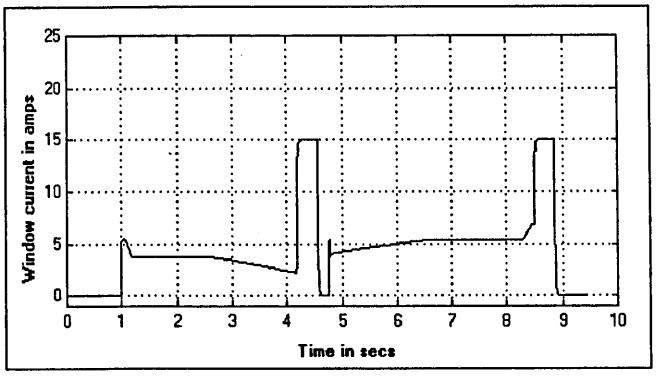

Figure 18: Window motor current

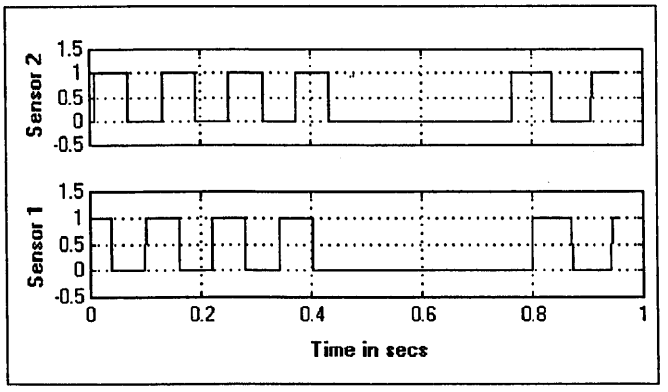

Figure 19: Window Hall-effect sensors

4.3.3 The Locking Subsystem

The main part of the plant model in this instance is the door latch assembly. This is a quite a complex piece of mechanical engineering, with many linkages, springs and levers, coupled with an electrical actuator and 6 microswitches. However, using the powerful paradigm provided by STATEFLOW®, it was possible to simplify the key behaviour down to something manageable. The basic approach was to separate the dynamics from the "state-transition" functions, see Figure 20. The dynamics were kept very simple, consisting mainly of a small permanent magnet DC motor, similar to the mirror motors described earlier, which is the core of the locking actuator. There are 3 drive signals, which in the "off" state are all grounded. The latch is locked by applying a 12v pulse for 750ms on the "lock" input. This activates the motor which moves a worm gear acting on the mechanism. When it reaches the lock

position, a microswitch switches the motor terminal from the "lock" input to the "superlock" input. To superlock the door latch, a further 750ms pulse is put on "superlock" input. Superlocking moves the worm gear further along its track, which in turn moves a small plastic cam to mechanically fix the actuator in position. (This makes it impossible to unlock the door mechanically, e.g. from the interior handle). Unlocking is achieved by a 12v pulse for 750 ms on the "unlock" input.

Apart from the movement of the lock motor, all the other behaviour is "state-transition" based, and can be modelled using a STATEFLOW® block within SIMULINK® as shown in Figure 20. This block contains a Harel statechart [29] with 3 concurrent and dependent superstates, representing the actuator, the door and the "bolt".

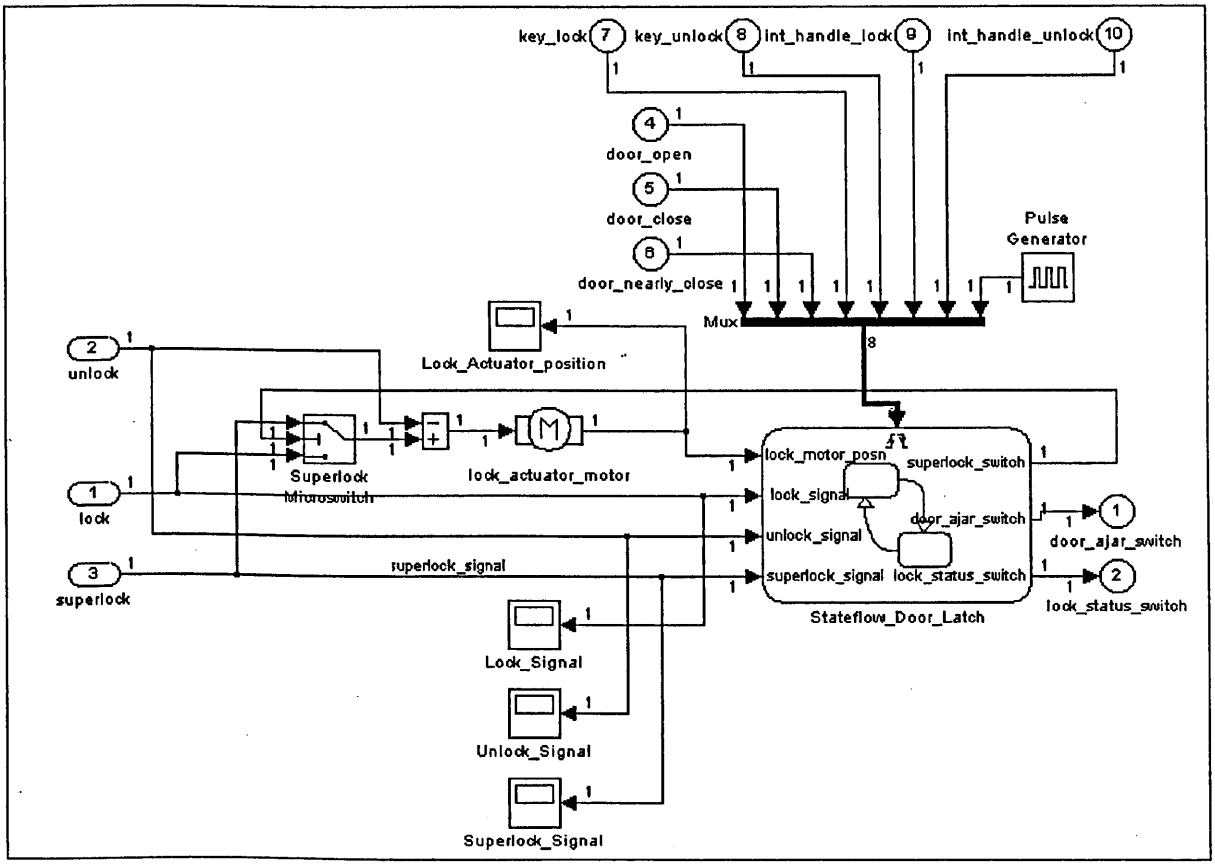

Figure 20: The locking plant model

Figure 21 shows the actuator superstate. A locking sequence goes through the states as follows:

- "free" , when the "bolt" can be locked and unlocked mechanically with the key or with the interior handle.
- "locking", when the lock actuator is moving the lock direction, i.e. a signal is present on the "lock" input.

- "locked", when the actuator has reached the lock position (lock_stop), and the superlock microswitch switches over (as described earlier). This is a sub-state of "free" as, of course, the door can be unlocked mechanically using the interior handle or the key.
- "Superlocking", when the actuator is in the process of moving to the superlock position
- (the worm gear moving the small plastic cam).
- "Superlocked". It is now impossible to unlock the door mechanically.
- "Unsuperlocking", moving the cam out of position. A signal is present on the "unlock" input and the motor is moving in the other direction.
- "Unlocking", as "unsuperlocking", except the superlock microswitch switches back over.
- "Unlocked", again a substate of "free" as it is possible to mechanically lock the door.

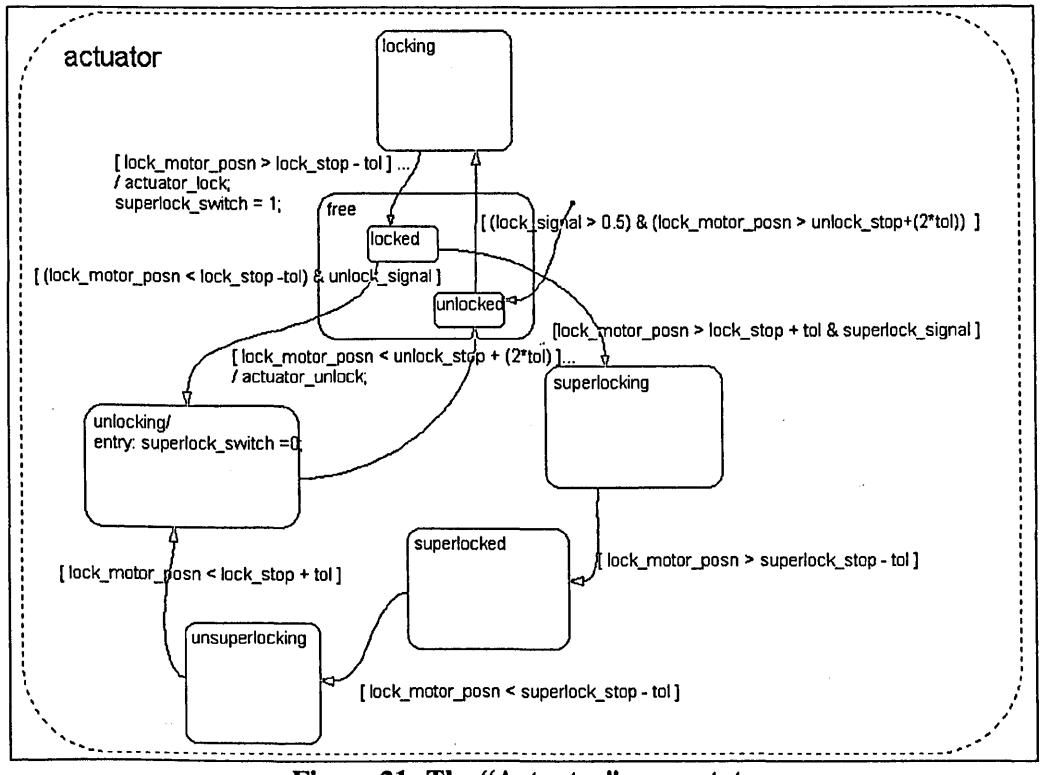

Figure 21: The "Actuator" superstate

Concurrently with the actuator, and dependent on its state, is the "bolt". This is somewhat abstract in that there is no single physical component identifiable as the bolt, but it represents combinations of mechanical levers and springs which physically lock the door. Its superstate is shown in Figure 22. It is basically either "locked" or "unlocked", and it moves between the two by the action of the electrical actuator, the interior handle, or the door key. In the real system, the door key barrel has two microswitches indicating the key in the locked and unlocked positions. These are used as inputs to the model ("key_lock" and "key_unlock"). There are two additional "error" states one which enables the door to be unlocked only via the key when the actuator is superlocked and the battery goes flat, and

another which allows for the bolt to be locked when the door is open (lock not synchronised). There is a microswitch output which indicates whether the mechanism is locked or unlocked.

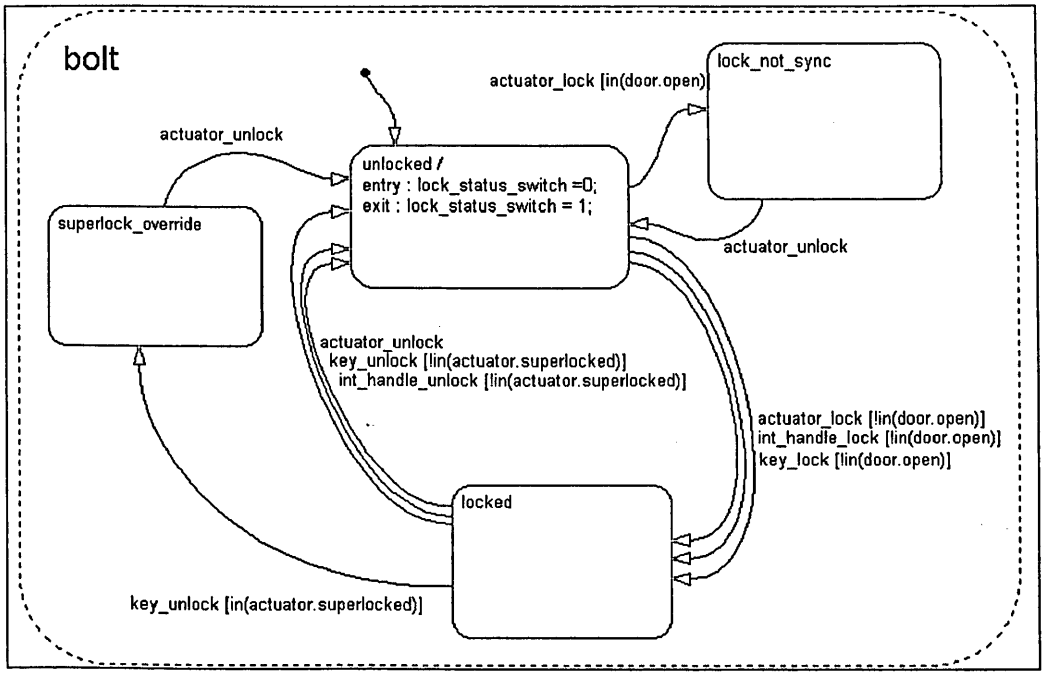

Figure 22: The "Bolt" superstate

The final superstate represents the door, as shown in Figure 23. It is either open, closed or ajar. Whether is can be opened from the closed state, depends on whether it is locked. There is a microswitch output which indicates whether the door is fully closed or open/ajar.

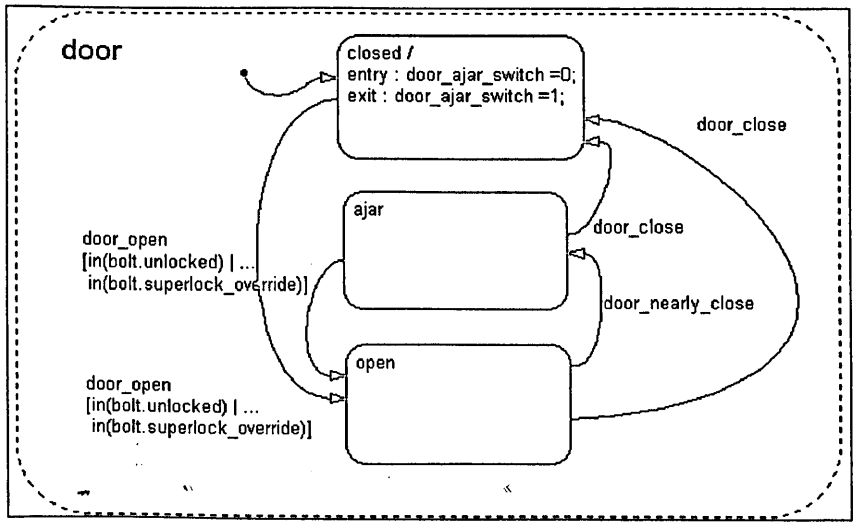

Figure 23: The "Door" superstate

There is a further microswitch in the door latch assembly which indicates when the exterior door handle is lifted, but this is not used by the locking subsystem.

In order to exercise this quite complex plant model, it was necessary to have more complexity in the controller than is required for both the windows and mirrors. Two further STATEFLOW® blocks were required to translate the key and interior handle signals into the correctly sequenced 750ms pulses, (superlocking takes place when the key is moved from unlock to lock within 3 seconds), and to account for the state of the other door (XK8 has two doors). These are not shown in the interests of brevity, but take a similar form to the plant models as described.

An illustration of the locking sequence is shown in Figure 24. It shows the actuator position as it first moves to the lock position, with 12v applied to the lock input, then as 12v is applied to the superlock input, it moves to the full, superlocked, position. A further 12v pulse on the unlock input, returns the actuator to the original unlock position. The microswitch signals, which are not shown, give the locked/unlocked status and the door closed/ajar status.

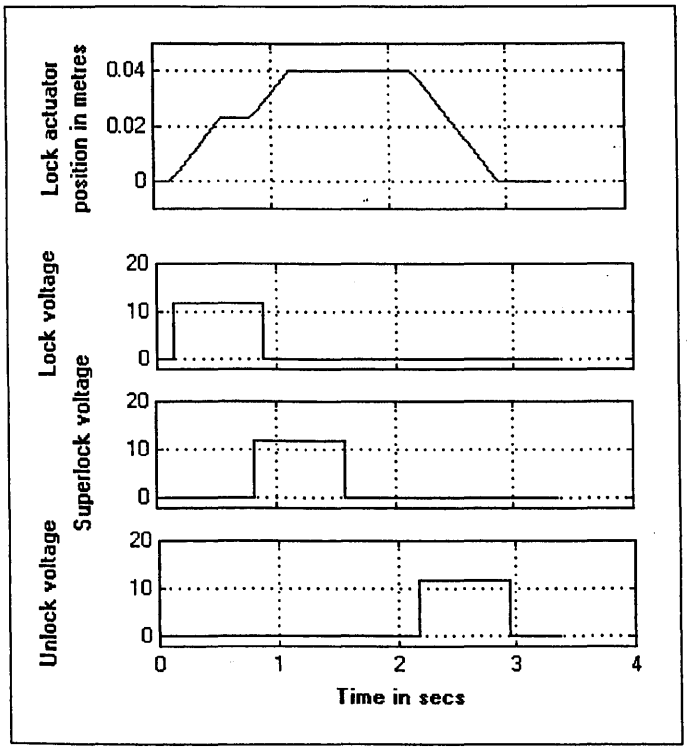

Figure 24: A locking sequence

4.3.4 Model validation

Model validation was not a major issue for the pilot study, because of the nature of the objective in this case, e.g. basic verification of software functionality in advance of testing on a real vehicle later in the development programme. Only trend-wise correlation was necessary, capturing just the major behaviours, and was relatively easy to achieve by matching data from a real car to the model using both

manual parameter adjustment and taking advantage of optimisation tools within the MATLAB environment. However, if HILST, rather than a real prototype, is to be used to make predictive decisions on the performance of a system, such as the design or tuning of a controller, then validated models with a good level of confidence would be essential. Otherwise the whole point of HILST may be negated, i.e. when the first real prototypes are built the system may be found not to perform well enough, even though it did as "hardware-in-the-loop", and problems will have been left undiscovered much later in the programme than would have been the case if traditional testing had been used. It is therefore of paramount importance to fully understand the objective of the model and the level to which HILST is to be used. This will enable an appropriate level of attention to be paid to model validation, and proper planning of activities such as data acquisition, parameter experimentation and system identification.

4.3.5 Integration of the Subsystem Models

As with many engineering problems, the approach to modelling required an ability to break the problem down into smaller constituent parts which were then addressed individually. For the example described here, this was apparent in the way the windows, mirrors and locks were initially modelled as separate subsystems. However, it was equally important that having modelled each of these individually, that they were then integrated together to form a model of the whole door control system. This raised two key issues.

Firstly, whilst the three sub-functions, windows, mirrors and locks, could be regarded as separate for the purposes of developing the plant models, to complete a model of the system as a whole it was not just a case of putting the three together. There was a significant amount of additional functionality which "falls through the cracks" when treating the main sub-functions separately. This comes from the interactions of the sub-functions with each other and, more significantly, with the rest of the vehicle. For example, the door lock control cannot be fully described by the driver's door alone, as the vehicle has central locking which must depend on the passenger's door too. Furthermore, the control for the central locking is distributed across several modules on the Body System SCP network, and therefore the DDM cannot be understood without modelling the interactions with 3 of the other 5 modules, the Passenger's Door Module (PDM), the Body Processor Module (BPM) and the Security and Locking Module (SLM). Each of these exchanges messages over the SCP network to produce the required system functionality. It is interesting that this approach seems to be more complex than necessary, but the functionality is deliberately distributed for added security, i.e. it is impossible to defeat the locking system by tampering with any one module. Modelling the behaviour of the SCP message traffic that

relates to the driver's door system proved to be tricky, not least because of the large number of connections that needed to be multiplexed and de-multiplexed within SIMULINK®. With larger mux/demux blocks it became increasing difficult to keep track of the order in which signals are vectorised, and great care was required to ensure that the ordering was correct. It is expected that this concern will be addressed in a future release of SIMULINK® by the introduction of a new bus block, from which all the vectorised signals can be referenced by name instead of sequence order.

Secondly, it was not possible simply to join three separately developed sub-function models together, at least not without breaking them down to low level blocks that are then effectively used to build a new model from the bottom up. This is a problem that would be a fundamental issue in a team where several individuals are working on different parts of the same model. If not addressed, it would result in difficulties in bringing the work together into a larger single model. One solution was to have some common, generic structure, into which all the individual models must fit, i.e. collecting the low-level blocks into sensible groups. It was then much easier to take the groups of similar blocks and combine them with mux/demuxs in a recursive manner. Figure 25 shows the structure of the model adopted here. Both the whole system and each of the three subsystems have a very similar structure. The combined model was then be put together more easily, for example by grouping the three subsystem "Test Inputs" blocks inside the single "Test Inputs" block at the top level. The result was a complete off-line model of the door control system, which contained all the features necessary to go on to the next phase, an real-time, on-line HILST application for the DDM.

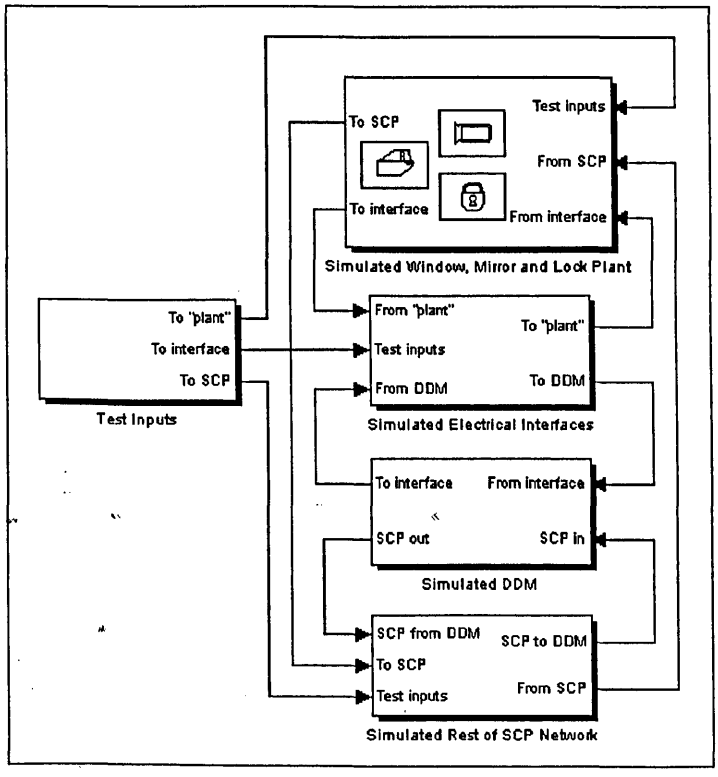

Figure 25: The overall off-line-model structure

4.4 The Real-time HILST Implementation of the DDM

Following the successful completion of the off-line model, the first step in moving towards a real-time simulation was the selection of a simple first-order, fixed-step integration algorithm (Euler), instead of the SIMULINK® default variable step algorithm used until now. A one millisecond fixed-step size was selected. At the first attempt, it was found that some parts of the model became numerically unstable, the offending portion being the window motor dynamics. This problem was corrected by adjusting the window motor parameters, and, when adjusted further to match measured data, the instability did not occur again. Hence, it was concluded that except for a slight loss of integration accuracy, the model was suitable for the real-time environment, where a fixed integration step is necessary.

One of the key steps in modifying the model itself towards a real time HILST implementation was the definition of the signals which must be passed to and from the real DDM via dSPACE I/O boards. For example, a pair of signals, such as those in Figure 15, for the vertical and horizontal mirror position feedback sensors, must be generated via a dSPACE Digital-to-Analogue Converter (DAC) board. This was a relatively simple task in principle, although one which was complicated by the need to fully understand the voltages, currents and loads associated with each pin on the DDM. For example, whether a switch input is active high or active low, to 12 volts or 5 volts.

It was at this point that the problem of how to interface the dSPACE system became prevalent, i.e. interfacing dSPACE's TTL digital and low current analogue signal levels, to an automotive environment requiring high currents and 12 volts. The solution was to construct an interface and signal-conditioning unit that can provide the real-world signals to the DDM and can buffer the signals to the dSPACE system. This interface unit was further specified to include so-called controllable loads. These allowed an analogue voltage from the dSPACE system to control the current draw from the DDM, up to a maximum of 20 amps, in order to follow the profile of the window motor current shown in Figure 18, and such that the DDM believed that it was controlling a real window system. This was an important feature because the DDM measures actual window motor current to achieve the anti-trap function. In all, the interface unit built for this application consisted of two such controllable loads, plus: 4 buffered digital inputs and simple loads (from the DDM); 19 buffered digital outputs (to the DDM); 2 analogue inputs with gains of 0.625 (16 volts at the DDM gives 10 volts to dSPACE); and 3 analogue outputs with gains of a 1.6 (10 volts from dSPACE gives 16 volts at the DDM). The interface unit was designed and manufactured to the authors requirements by a small local electronics company.

In addition to the electrical signals, the real time program must also have the ability to send and receive SCP messages between the simulation and the real DDM. No off-the-shelf solution was available from dSPACE (as SCP is proprietary to Ford), therefore it was necessary to have special hardware and software developed for this project. This became a small project in its own right, and one that puts the flexibility of the dSPACE platform and Mathworks software to the test. The solution was arrived at using the dSPACE prototyping board and the target language compiler facility within SIMULINK®. This provided a very neat and easy to use interface in which all the messages to be received and transmitted are defined within a standard spreadsheet. This is important to ensure that other Jaguar engineers, who may not have much software expertise, are able to work with the system in future applications. It is also worth mentioning that making the hardware-in-the-loop system capable of replacing all the SCP messages to and from the other body system modules, places a restraint on the configuration of the SCP network itself. Within SCP, messages are "stamped" with the physical address of the transmitting module. As there are 7 body modules on XK8, and only one SCP board in the dSPACE system (with only one physical address), it is important that the receiving module does not use the physical address. It is fortunate that in the implementation of SCP used on XK8, Jaguar has only specified so-called "broadcast mode" messages that do not use the physical address. If HILST is to be widely adopted, this will need to be a design constraint on the SCP network for future applications.

A further complication arose with the message "request" facility that is also a part of SCP. However, this problem was solved relatively easily within SIMULINK®. Figure 26 shows the SIMULINK® model for the "inertia switch" signal which needs to be transmitted as an SCP message on the transition from off-to-on or on-to-off, and when requested to do so upon receipt of a request message. Two SCP messages are defined, one for "active" and one for "inactive". A transmission of the appropriate one is triggered by a rising edge of the corresponding SIMULINK® signal.

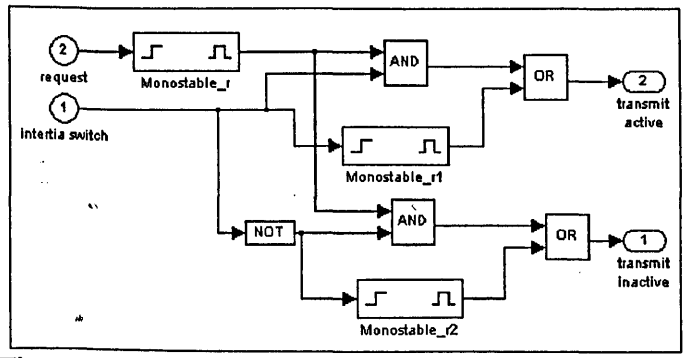

Figure 26: SCP transmission for the inertia switch signal

Figure 27 shows the overall top-level of the model once it had been adapted for real-time HILST application, together with some detail of the blocks required for the dSPACE and SCP interfacing. This can be compared with Figure 25. The "Test inputs" and "Simulated Window-Mirror-Lock" blocks are essentially the same as in the off-line model, whilst the "Simulated DDM" block has gone altogether now the real DDM is in the loop.

The final step necessary to complete the hardware-in-the-loop demonstration was to consider mechanisms for enabling the user to interact with the real-time program. dSPACE provide a software tool, called Cockpit®, for exactly this purpose. Cockpit® allows the user to tap directly into the parameters and signals of the real-time programme from an off-line PC, without interrupting the simulation. Therefore, each of the switches and input events needs to be represented. Furthermore, it was also possible to represent outputs from the model, such as window and mirror position, as bar displays within the same window. Figure 28 shows a screen shot of Cockpit® for the DDM HILST implementation.

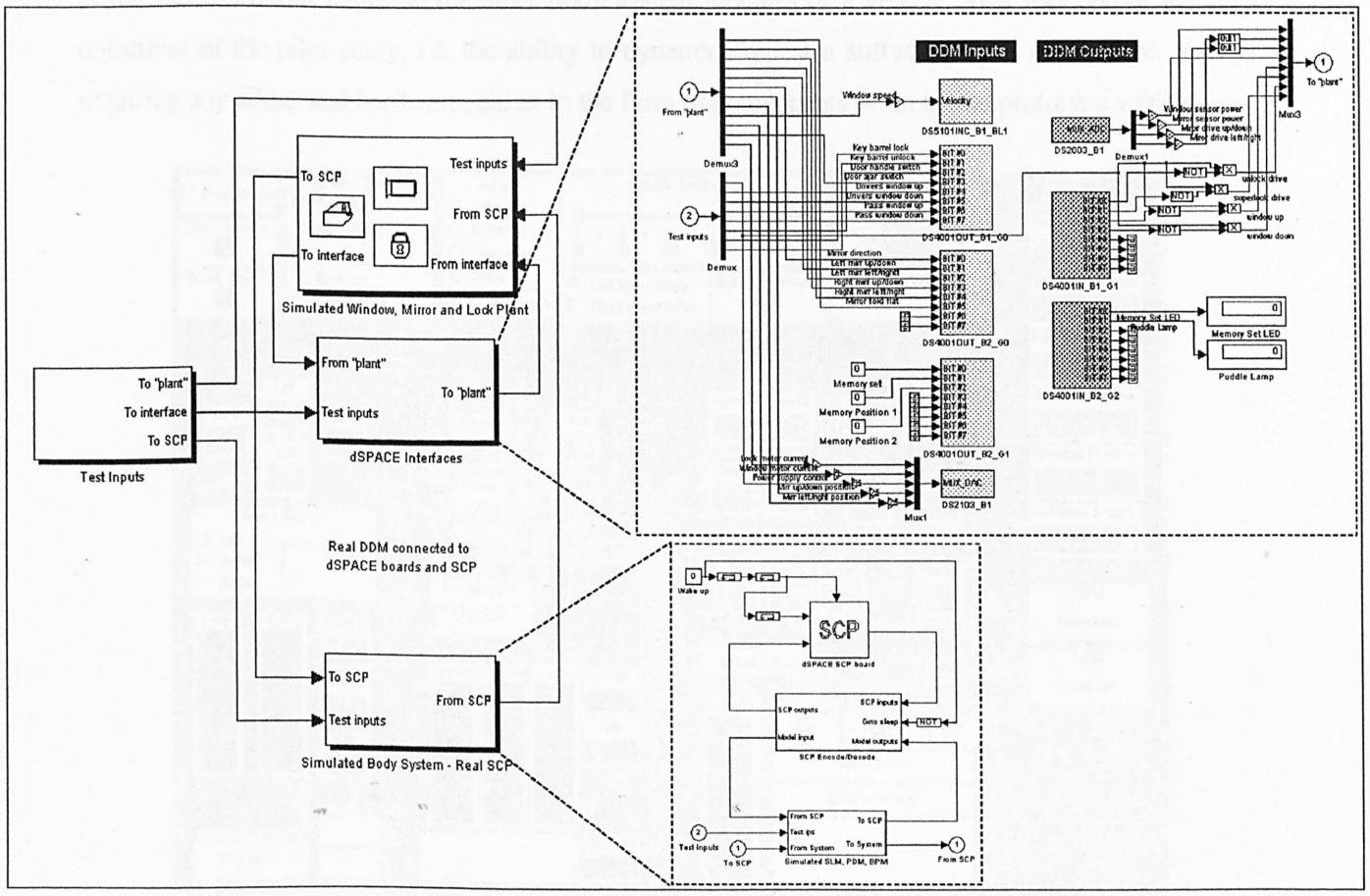

Figure 27: The real-time model structure and the dSPACE I/O interface

4.5 Results of the Pilot Project

Once the model and its interfaces were developed within SIMULINK®, it was a straightforward task to build the real-time code and download it to the dSPACE system, as the whole process is handled automatically by a "make" file. With the real time program running on the dSPACE hardware (a 1ms simulation step size on a TMS320C40 DSP processor board executes in around 350μs), connected to the real DDM via the interface unit mentioned earlier, it was possible to exercise the functionality of the DDM. For example, using the mouse to click on the driver's window down button in Cockpit®, the DDM will activate the window motor drive signal, which would in turn be responded to by the model and the simulated window moves down. Note how the controllable load within the interface unit profiles the window motor current as shown in Figure 29 (compare to the off-line result in Figure 18). This data was taken using another dSPACE software tool designed for data acquisition from the real-time simulation, called Trace®. In addition, the mirror sensor signals and the locking actuator responses are presented in Figure 30 and Figure 31, and these can be compared with the off-line results in Figure 15 and Figure 24 respectively. Hence, it can be seen that this gives the ability to fully test the functionality of the DDM without requiring the other body system modules or a vehicle. This was one of the main objectives of the pilot study, i.e. the ability to dynamically test a software-based control unit without requiring any additional hardware, either in the form of components or an actual prototype vehicle.

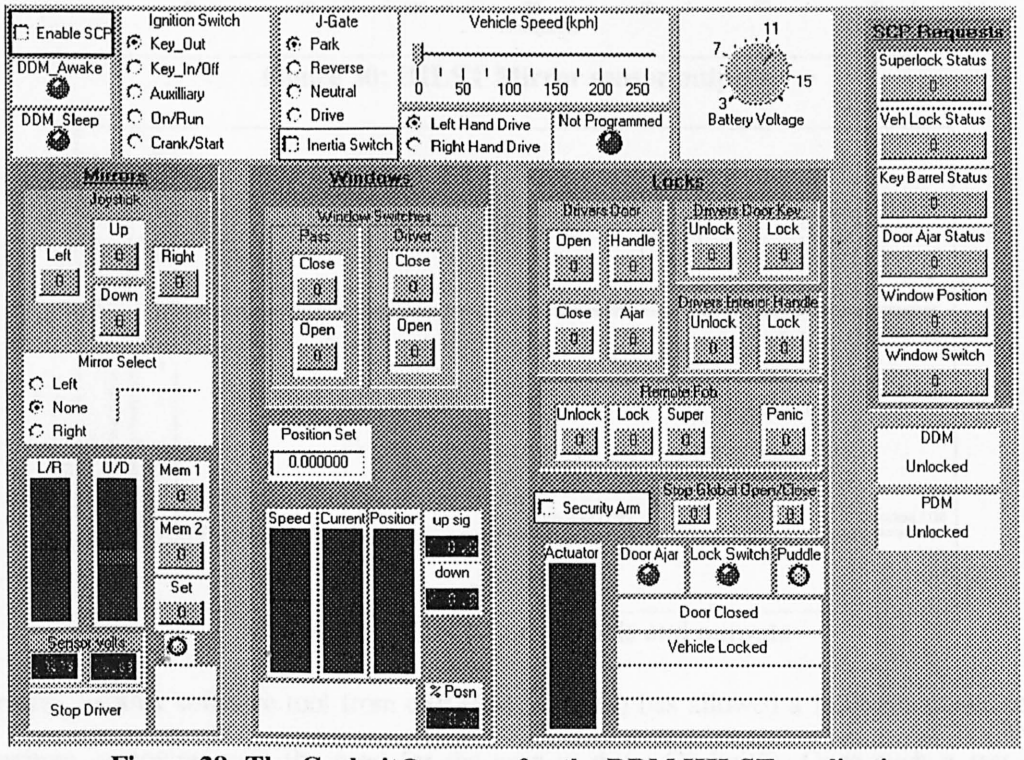

Figure 28: The Cockpit® screen for the DDM HILST application

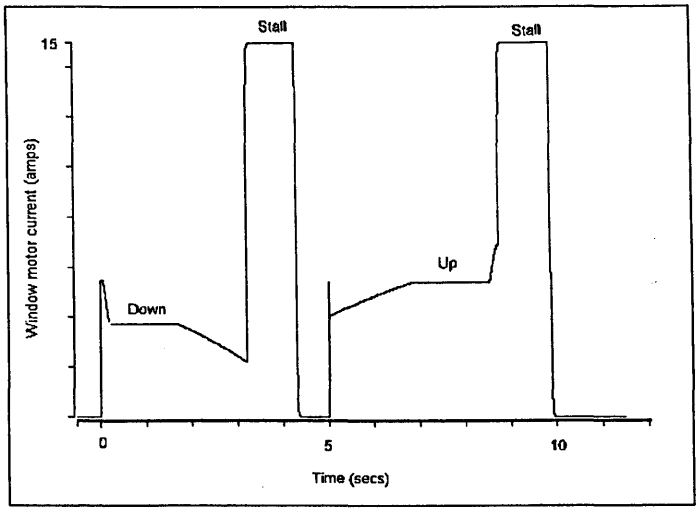

Figure 29: HILST window motor current

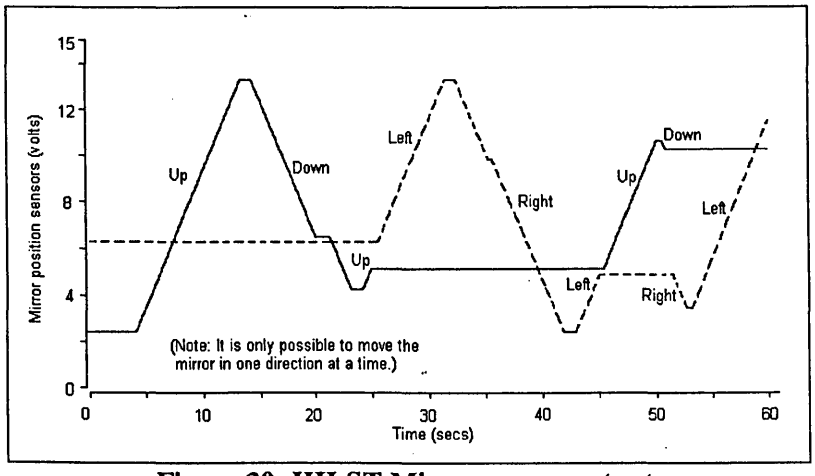

Figure 30: HILST Mirror sensor outputs

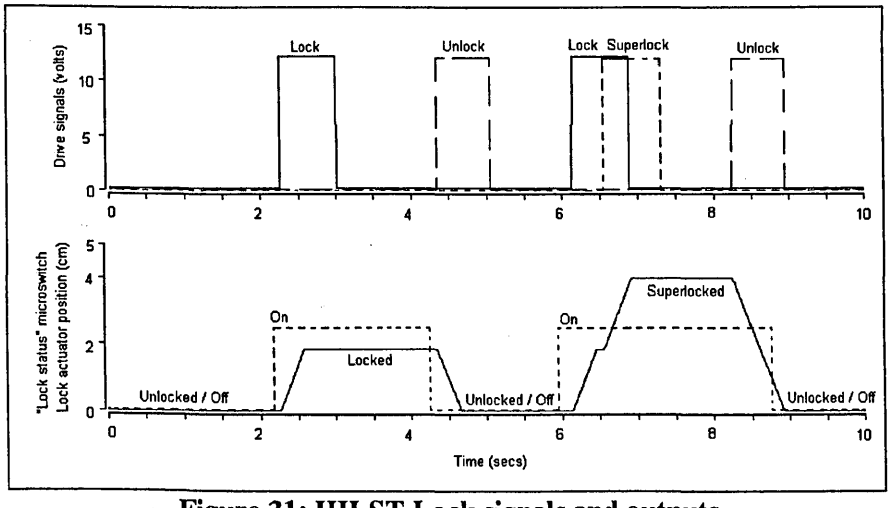

Figure 31: HILST Lock signals and outputs

Furthermore, another software tool from dSPACE (MLib®) has allowed a fully automated test script to be written, removing even the need for the user to drive Cockpit®. Once such a test script is constructed, it is possible to repeat tests and report results automatically, thus saving time and allowing more consistent and rigorous testing throughout. For the HILST pilot study using the DDM described

here, this was successfully demonstrated by coding up a small portion of the actual test plan for the XK8 relating to the door functions. It would be possible, from this point, to go on and code up the whole of the vehicle test plan following the same principles.

Once this automated test script was shown to work, the whole of the DDM HILST application was successfully demonstrated to senior management and other Jaguar engineers with an interest in the approach.

4.6 The Management of the Exploitation of HILST

The pilot project was successful, and it provided a large proportion of the information and experience necessary to answer the questions posed at the beginning:

How practical a proposition really is Hardware-in-the-Loop Simulation Testing? The driver's door system was a real application, from one of Jaguar's current products. It was possible to show that it could be successfully modelled, both as a pure, off-line simulation, and by putting the real controller in the loop. There is reason to believe, therefore, that a similar process is possible for many of the other control modules on the car. However, it is reasonable to assume that each new application will have its own set of special technical problems to solve in order to make HILST work.

Can simulation and HILST ever truly replace the testing of real cars, and if so, what are the realistic capabilities and limitations? It has been demonstrated that it was possible to exercise the DDM functionally as it would be in a real car. However, in constructing the models, the author became aware of the fact that a model can only capture behaviour that is know about. If there was a vehicle effect that was due to some previously unknown interaction, such as a component wear problem causing a change in sensor or actuator performance, then this could only be tested for with the benefit of hindsight. It is therefore the author's belief that simulation and HILST does not replace real vehicle testing completely – there will always be something new to learn once the system is in its true environment. However, as experience is gained, the problems and issues found on the real car can be fed back to the simulation process, and if necessary the models can be refined to capture the "new" knowledge of a problem area for next time. Hence, rather than saying simulation and HILST can replace the testing of real cars, perhaps it is more true to say it provides a mechanism for capturing experience gained on real cars, which can be used to ensure that such problems can be avoided in the future. Figure 32 shows a graphical representation of how the author visualises the use of simulation and HILST versus

traditional prototype testing. Basically, it must be accepted there needs to be several iterations before it would be possible to replace a significant large proportion of prototype vehicles. Until then, both methods must be run in parallel, and the results from the real testing fed into the modelling and simulation process to transfer that experience across for next time.

Are there any additional benefits not currently known about? During the rounds of demonstration of the pilot project to senior management and others, one significant new potential benefit came to light. That was the possibility of using HILST not just to verify and validate the ECUs, but also to verify test equipment used on the production line and in the dealer network. There could be very significant benefit here, as until now, it has only been possible to begin to test and debug the test equipment software once the first cars are rolling down the production line, by which time problems can cause severe disruption to the programme. It was also interesting that, once people saw the capabilities of HILST on the DDM, they would often begin to relate it to problems they had experienced in the past on other systems, and start to see additional benefits for themselves.

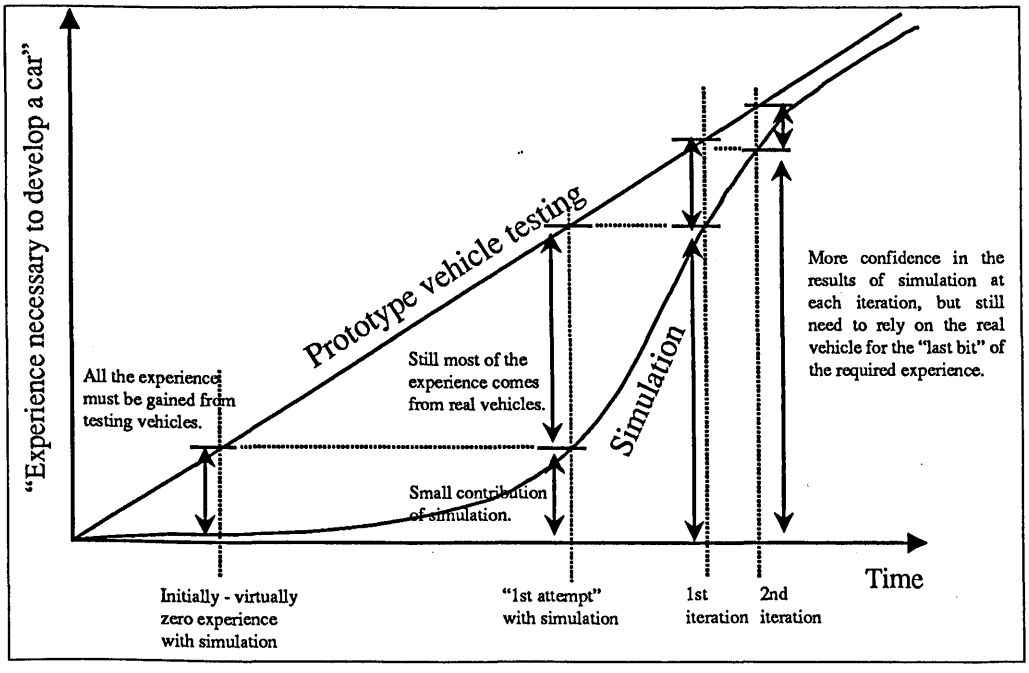

Figure 32: Simulation v Testing Experience

Which tools are needed, and are the tools selected suitable? The tools selected were the MATLAB®/SIMULINK® and STATEFLOW® tools for modelling, and the dSPACE platform for the Hardware in the Loop part. No significant problems arose with the tools, and it was concluded that there would no reason not to continue to use them on a wider basis.

How can existing processes be adapted to include simulation and HILST? As described earlier, there are 3 distinct processes associated with this type of work – pure, off-line, simulation; rapid control prototyping (RPC); and hardware-in-the loop simulation testing (HILST); all of which use similar tools but for different reasons. In an ideal process, pure simulation should come first. It is an excellent way of gaining a detailed understanding of a system and how it can be controlled, and it lends itself well to experimentation. Off-line modelling should be a pre-requisite for HILST, but this does not have to be the case. RPC is used to explore control system requirements by allowing the rapid prototyping of different control strategies with the real "plant". Hence the ideal work flow would be pure simulation first – to investigate the system concept; RPC next – to develop and specify the control algorithm; and HILST last – to verify that the controller implementation matches the specification. However, for any given project, it may be the case that it is not necessary to follow this process, but rather to select the most appropriate technique for a given point in time. For example, there is no point in spending time on a complete off-line model if the controller has already been implemented – as in the pilot study. Also, in general, the work flow described will not be sequential, and the three phases will overlap, e.g. it may be appropriate to feed experience gained during RPC into the development of the off-line model. It is important not to lay down rigid process requirements, but to retain flexibility to select the best approach for each project given its circumstances.

How long does a typical HILST application take to set up? The pilot project did take slightly longer than it was originally planned. However, this was mainly due to delays in ordering equipment and some technical issues which needed to be solved (see next paragraph). Whilst it has to be assumed that every HILST project will be different, the pilot project did confirm that executing a HILST project is still quicker and easier than using a prototype car to achieve the same ends. Hence, whilst it is not really sensible to state how long a HILST project will take, the pilot project did give some confidence that the task is compatible with the timing faced by the new vehicle programmes in Jaguar. HILST is a realistic proposition in terms of the time available, provided that sufficient effort is put in early on, see Figure 32.

Are there any key technical issues which need to be understood and solved? There were two major technical issues which the pilot project drew attention to. Firstly, the problem of SCP, Ford's Standard Corporate Protocol [28]. SCP is a communication protocol used by Jaguar for mulitplexing signals for the "body system". The body system consists of all the secondary control systems in the car, and represents a significant proportion of the electronics. Because SCP is proprietary to Ford, the dSPACE HILST platform did not support it. Therefore in order to make HILST accessible to Jaguar generally, SCP capability for dSPACE was important. A solution was developed as part of the pilot project,

which is now also being used within Ford itself. The second major technical problem was that of interfacing. A dedicated interface box was development for the pilot project, but the experience has proved invaluable for the evolution of a new, generic interfacing system which can be configured for most of the types of input and output which will be required. It would not have been acceptable, both from a cost and time point of view, to have to procure a dedicated interface box for every system Jaguar wished to apply HILST to. This generic interfacing system has been designed and built by the same local company that produced the application specific interfacing electronics for the pilot study. The generic system provides the capability for configuring the interface electronics via software, offering a very flexible and unique solution that could not be purchased prior to this project. However, the company concerned, after they have developed the system for Jaguar, will offer it on the open market as new product available to any company engaged in HILST.

What are the set-up costs and training needs? HILST and simulation facilities are expensive. However, the pilot project provided the author with some insight which enabled a strategy to be put together to set up a large scale (7-figure) investment programme in HILST at Jaguar. This strategy was based on several principles. Firstly, the principle that there would be a key set of engineers in the organisation who would need to be responsible for simulation and HILST applications on their respective systems. It would not be desirable to have all the activity concentrated in one team, as this would be likely to result in little change in the way most of the work was done, but rather to develop the techniques widely across the engineering community. Hence a set of key people were identified in the teams responsible for the different areas of electronic control systems in the car. A training course was developed which would meet the needs of these engineers, such that they would be given an opportunity to refresh their knowledge of control engineering, learn about the tools and their capabilities and do some practical experimentation with real hardware "in-the-loop". Around 40 engineers received the training initially. The second principle was the need to be able to develop off-line simulation within the office environment. This led to plans for a significant number of MATLAB® licences to be made available on the network across the Electrical/Electronic Engineering department. Next, the HILST and RPC facilities. A new, purpose built laboratory has been constructed at Jaguar for hardware-in-the-loop simulation testing, and an investment plan has been established to equip the lab with a number of HILST and RPC "rigs", each with its own workstation, dSPACE hardware and generic interfacing system. Finally, a small team has been established by the author to initiate and lead HILST and simulation projects, alongside the engineers who are primarily responsible for the various electronic control systems. The objective of this team are to provide support and expertise, which will ultimately result in these system engineers taking on the methods as a normal part of their work.

To conclude, there can be no doubt that HILST is here to stay. The Engineering Doctorate provided the "springboard" which to lead the project from nothing, to a multi-million dollar investment, which is seen as strategically important to Jaguar's future. All the elements are now in place to make HILST part of the mainstream development process. A reasonable capital budget, a high-profile new laboratory, a team to develop the HILST applications and plans to recruit to grow the team further, a training course which has given 40 key individuals an insight into simulation tools and HILST, on-going, and most importantly , the full backing of senior management. There is no reason why the team cannot now go on and deliver all the promise of HILST as a new engineering tool, and enable electronic control systems for Jaguar to be designed, faster, cheaper and better than ever was possible before.

5. The Future for Automotive Electronic Control Systems

This report started by setting the scene for automotive electronic control systems, and explained how the special needs of software presented some interesting challenges for the industry. Two of those challenges have been tackled during the course of the author's Engineering Doctorate – the challenge of using software in safety-related control systems; and the challenge of how to adapt the development process for software-based control systems, to take advantage of advanced simulation methods.

These two distinct but interrelated projects required different approaches. The first was concerned with using the industry's own guidelines on safety-related software, to ensure that a specific application achieved a sufficient level of safety integrity, and to provide a significant amount of documentary evidence. The outcome was the first time that such a project has been shown to comply with the MISRA Guidelines in such detail, and resulted in S-Type having a full-authority electronic throttle, a first for Jaguar and Ford. The second project was concerned with a new process to enable all software-based control systems to be developed more efficiently, and to try out these new processes on a pilot project. The success of the pilot study enabled senior management to be convinced of the benefits of the technology, and resulted in a 7-figure investment programme. It was mentioned in the introduction how it was the author's experience with testing the first generation of electronic throttle which led to an interest in hardware-in-the-loop simulation. Although this technology was not available to assist with the S-Type electronic throttle monitor project, with a significant testing activity which still had to take place on the test track, it is hoped that everything will be in place to enable the next electronic throttle application to be extensively validated in the HILST lab.

At the end of this Engineering Doctorate, the S-Type electronic throttle, with its safety monitoring subsystem, had been successfully approved for production, and Jaguar was just beginning a new era in its approach to control systems development. In other words, one project reached a clear end point, whilst the other provided the starting point for future work, and both have provided ample opportunities for innovation in product and process. However, to conclude this report, it is worth taking some time to speculate about the future in the context of electronic control systems, the technological developments that are on the horizon and the business environment in which the industry must operate.

There is little doubting the fact that the automotive industry is one of the most global and competitive industries in existence. Its survival depends on an ability to change and adapt quickly to the market place. For example, instead of regarding itself as an engineering-led industry, it must rather begin to see itself as a consumer-led industry. Why should the automotive industry regard itself as different to the white-goods, or even the telecoms industries? A consumer is more than just a customer who buys a car, but is also interested in a good quality service, and the longer term costs of ownership, not just the purchase price. Furthermore, he or she expects to be able to find a product which meets their needs, and not the other way round. The successful car companies therefore, will be the ones which can offer a wide range of products, which offer value, style and modern technology, and which can react quickly to the fickle nature of the consumers' whims. Jac Nasser, the newly elected president of Ford, was recently quoted as saying, "If every single person in the Ford team really had the emotions and the thinking process of the customer, and felt and acted as an owner and shareholder, then you have the essence of a truly leading global automotive company." [30].

Such thinking is leading to a restructuring of the whole industry, both from a process viewpoint and an organisational viewpoint. Product development is key to Nasser's vision, to be able to create the products which can not just meet a need, but also generate emotions. This requires a nimbleness which has not so far been the norm. Product Development functions must be responsive – to be able to hit the market at the right time; technologically advanced and efficient – to be able to offer more and more features whilst reducing costs; and competent – to ensure that products do what the customers expect of them for as long as required, i.e. exhibit high quality. To achieve such results is no longer the domain of localised all-round engineers working on individual components, but is now dependent on highly specialised teams of vehicle manufacturer engineers and global suppliers, working on complete systems. The supply chain is increasingly tiered, whereby vehicle manufacturers only deal with a handful of large systems suppliers, which in turn deal with any second-tier suppliers, and so on. These first tier suppliers are being further defined as full-service suppliers, able to work seamlessly with the vehicle manufacturer on all aspects of specification, design, development, manufacture, supply and service for

their system. Indeed, during the course of this Engineering Doctorate, Ford's own automotive components division was restructured to create "Visteon", a separate operating company able to provide full-service supply for whole systems. From a vehicle manufacturers viewpoint, the product development task becomes one of working with the full-service supplier to define requirements for systems, to integrate those systems together in to a complete vehicle, and to validate that they will meet customer needs and quality expectations.

Finally, another factor in the automotive business environment is the issue of over-capacity and an unsustainable competitive position, leading to rationalisation. This is mainly taking place through a series of major mergers and acquisitions. At the time of writing, Ford had just announced the take-over of Volvo. When such take-overs or mergers occur, management of course look for efficiencies. For product development, this is most visible through the practice of "platform sharing", whereby several different product lines can be derived from a common platform, such that many of the non-customer visible systems and components can be shared between a number of vehicle programmes. This allows the development costs of these common items to be spread over a larger production volume, reducing overall cost. However, in practice such arrangements can increase the complexity of an individual programme, and force compromise in areas where differing needs persist.

So against this background, where is automotive electronic control going? Software based control systems will continue to grow in significance, as a means of pandering to the consumers' whims. This growth can be seen in Figure 33, which shows the past and predicted dollar value of semiconductors in the average car, with the total market expected to be worth $60 billion by 2001[31].

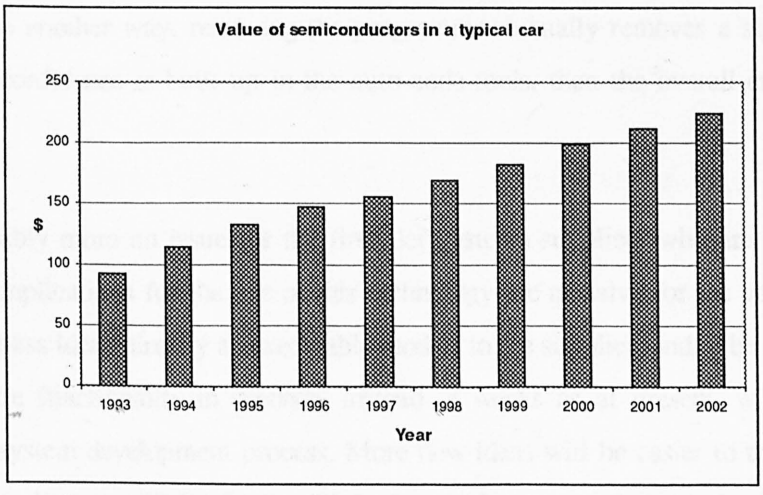

Figure 33: The value of semiconductors in a typical car

If the growth of electronic control systems is assured, to offer more high-tech features and take more control away from the driver, then this will also assure the future growth of the problems associated with software, such as those described in the introduction to this report. However, there are a number of technological advances which are targeted at ensuring that problems areas do not grow accordingly.

In this report, the subject of simulation and hardware-in-the-loop simulation testing has been explored. Simulation and HILST as described will help to develop control systems, however this is just the beginning. The goal for many of the tool developers, Mathworks and dSPACE included, is to reach a situation whereby no software needs to be written. Instead it is automatically generated directly from the simulation, so-called "auto-code". Whilst it is true that auto-code has been around for some time, indeed HILST and RPC depend on it, the code produced is unsuitable for use in a real control system implementation. There are two main reasons for this. Firstly, it is inefficient. A competent programmer has always been able to write more efficient code, as measured by ROM size and execution time, from a given specification than the auto-code tools were able to generate. Whilst code size and execution time are not big issues for HILST and RPC, which can use as much memory and as powerful a processor as required, they are for production Electronic Control Units, which seek to minimise cost in all aspects. Secondly, there have always been doubts about the robustness of auto-code. How can the auto-code tools be trusted to generate code which is correct? This is further compounded by the fact that code generated is often unreadable by a human, and it is not easy to inspect the code manually to verify its correctness. Also, what about safety concerns? However, the auto-code tools are improving, and, at least as far as efficiency goes, claims are now made that the tools can actually create software that is more efficient than a human programmer could [32]. As for robustness, the tool developers are working towards code that is readable, well structured and easy to inspect and test. Furthermore, looking at the issue another way, removing the programmer actually removes a significant source of error. Once some confidence is built up in the auto-code tools, then the overall error rate should be much lower.

Auto-code is probably more an issue for the first tier systems suppliers who are responsible for the software, but the implications for the use of this technology are massive for the vehicle manufacturer too. To be able to pass ideas directly as executable models to the supplier, and to be able to turn around changes to software functionality in seconds, instead of weeks as at present, will revolutionise the electronic control system development process. More new ideas will be easier to try out, development time will be cut and costs will be lower. With the current state of the technology, it is perhaps conservative to predict that the pure software programmer in the automotive industry will be redundant

within 5 years. Programmers would be well advised to start re-training in the systems simulation and modelling skills which drive the auto-code tools.

The main focus on the auto-code tools is the C language, and they are able to generate C code for the functionality captured in a graphical model, whilst it is still necessary to write many of the "low-level" software modules manually. This is the software which accesses the hardware resources of the microcontroller ICs, mainly concerned with input and output, such as counter/timers, analogue to digital converters, bit I/O, serial communications channels etc. However, this area is the subject of recent developments in real-time operating systems. These operating systems provide a standard interface between the application software, such as that which can be generated by auto-coders, and the I/O resources of a microcontroller. Recent progress in the standardisation of real-time operating systems suitable for automotive applications (i.e. they also must be efficient in terms of code size and execution overhead) is significant [33]. The new S-Class Mercedes, launched in 1998, is reputed to have several Electronic Control Units running a version of the OSEK real-time operating system. Hence, operating systems, coupled with auto-code, are a truly mouth-watering prospect for those interested in responsive and efficient new product development.

Other developments include the use of languages other than C. For example, there is a lot of interest in Embedded C++ [34], which offers the potential for object orientated design, currently one of the most fashionable approaches. Object orientation lends itself well to re-use, which is why there is a lot of interest. Re-use is seen as the "holy grail" of software engineering, however, the author remains sceptical as to its true value. If the process of creating software, even in ordinary C, is automatic, then the important aspect is not re-use of the code itself, as that can easily be generated, but rather re-use of the models from which it is derived. Also, C++ has many problems associated with its complexity [34], and therefore needs to be approached with caution.

Another development worthy of a mention is the advent of vehicle-based communications. Initially this has been primarily used to reduce wiring, by multiplexing signals on to a digital serial bus, such as CAN [35] or SCP [28]. This allows data to be shared between a number of individual electronic control units, and has led to a greater distribution of functionality, i.e. each system non longer has its own dedicated controller, but functions are often "spread out" between two or more control units. Lately, this philosophy has been extended further with new communication protocols, such as D2B (Domestic Data Bus) [36], which use optic fibres to reach ultra high data rates, suitable for real-time audio and video information. Such developments are the basis for full blown telematics systems, where communication not only takes place between in-vehicle systems, but also between the vehicle and the

transport infrastructure. Telematic devices will be "plug-and-play", just like consumer electronic devices in the home, such that customers will be able to choose what they want, and when they want it.

So what about the other major part of this Engineering Doctorate, safety? It can be seen that electronic systems are developing at a very rapid pace, and more control is being removed from the driver. How can all this technology be made safe? There have also been developments here too, such as the X-by-wire project [37], which aimed to demonstrate how steering, engine power and braking can all be safely brought under direct computer control. Safety implies integrity, and integrity implies confidence that the system will always perform in a predictable, specified manner, a situation which is not easy to conceive given the dramatic increase in complexity. However, by using approaches such as that described in this report for the S-Type electronic throttle, it is possible to combine the power of the technology with dependability. Safety-critical architectures are possible by using small simple cheap microprocessors as the basis for independent monitoring systems. These can be hand coded in a conventional way, for example using a safe-subset of C [22], thus avoiding any of the concerns about auto-code. However, all the benefits of auto-code, operating systems, object orientation, or anything else, can still be realised on the main part of the control system, provided its safe operation is always maintained within the envelope of the monitoring system. It is also reasonable to hypothesise about the use of distributed processing as a means of achieving fault tolerance. It may be possible to provide monitoring functions, redundancy and diversity by using the in-vehicle network to distribute such features across a number of control systems, such that, if a fault occurs, the system could re-configure itself.

In conclusion it can be seen that the automotive industry is facing rapid and wide ranging changes, and that electronic control systems development are at the forefront of that change for new product development. The business pressures are intense – faster, better, cheaper – whilst at the same time the speed of new developments are accelerating. The specific electronic control systems issues, tackled in this Engineering Doctorate portfolio, have dealt with the "here and now", and helped to establish Jaguar in a good position to flourish as part of the Ford Motor Company in the future. It has been demonstrated that Jaguar is able to deliver new technology to the market, not just more quickly and cheaply, but also safely.

6. References

[1] Hatton, L., *Safer C: Developing Software for High-Integrity and Safety-Critical Systems.* McGraw-Hill, 1995. ISBN 0-07-707640-0.

[2] MISRA *Development Guidelines for Vehicle Based Software.* Published by the Motor Industry Research Association, Nuneaton, UK on behalf of the Motor Industry Software Reliability Association. November 1994.

[3] PASSPORT II. *Framework for Prospective System Safety Analysis, Volume 1 - Preliminary Safety Analysis.* European Union DRIVE II Project V2058. December 1995.

[4] PASSPORT II. *Framework for Prospective System Safety Analysis, Volume 2 - Detailed Safety Analysis.* European Union DRIVE II Project V2058. December 1995.

[5] QS9000 Quality System Requirements. Chrysler Corporation, Ford Motor Company and General Motors Corporation. Feb 1995. Available from Carwin Continuous Ltd. Thurrock, Essex UK.

[6] Wright C. J., Product Liability -The Law and its Implications for Risk Management. Blackstone Press, London. 1989. ISBN 1-85431036-4.

[7] Product Liability and Computer Software. The Effect of the Consumer Protection Act 1987, Eversheds Legal Services Ltd., London. November 1989.

[8] IEC 61508 : *Functional safety of electrical/electronic/ programmable electronic safety-related system.*, Parts 1-7. International Electro-technical Commission, Geneva, Switzerland. Sub-committee 65A, draft for public comment, 1995. [Previously IEC 1508].

[9] *Transport Statistics of Great Britain.* 1996 Edition. The Department of Transport. HMSO, London.

[10] Kendall, I. R. *The Safety Assurance of the XK8 Electronic Throttle.* Digest no. 96/281. Proceedings from IEE Special Colloquium on The Electrical System of the Jaguar XK8. 18th October 1996. The Institution of Electrical Engineers, London.

[11] *Failure Mode and Effects Analysis Handbook.* Automotive Safety and Engineering Standards. Ford Motor Company, December 1995.

[12] Jesty, P. H. and Hobley, K. M. *Integrity Levels and their Application to Road Transport Systems.* Proceedings of the 15th International Conference on Computer Safety, Reliability and Security (SafeComp '96), Springer-Verlag, pp. 365-374, 1996, ISBN 3-540-76070-9.

[13] ISO 9000-3. *Guidelines for the Application of ISO 9001 to the Development, Supply and Maintenance of Software.* The International Standards Organisation, 1991.

[14] *TickIT.* A Guide to Software Quality Management System Construction and Certification using ISO 9001/EN 29001/BS 57450 Part 1. The DISC Project Office (DTi). Issue 2, 1992.

[15] Diller, A. Z - *An Introduction to Formal Methods*, 2nd Edition. John Wiley and Sons, Chichester, UK. ISBN 0-471-93973-0.

[16] Kendall, I. R. and Hobley, K. M. *A safety analysis methodology and its automotive application.* Automotive Electronics, Autotech Congress, 4-6 Nov 1997. IMechE Seminar Publication 1997-10. Institution of Mechanical Engineers, London. C524/208/97. pp79-90.

[17] Yourdon, E. N. *Modern Structured Analysis.* Yourdon Press, USA. 1991. ISBN: 0-135-98624-9.

[18] Andrews, J.D. and Moss, T.R.. *Reliability and Risk Assessment.* Longman Scientific and Technical. 1993. ISBN 0-582-09615-4.

[19] Hatley, D. and Pirbhai I. *Strategies for Real-Time System Specification.* Dorset House Publishing, New York, 1988. ISBN 0-932633-11-0.

[20] Mellor, S. and Ward, P. *Structured Development for Real-Time Systems*, Volumes 3. Prentice Hall, 1985.

[21] Cullyer, W. J., Goodenough, S. J. and Wichmann, B. A. *The Choice of Computer Languages in Safety-Critical Systems.* Software Engineering Journal 6(2): pp51-58, March, 1991.

[22] MISRA *Guidelines For The Use Of The C Language In Vehicle Based Software.* (MISRA C). Published by the Motor Industry Research Association, Nuneaton, UK on behalf of the Motor Industry Software Reliability Association. April 1998.

[23] McCabe, T. J. *A Complexity Measure.* IEEE Transactions on Software Engineering, Volume 1 SE-2 pp 308-320, December 1976. Cited in [24].

[24] *Software Metrics*, MISRA Report 5, February 1995. Published by the Motor Industry Research Association, Nuneaton, UK on behalf of the Motor Industry Software Reliability Association.

[25] *Automotive Engine Modelling for Real-Time Control Using MATLAB/SIMULINK.* Weeks, R. (Modular Systems, USA) and Moskwa, J. (University of Wisconsin-Madison, USA). SAE Technical Paper Series, No. 950417. The Society of Automotive Engineers, Warrendale, PA, USA, 1995.

[26] Crossley, P. R. *Modelling and Analysis of traction control systems in automobiles.* PhD Thesis, Dept of Engineering, University of Warwick, February 1992.

[27] Ross, B. and Savage S. (1996) "*The XK8 body electronics control system*". Digest no. 96/281. Proceedings from IEE Special Colloquium on The Electrical System of the Jaguar XK8. 18th October 1996. The Institution of Electrical Engineers, London.

[28] *Class-B Data Communications Network Interface.* SAE J1850, Society of Automotive Engineers, Warrendale, USA. July 1995.

[29] Harel D. "Statecharts: a visual formalism for complex systems," Science of Computer Programming, vol. 8, no. 3, pp231-274,1987.

[30] *Moving Ford Away From Mainline Thought.* Automotive Industries, December 1998, pp27-29. Cahners Business Information, CO, USA.

[31] *Automotive Electronics Overview. Spotting the trends for 1999 and beyond.* Automotive Industries, August 98, pp36-37. Cahners Business Information, CO, USA.

[32] Hanselmann, H. *Development Speed-up for Electronic Control Systems.* SAE technical paper 98C037. Proceedings from Convergence 98, 19-21 October 1998, pp247-261, Society of Automotive Engineers, Warrendale, USA. 1998.

[33] Open Systems and the Corresponding Interfaces for Automotive Electronics : OSEK. *Operating System Specification.* Version 2.0 revision 1. University of Karlsruhe, Germany. 15 October 1997. [http://www-iiit.etec.uni-karlsruhe.de/~osek/].

[34] Embedded C++. *The Language Specification & Libraries*, Version WP-AM-002. 8 Aug 1997. The Embedded C++ Technical Committee, ISO C++ (SC22/WG21). [http://www.caravan.net /ec2plus/index.html].

[35] ISO 11898:1993 *Road vehicles -- Interchange of digital information -- Controller area network (CAN) for high-speed communication.* International Standards Organisation, Geneva, Switzerland. 1993.

[36] IEC 1030 : 1991. *Audio, visual and audiovisual systems. Domestic Data Bus (D2B).* International Electro-technical Commission, Geneva, Switzerland. 1991.

[37] E. Dilger, L.A. Johansson, H. Kopetz, M. Krug, P. Lidén, G. McCall, P. Mortara, B. Müller, U. Panizza, S. Poledna, A.V. Schedl, J. Söderberg, M. Strömber, T. Thurner: *"Towards an Architecture for Safety Related Fault Tolerant Systems in Vehicles."* Proceedings of the ESREL'97 International Conference, Lisbon, 17-20 June 1997. European Safety and Reliability Association. [http://www.vmars.tuwien.ac.at/projects/xbywire/ projects/new-esrel97.html]

7. Acknowledgements

The author would like to thank the following individuals, without whose efforts and support this work would not have been possible:

EngD Academic Mentors:
Dr Peter Jones, Dr Steve Thomas

Jaguar Management:
Ian Smith, Pete Earp, Tony Davis, Jack Hynds (industrial mentor), Nick Barter

Ford Electronic Throttle Team (USA):
Steve Deasy, Paul Szuszman, Lynn McCormick, Steve Szwabowski, Mark Malone, Mike Lindlbauer, Boris Melnyck, Jonathan Chu, Ray Bzymek.

Jaguar Liaison (USA):
Ron Price

Jaguar X200 EMS Team:
Steve Large, Helen Monkhouse, Vaz Serghi, Lisa Johnson, Nik Wilkinson, Phil Whiffin.

York Software Engineering Ltd.:
Andy Coombes, Ron Pierce, Jenny Butler

HILST Team:
Huy Ho, Peter Bennett, Matthew Williams, Stuart Rooksby, John Hildreth.

Body Systems Team:
Barry Ross, Kevin Gumbrell, Steve Cox.

Jaguar Finance:
John Ryan

Jaguar Purchasing:
Samantha Goldsworthy

Cambridge Control Ltd.:
Chris Hayhurst, Dave Maclay, Clive Amos, Gavin Walker, Dave Roberts, Sham Ahmed.