Evolutionary Functional Testing of a Vehicle Brake Assistant System

Oliver Buehler* Dr. Joachim Wegener¶

*STZ Softwaretechnik
Entennest 2, D-73730 Esslingen, Germany
Oliver.Buehler@stz-softwaretechnik.de

¶Research and Technology, DaimlerChrysler AG
Alt-Moabit 96 a, D-10559 Berlin, Germany
Joachim.Wegener@daimlerchrysler.com

1 Introduction

The safety of today's vehicles is one of the major concerns of the automotive industry. Modern cars are equipped more and more with vehicle assistance functions in order to increase vehicle safety [10]. Vehicle assistance functions are realised using electronic control units (ECUs), which are embedded systems, designed for the use in a vehicle. It is of critical importance that the ECUs used in vehicles operate properly. Thus, the high quality of the ECU's software is an important requirement for a correct vehicle assistance function. Up to now, one of the most important measures to ensure software quality is extensive testing, but testing is time-consuming and costly. This is, because testing requires the definition and execution of test cases and the evaluation of the test results. Manual definition of test cases is difficult and time-consuming. Evolutionary testing has the potential to automate test case design.

A test environment for an evolutionary functional test of a brake assistance function will be described in this work. First, the brake assistance function is described, which is subject to testing as application in a vehicle. Second, an introduction to evolutionary testing is given. Furthermore, the theory and architecture for an evolutionary functional test of a brake assistance function is presented. After that, the practical experiments which were carried out and the obtained results are shown. Finally, a conclusion is drawn and some prospects for possible future work are given.

2 The Application Vehicle Brake Assistant System

A brake assistant (BA) of the first generation is a vehicle assistance function to help drivers in critical situations to slow down and stop their vehicles hard and fast. An example could be a situation, where an object suddenly appears in the path of the car, and the driver jumps on the brakes. In many cases the driver's braking is not hard enough, and in these situations, a driver gives away valuable meters of braking distance. To support the driver in such an emergency braking situation, the BA measures the velocity of the brake pedal and decides within a fraction of a second whether the driver intends to make an emergency stop. Once the BA has recognized an emergency braking situation, it immediately initiates full braking pressure, regardless of the drivers brake pedal position. The effect is, that within the first second of the emergency braking situation, the full braking pressure is established much faster than without the aid of a BA. This saves valuable braking distance at a high vehicle speed from the beginning of the situation, which could ultimately be essential in avoiding or mitigating a collision. In a practical verification experiment conducted by the German ADAC [11] with average drivers, on full braking from a speed of 100 km/h, in many cases the braking distance was reduced by several meters.

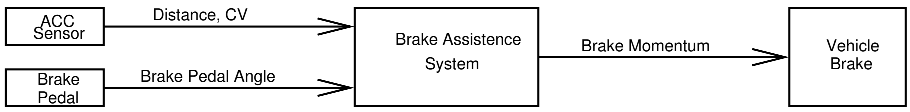

Figure 1: Principle of an Improved Brake Assistance System.

Driving assistance systems of the second generation make use of the interconnection of sensors, actors and ECUs. These can already be part of the vehicle as components of safety and comfort systems. An improved brake assistant, as shown in Figure 1, can use e.g. the information from a long-range sensor of the adaptive cruise control (ACC) to detect critical traffic situations. Then the ACC sensor passes the information regarding distance and closing velocity (CV) of the object further on in the lane, to the brake assistant. So the BA can rate the situation based on two sources of information. In addition to that, using the distance and CV values from the ACC sensor, the BA can calculate a suggested value for an ideal deceleration of the vehicle, in order to avoid a collision. On the other hand, the additional sources of information due to the interconnection of sensors, actors and other ECUs makes the improved brake assistance system much more complex, compared to a brake assistant of the first generation. An additional source of input means additional dependencies and increases the number of input situations to be handled by the implementation of the function. This complexity and the practical relevance of the subject makes the brake assistance system an interesting application for a study about the feasibility of an evolutionary functional test environment.

3 Introduction to Evolutionary Testing

The systematic test is an inevitable part of the verification and validation process for embedded systems. Testing is aimed at finding errors in the system under test and giving confidence for its correct behaviour by executing the system with selected input situations. From all of the test activities, test case design is assigned decisive importance. Test case design determines the quality of the test, since the selection of the test data, to be used to test the test object, establish the kind, scope and thus the performance of the test.

Evolutionary testing is characterized by the use of meta-heuristic search techniques for test case generation. The test aim considered is transformed into an optimization problem. The test objects input domain forms the search space in which test data that fulfils the respective test aim is searched for. Due to the non-linearity of software (if-statements, loops etc.) the conversion of test problems into optimization tasks mostly results in complex, discontinuous, and non-linear search spaces. Therefore, meta-heuristic search methods are employed, e.g. evolutionary algorithms, simulated annealing or taboo search. In our work, evolutionary algorithms are used to generate test data because their robustness and suitability for the solution of different test tasks has already been proven in previous work, e.g. Wegener and Grochtmann [7].

A number of papers have been published in the last years which have successfully applied evolutionary algorithms for test data generation. They have pursued various test goals with different test methods: Work on the automation of test data generation for structure tests is most widespread, e.g. Jones et. al. [4], Tracey [6], Wegener et. al. [9]. Further work deploys evolutionary algorithms for testing non-functional properties, such as safety constraints (Tracey [6]) or timing constraints (Wegener and Mueller [8], Gross et. al. [3], Puschner and Nossal [5]). The fitness functions are based on the calculation of a distance to the violation of the safety conditions or, for the temporal behaviour test, on the measurement of the execution times for the test data generated.

The application of evolutionary tests for functional properties is not widespread. Singular work by researchers such as Jones et. al. [4] applies evolutionary algorithms to generate test data for formally specified test cases. Initial work which allows complete automation of the functional test, without using formal specification techniques was presented by Buehler and Wegener ([1] and [2]). In the following, we present an new application area for evolutionary functional testing: automation of the functional test of the brake assistance system.

4 Evolutionary Functional Test of the Brake Assistant

As considered in the previous sections, the application of the brake assistance system is complex because of a huge number of possible test inputs, due to the interfaces to many sensors and ECUs. The demanded quality of the ECU's software requires extensive testing of the system. However, test case design is difficult, time-consuming and costly. So the objective of the evolutionary functional test is to automate test case design. In order to accomplish this aim, the problem of test case design for the application is transformed into an optimisation problem. To do this, two problems have to be solved. First, a representation of test inputs in a search domain has to be designed. Second, an adequate fitness function to evaluate the test results has to be found. Finally, these components have to be combined, with the implementation of an evolutionary algorithm, to the test environment.

4.1 Test Data Generation

Models for test data generation have to solve the problem, how individuals can be used to represent test inputs. Several different models are possible for the brake assistant to transform individuals into test inputs. The models have to ensure, that all possible input scenarios can be generated.

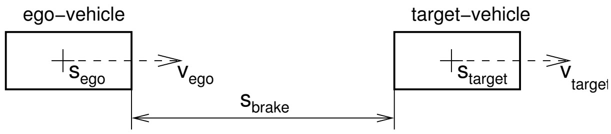

Figure 2: Rear-End Approaching Scenario.

Very complex models, which can produce an enormous variance of different scenarios have the problem, that the number of variables required by these models increases quickly leading to a multidimensional search space. Therfore, for this study we choose a simplified model, which only requires three variables as parameter settings for a collision maneuver.

The three variables in this model have the following semantics. First, the velocity v_{ego} of the ego vehicle, second the velocity v_{target} of the target vehicle, and third the trigger distance s_{brake} marking the initial moment of braking for the ego vehicle, as shown in Figure 2.

4.2 Design of a Fitness Function

The design of a suitable fitness function, used to evaluate the generated scenarios of the test runs, is a vital component of the test environment because its design determines how well the search converges towards successful test cases, which reveal errors in the implementation of the unit under test. For this application, the fitness function is based on two quantities, time-to-collision and brake momentum.

The *time-to-collision* (TTC) is a quantity which is often considered in the context of collision critical situations. It describes the time that remains before a collision happens. To calculate its value, the constant speed of all objects involved is assumed. For every moment t of the situation, the distance $s_{dist}(t)$ between the vehicles and the closing velocity $v_{cv}(t) = v_{ego}(t) - v_{target}(t)$ can be considered. With $v_{cv,min} = 0.002\frac{m}{s}$, the TTC is calcualted as

$$TTC(t) = \frac{s_{dist}(t)}{max(v_{cv}(t), v_{cv,min})} \tag{1}$$

and can be seen as a degree of *how critical* a situation is in that moment of the scenario. The higher the closing velocity and the lower the distance between ego vehicle and target vehicle is, the higher the vehicle retardation has to be in order to avoid a collision. The vehicle retardation is related with the brake momentum requested from the vehicle. A high momentum request from the vehicle brake should result in a high vehicle retardation. The brake momentum requested from the vehicle $M_{brake,vehicle}$, can be seen as the sum of two components, the brake momentum requested by the driver $M_{brake,driver}$ and the brake momentum added by the brake assistance system $M_{bas,add}$.

$$M_{brake,vehicle}(t) = M_{brake,driver}(t) + M_{bas,add}(t) \tag{2}$$

According to the requirements of the application, the brake assistance system should support the driver in a critical situation with a high vehicle brake retardation. On the other hand, the brake assistance system must not enhance the driver's brake momentum when the situation is not critical. The intention of the test is to find situations where the brake assistance system does not fulfil its requirements. For the brake assistance system, from a tester's point of view, two kinds of situations are interesting. First, when the brake assistance system does not enhance the brake retardation, even though the situation is critical. Second, when the brake assistance system does enhance the brake retardation in a situation which is not critical. To put this into a fitness function the following approach is used: The TTC is a measure of how critical a situation is (in that moment), where a high TTC value corresponds to an uncritical situation and a low TTC value represents a critical situation. The brake momentum added by the brake assistance system is a measure of the enhancement of the brake retardation in that moment, where a high added momentum shows an intensive enhancement and a low added momentum represents no enhancement of the braking. The situations of interest can now be expressed using this measures.

Measures for both situations can be translated for a single moment into the brake assistance added momentum $M_{bas,add}$, multiplied by the time-to-collision value TTC. To get one single fitness value for the overall situation, all values are integrated over the time.

$$v_{fitness} = \int TTC(t) \times M_{bas,add}(t)\ dt \tag{3}$$

When we replace TTC and $M_{bas,add}$ by the corresponding expressions, based on values measurable within the test environment, we come to

$$v_{fitness} = \int \frac{s_{dist}(t)}{max(v_{cv}(t), v_{cv,min})} \times (M_{brake,vehicle}(t) - M_{brake,driver}(t))\ dt \tag{4}$$

For the first situation of interest, when the brake assistance system provides low or no brake enhancement at all in a critical situation, the result of the fitness value should be low, because no or low values of $M_{bas,add}$ appear combined with low values of TTC for a long period of the situation. Vice versa, for the second situation, where the brake assistance system enhances the drivers braking in an uncritical situation, the corresponding fitness value should be a high value, because during high values of TTC a high value of $M_{bas,add}$ appears for a longer period of time. A restriction for this kind of fitness calculation is that it fails when the brake assistance system is not activated during the situation at all. To ensure this during the optimisation process, bad fitness values are assigend to situations where the brake assistance system was not activated during the situation.

5 Experiments and Results

With the components described in the previous section, a test environment for an evolutionary functional test of the brake assistance system was composed. The test environment comprises an implementation for a brake assistance system, a simulation environment for rear-end approaching scenarios, the previously described model for generating test data and an implementation of the fitness function presented in the previous section.

The evolutionary algorithm performs the search and passes the DNA values from its individuals to the test data generator. The DNA values are translated into simulation parameter values, v_{ego}, v_{target} and s_{brake}. These parameter values are given to the simulation environment, which comprises components for the environmental-, vehicle- and driver-simulation and an implementation of the brake assistance system, the unit under test (UUT). With that, it performs the simulation of the rear-end collision maneuvers, according to the given simulation parameters. The simulated braking scenario is presented to the fitness function, where a fitness value is calculated from, corresponding to equation (4). The scenarios in the simulation environment always start with the generated velocities v_{ego} for the ego vehicle and v_{target} for the target vehicle with a fixed distance between the vehicles. During the scenario, when the distance between ego and target vehicle falls under the given parameter value s_{brake}, a predefined and fixed course of driver braking momentum is started.

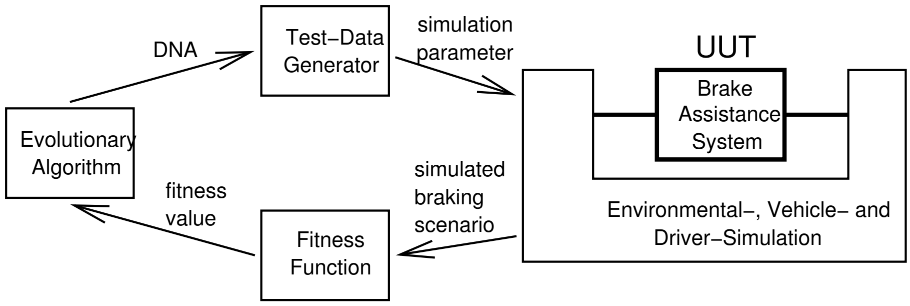

Figure 3: Test Environment for the Brake Assistance System.

All experiments were performed using a test environment with this configuration. The fitness function was configured to assign good fitness values for high values of $v_{fitness}$ from equation (4). The evolutionary search was configured with a population size of 80 individuals and a search length of 100 generations as stopping criteria. The borders for the search domain were beween 10 and 50 m for s_{brake}, between -50 and -2 m/s for the closing velocity v_{cv} and between 0 and 20 m/s for the velocity of the target vehicle v_{target}.

The search finally converged to one kind of scenario, shown in Figure 4. The figure shows the time in seconds on the X-axis and the scale of the brake momentum in Newton meters on the Y-axis, except for the signal, labeled as 'brake assistant active', which is a boolean type. The dotted line shows $M_{brake,driver}$, the brake momentum initiated by the driver, labeled as 'driver brake momentum'. The solid line shows $M_{brake,vehicle}$, the brake momentum requested from the vehicle car brakes, labeled as 'vehicle brake momentum'.

Figure 4 shows that the driver's braking starts approximately at 5.5 s and increases up to 4000 Nm at about 6.3 s. The brake assistant activates at 6.2 s, and from then on adds brake momentum at the driver's request, to an initial value of 5093 Nm for the vehicle brake momentum $M_{brake,vehicle}$, which descends to the level of the drivers request over the next 1.3 s. After that, the vehicle brake momentum $M_{brake,vehicle}$ corresponds to the driver brake momentum $M_{brake,driver}$ for the next 2 s. So far, everything looks fine. But then, starting

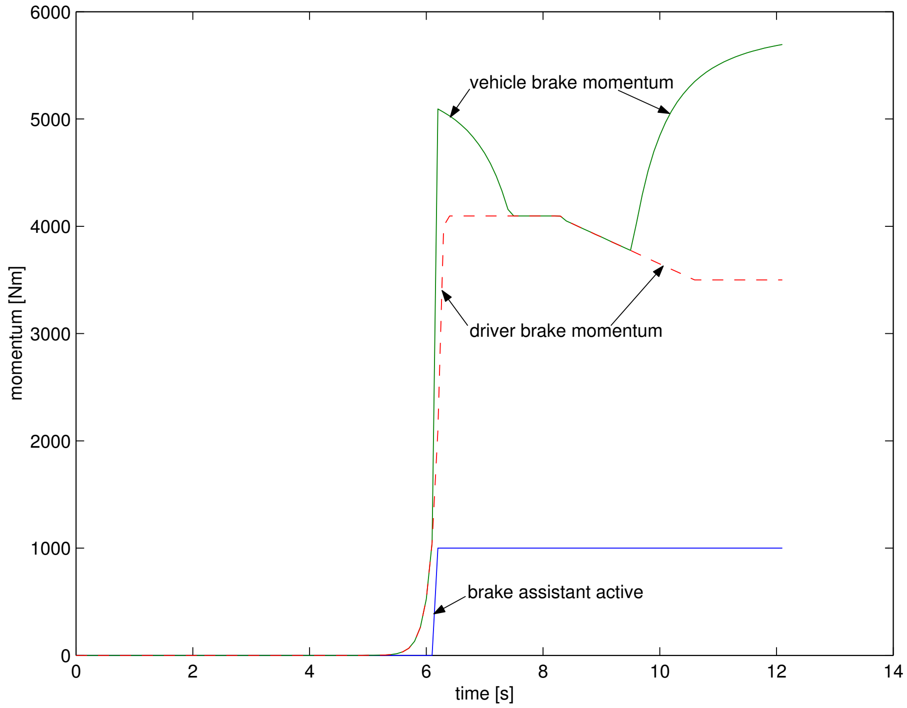

Figure 4: Brake Momentum of Situation Found During Search.

at time index 9.5 s, the brake assistance system again starts to add brake momentum, and increases the vehicle brake momentum $M_{brake,vehicle}$ to a level of 5680 Nm at time index 12 s. This is an inappropriate reaction of the brake assistance system in that situation, because the situation was already uncritical by that point.

6 Conclusion and Future Work

In this work evolutionary functional testing was applied for the testing of a brake assistance system, which supports the driver's braking maneuvers by increasing the brake momentum in critical situations. For the brake assistance system a model for test data generation and fitness evaluation was developed and a test environment was designed to entirely automate the test. The test environment was capable of automaticly finding situations, where the implementation's behaviour deviated from the defined functional requirements for the system. It turned out that the implementation used for the simulation contained errors. The application showed that the test environment with the chosen fitness function converged against situations where additional braking performed by the assistance system is inappropriate. Those situations found by the evolutionary functional test were no boundary situations, which are typical candidates for conventional test case design. So, it would be unlikely to find these error situations with conventional test techniques. Future work on this subject would comprise experiments with extended and more complex models to generate test data. In addition to that, further work on the design of fitness functions for this kind of application is conceivable.

References

[1] Buehler, O., and Wegener, J. (2003): "Evolutionary Functional Testing of an Automated Parking System". Proceedings of the International Conference on Computer, Communication and Control Technologies (CCCT '03) and the 9th. International Conference on Information Systems Analysis and Synthesis (ISAS '03), Florida, USA.

[2] Buehler, O., and Wegener, J. (2004): "Automatic Testing of an Autonomous Parking System Using Evolutionary Computation", at the 2004 SAE World Congress, Detroit, Michigan, USA, March 8-11, 2004. SAE Technical Paper Series 2004-01-0459.

[3] Gross, H.-G., Jones, B., and Eyres, D. (2000): "Structural performance measure of Evolutionary Testing applied to worst-case timing of real-time systems". IEE Proc.-Softw., vol. 147, no. 2, pp. 25 30

[4] Jones, B. F., Eyres, D. E., and Sthamer, H.-H. (1998): "A Strategy for using Genetic Algorithms to Automate Branch and Fault-based Testing". The Computer Journal, vol. 41, no. 2, pp. 98 107

[5] Puschner, P., and Nossal, R. (1998): "Testing the Results of Static Worst-Case Execution-Time Analysis". Proc of the 19th Real-Time Systems Symposium, pp. 134 143

[6] Tracey, N. (2001): "A Search-Based Automated Test-Data Generation Framework for Safety-Critical Software". Dissertation, University of York, Department of Computer Science.

[7] Wegener, J., and Grochtmann, M. (1998): "Verifying Timing Constraints of Real-Time Systems by means of Evolutionary Testing". Real-Time, Systems, vol. 15, no. 3, Kluwer Academic Publishers, pp. 275 298

[8] Wegener, J., and Mueller, F. (2001): "A Comparison of Static Analysis and Evolutionary Testing for the Verification of Timing Constraints". Real-Time Systems, Vol. 21, No. 3, Kluwer Academic Publishers, pp. 241 - 268

[9] Wegener, J., Baresel, A., and Sthamer, H. (2001): "Evolutionary Test Environment for Automatic Structural Testing". Information and Software Technology, Special Issue devoted to the Application of Metaheuristic Algorithms to Problems in Software Engineering, vol. 43, pp. 841 - 854

[10] Hyperlink: "All-Around Safety - The Vision of Accident-Free Driving Draws Near". `http://www.daimlerchrysler.com`

[11] Hyperlink: "So bremst der Assistent - Automatische Helfer sind ein Sicherheitsplus". `http://www.adac.de`