Evolutionary Functional Testing of Continuous Control Systems

Andreas Windisch
Daimler Center for Automotive IT Innovations
Technische Universität Berlin, Germany
andreas.windisch@dcaiti.com

Felix Lindlar
Daimler Center for Automotive IT Innovations
Technische Universität Berlin, Germany
felix.lindlar@dcaiti.com

Sebastian Topuz
Daimler AG
Group Research & Advanced Engineering
Sindelfingen, Germany
sebastian.topuz@dcx.com

Stefan Wappler
Berner & Mattner Systemtechnik GmbH
Automotive Division
Berlin, Germany
stefan.wappler@berner-mattner.com

ABSTRACT

Evolutionary functional testing is an approach to automatically generating test cases that violate a certain functional system requirement. This paper proposes an approach suitable for efficiently testing complex continuous control systems using an evolutionary testing framework that has been optimized for deployment in the industrial domain. Features of this test environment include the ability of automatically generating and optimizing continuous test data sets. The efficiency of the method is assessed by performing an experimental case study and by comparison with random testing. Results of this work indicate high usability and efficiency of the proposed method.

Categories and Subject Descriptors

D.2.5 [**Testing and Debugging**]: Testing tools

General Terms

Experimentation, Verification

1. INTRODUCTION

Evolutionary functional testing, a search-based approach to automatically generating relevant test cases, is a relatively recent technique that has been shown to be successful for numerous academic test objects as well as for various industrial ones [2, 4]. However, only little work has been done to apply this technique to systems that process continuous input signals. Two problems must be solved to perform an evolutionary functional test. (1) A suitable fitness function must be derived from the functional specification of the system under test. The fitness function evaluates every test run with respect to its criticality in terms of the system property under examination and assigns a fitness value to it. The fitness values are used to optimize the test data towards a failure of the system behavior. (2) The search space must be defined. Complex embedded systems often depend on a huge number of inputs. Effort has to be put into limiting the search space to the relevant data ranges to make the search feasible. This work introduces a test environment with the ability to generate continuous signals. In order to assess the efficiency of the proposed approach an extensive case study has been carried out. The system under test is an adaptive cruise control system (ACC) that is currently in a development state of serial production.

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.
ACM 978-1-60558-325-9/09/07.

2. TEST ENVIRONMENT

The Evolutionary Testing Framework (ETF) has been developed within the EU-funded project EvoTest [5]. The optimization engine of this framework is dynamically generated by the Evolving Objects library [3].

The creation of relevant test scenarios for dynamic systems in general, i.e. the creation of system input signals by arrays of numbers, is hard to realize for real-world models as these often require the signals to have a specific minimum length. In combination with a small sample rate, which is often used, the resulting size of the arrays representing the signals prevent the application of optimization engines because of the huge search space. In order to enable search for continuous signals, a suitable representation of the signals must be used which is more compact and results in much smaller search spaces than using the arrays that contain all sampling points.

In our work, the underlying idea of how to efficiently encode signals generally is based on the work of Baresel et al. [1]: a signal is considered to be the concatenation of a certain number of parameterized base signals. These signal segments can be specified using the three parameters amplitude, transition type and width. The amplitude of each signal segment is constrained to the signal boundaries as specified in the interface of the system under test. The transition type of the signal segments determines the way in which the starting amplitude is connected to the ending amplitude and thus highly influences the nature of the generated signal. Finally, parameter width influences the temporal extent of the signal segment; it is used to assure further variability of the generated signals [6].

3. EXPERIMENTS

This chapter provides a brief introduction to the ACC system and the driving scenario used in the optimization runs. Furthermore the idea of the fitness function design is described. The chapter concludes with the experimental setup and results.

Adaptive Cruise Control System.

The ACC system has two main functionalities: cruise control and distance control. The cruise control maintains the setpoint speed intended by the driver independently of the engine load. In case of a violation of the minimum distance to a preceding vehicle the distance control adjusts the speed by braking with limited braking power. If the maximum braking power of the system is insufficient to maintain a safe distance, the driver is alerted. The driver then has to take back control over the car to resolve the situation.

Test Scenario.

Two vehicles are taking part in the simulation: a vehicle that is equipped with the ACC system and a preceding vehicle. Both vehicles are driving in the same lane following a straight road. The preceding vehicle drives the speed that is provided by an input signal and varies during the progress of the simulation sequence. The goal of the test is to find a scenario where the minimum distance criterion is violated, i.e. an accident is inevitable, without a warning signal being raised.

Fitness Function.

In order to guide the evolutionary search towards a violating scenario, the fitness function must be designed accordingly. We designed the fitness function so that it takes the time to collision (TTC) as well as the time of the arrival of the driver warning into consideration. The warning signal is either 0 if a warning is not to be displayed, or it is 1 if the driver must be alerted. If the warning light is showing up, the time to collision is not relevant for us anymore. Thus, we can simply multiply the warning signal with a big value δ and add it to the TTC signal, so that these values are assuredly not the minimum values chosen by the min function. The fitness value is calculated as follows:

$$Fitness = min(TTC(t) + Warn(t) \cdot \delta) \quad (1)$$

Experimental Setup.

The experiments are carried out as a Model-in-the-Loop test. Both the closed-loop ACC control system and a virtual environment are provided as MATLAB SIMULINK models. During the test runs these models are executed with signals generated by the ETF. The utilization of the ACC control lever and the velocity of the preceding vehicle are the signals that ETF optimizes. We used the following search settings: a population consists of 100 individuals and selection is carried out by stochastic universal sampling with generation gap of 85%. Evolving a new generation is done by applying gaussian crossover, gaussian mutation and elitest reinsertion. The termination criterion is to reach a certain fitness value or the maximum number of generations. In order to acquire results with sufficient statistical significance, the test data generation process for both GA and random testing were repeated 30 times.

Results.

Figure 1 shows the convergence progress for the optimization runs that applied genetic algorithms and random data generation respectively.

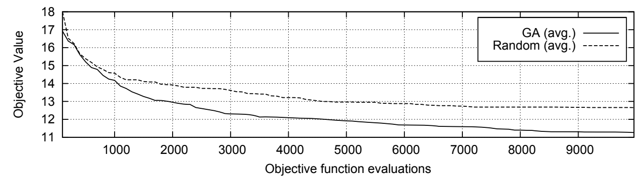

Figure 1: Convergence characteristics of GA and Random for the conducted experiment

By optimizing the driving scenario the ETF was able to find situations that can be classified the more critical the more the optimization progresses. As expected, the ETF was able to find better solutions than Random. However, both approaches were not able to find a driving scenario for which the ACC controller would incorrectly warn the driver too late. This is indeed not surprising because of the advanced development state of the controller software.

4. CONCLUSION

The results of the experiment show that evolutionary functional testing of embedded systems is a very promising approach to ensure safety requirements. Even though no driving scenario has been found that violates the requirement, the system under test has still been executed and thus assessed with a great number of different input stimuli. Hence it was possible to greatly enhance confidence in the system. Another big advantage of this approach is that after defining the fitness function to be used for optimization, no human effort is needed to execute this large amount of test scenarios. Future work will include more experiments examining further safety requirements or taking into account further test objects and tuning the metaheuristic search engine used for optimization.

5. REFERENCES

[1] A. Baresel, H. Pohlheim, and S. Sadeghipour. Structural and functional sequence test of dynamic and state-based software with evolutionary algorithms. In *Proceedings of Genetic and Evolutionary Computation Conference*, pages 2428–2441, 2003.

[2] O. Bühler and J. Wegener. Evolutionary functional testing of a vehicle brake assistant system. In *Proceedings of 6th Metaheuristics International Conference*, August 2005.

[3] M. Keijzer, J. J. Merelo, G. Romero, and M. Schoenauer. Evolving objects: A general purpose evolutionary computation library. In *Artificial Evolution*, pages 231–244, 2001.

[4] H. Sthamer and J. Wegener. Using evolutionary testing to improve efficiency and quality in software testing. In *Proceedings of 2nd Asia-Pacific Conference on Software Testing*, July 2002.

[5] T. Vos. Evotest. `http://www.evotest.eu`, Sept. 2006.

[6] A. Windisch and N. A. Moubayed. Signal generation for search-based testing of continuous systems. In *2nd SBSE-Workshop in conjunction with ICST*, April 2009.