# Abstract

Testing software-intensive systems can be challenging, especially when safety requirements are involved. Property-Based Testing (PBT) is a software testing technique where properties about software are specified and thousands of test cases with a wide range of inputs are automatically generated based on these properties. PBT does not formally prove that the software fulfils its specification, but it is an efficient way to identify deviations from the specification. Safety-critical systems that must be able to deal with faults, without causing damage or injuries, are often tested using Fault Injection (FI) at several abstraction levels. The purpose of FI is to inject faults into a system in order to exercise and evaluate fault handling mechanisms. The aim of this thesis is to investigate how knowledge and techniques from the areas of FI and PBT can be used together to test functional and safety requirements simultaneously.

We have developed a FI tool named FaultCheck that enables PBT tools to use common FI-techniques directly on source code. In order to evaluate and demonstrate our approach, we have applied our tool FaultCheck together with the commercially available PBT tool QuickCheck on a simple and on a complex system. The simple system is the AUTOSAR End-to-End (E2E) library and the complex system is a quadcopter simulator that we developed ourselves. The quadcopter simulator is based on a hardware quadcopter platform that we also developed, and the fault models that we inject into the simulator using FaultCheck are derived from the hardware quadcopter platform. We were able to efficiently apply FaultCheck together with QuickCheck on both the E2E library and the quadcopter simulator, which gives us confidence that FI together with PBT can be used to test and evaluate a wide range of simple and complex safety-critical software.