A Systematic Mapping Study on Test Automation Approaches for Flight Simulators

Linus Jensén

Master's Thesis

Supervisor: Adnan Ashraf

The Faculty of Science and Engineering

Åbo Akademi University

2022

Abstract

Background: Test automation is the practice of running tests automatically for the purpose of improving software quality. This is a field that is used in many kinds of areas. Flight simulators are software and hardware that help us achieve flight training and testing without flying. Test automation for flight simulators aims to improve the testing processes for this kind of software.

Objective: To present and provide a comprehensive, unbiased overview of the state of test automation approaches for flight simulators

Method: A Systematic Mapping Study (SMS) of the existing test automation approaches for flight simulators

Results: 20 papers whose main topic or mentions test automation approaches for flight simulators. The results show that there are multiple ways of achieving test automation, with the purpose of speeding up the testing process and this is mostly on the software level. This field is growing and has since 2007 grown with almost yearly publications.

Conclusions: This is a growing field, and it will likely continue to grow as technology advances and new testing methods and flight software emerges. Currently there is not any go to approach that is widely used, although there are several methods that are slightly more popular than others.

Table of Contents

List of Figures

List of Tables

1. Introduction

Software testing is a time-consuming process that can be avoided by automating it. Bigger projects such as flight simulator software consumes a lot of manpower and hours. Software testing would benefit from a standardized and automated testing process to determine software quality.

To understand where the field stands today and what approaches/methods are being used for test automation for flight simulators it has been deemed appropriate to conduct a Systematic Mapping Study (SMS) on this topic. The questions we want answered are the following:

- What approaches/ methods are in use?
- For what are they used?
- How are they implemented and how are they evaluated?
- What does the publication patterns look like? Is the field growing or declining?

To achieve this an SMS will be performed and the objective is to provide a comprehensive, unbiased overview of the state of test automation approaches for flight simulators.

The study proceeds as follows. Chapters 2 and 3 present the design of the study and how it was performed. Chapter 4 contains the results of the SMS. In chapter 5, we will discuss the findings. Chapter 6 will cover the threats to validity and chapter 7 will cover related works and chapters 8 and 9 will be the conclusion and a Swedish conclusion.

2. Related work

This chapter discusses related works in the field of automated testing and flight simulators. There are a few studies on the topic of this study, but systematic studies on separate topics of test automation and on different kinds of simulators exist. Here is an example of one similar study and two other studies, one focusing on software testing and the other on test automation for robotics.

Raulamo-jurvanen et al. [1] conducted a literary review on choosing the correct test automation tool. The review covers tool selection recommendations and processes.

Afzal et al. [2] conducted a study on test automation for robotics simulations. The study focuses on how developers use simulations in the testing procedures of robotics development. Ten challenges are identified that impede developers. An outline for improvement is provided as well.

Garousi et al. [3] have published a report on automated testing of simulation software in the aviation industry. The report reviews what the industry needs and why. It presents planning for test automation and tool requirements and introduces a test framework for test automation for a helicopter simulator.

3. Research Methodology

The topic of automated testing is a well-covered field, as multiple studies have been conducted on the topic [4] [1]. Flight simulation and flight testing are also fields that are well covered [5].

The subject of this study, test automation approaches for flight simulators, however, it is not a well-covered topic currently. Therefore, a systematic study mapping the current trends and approaches is warranted. This systematic mapping study will follow the methods outlined by Petersen et al. [6]. in which an outline of how to approach a mapping study is given in three phases, planning, conducting and reporting.

3.1 Planning

The planning phase of a systematic mapping study should include a valid need for the study. The research questions and a research goal should be defined. A research protocol should be made and followed. The research protocol should be followed when conducting the study.

3.2 Conducting

The results of the planning phase are set into practice in order to conduct the systematic mapping study. This phase follows three steps.

- Search and selection: In this phase the search strings that are worked out in the research protocol are put into use in the chosen digital libraries. The search strings are modified for each library formatting, if needed. Duplicates and irrelevant findings are merged or removed. Selection criteria are applied to find primary studies. Backward and forward snowballing is applied to further expand the list of primary studies. When these steps are concluded, a final list of primary studies is ready for extraction.
- Data extraction: In this phase the data extraction strategy that is set in the protocol is followed to extract the sought-after data in the primary studies
- Data synthesis: The extracted data is analyzed and summarized. The results of the data synthesis will be used to provide answers to the research questions.

3.3 Reporting

This section covers the final phase of the systematic mapping study where the extracted data and the mapping study are being discussed and main findings reported. The threats to validity of the study are discussed as well.

3.4 Research questions and goal

This work aims to identify and evaluate the following research questions:

RQ1. What approaches (methods, algorithms, techniques, frameworks) exist for test automation for flight simulators?

Rationale: There are numerous methods to conduct test automation, but what methods are relevant for flight simulators?

Outcomes: A classification of the methods used in test automation examples in the primary studies.

RQ2. What are the purposes for the test automation for flight simulators?

Rationale: To identify what is being automated and what is not.

Outcomes: A set of purposes for test automation in the primary studies.

RQ3. What is the level and scope of implementation of test automation for flight simulators?
Rationale: To identify the implementation levels of the test automation approaches in the primary studies.
Outcomes: A list of how test automation is implemented.

RQ4. How is existing test automation evaluated?
Rationale: How is the test automation approaches evaluated or are they at all?
Outcomes: A list of evaluation results for the primary studies.

RQ5. What are the publication trends regarding test automation for flight simulators?
Rationale: To identify the state of existing research on test automation for flight simulators.
Outcomes: A descriptive report on the publishing trends on the primary studies.

The research questions are based on the Population, Intervention, Comparison, Outcomes and Context (PICOC) criteria in accordance with the guidelines [7].
In table 1. The PICOC is presented

Table 1: PICOC

Aspect	**Value**
Population (P)	Flight Simulators
Intervention (I)	Test automation for flight simulators
Comparison (C)	No comparison intervention
Outcomes (O)	An overview of the state-of-the-art on test automation approaches
Context (C)	Test automation for flight simulators

3.5 Search and selection process

The search and selection process are a vital part of the multiple stages that a systematic mapping study consists of. It is important that this process is documented well to allow the study to be replicated so that the results can be verified.

The search and selection process consists of a initial search of the chosen databases and libraries. The result of this search consists of duplicates and other irregularities that need to be removed. Therefore, the next step is merging and removal of impurities. The next step is title and abstract review to further filter out unwanted studies. After this snowball sampling is performed to complement the initial search. The next step is the data extraction, where relevant data is used to form the basis of the study. The last step is the data synthesis. In figure 1. We can see an illustration of this process.

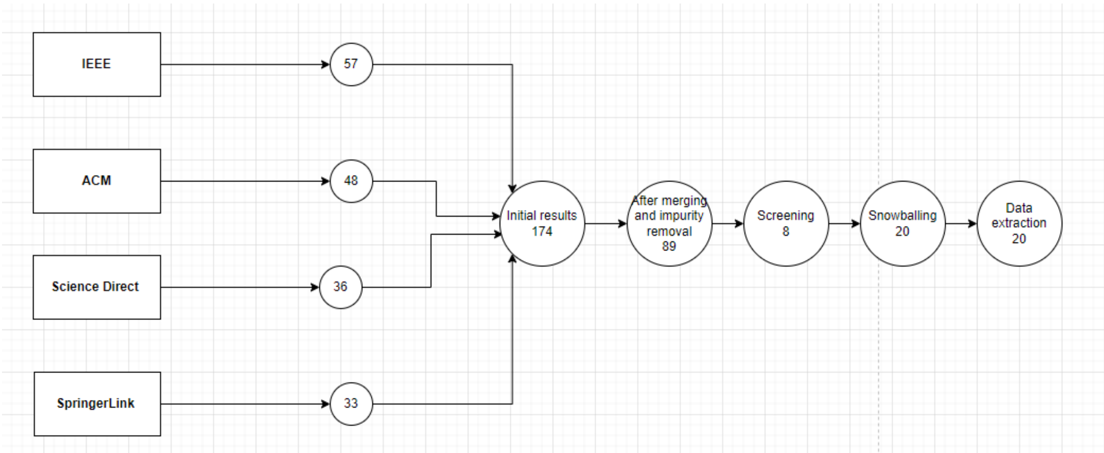

Figure 1: Selection Process

3.5.1 Initial search

The initial search will be conducted on four sources. The libraries used in the search are IEEE Xplore Digital Library, ScienceDirect, ACM Digital Library and SpringerLink. All four databases are widely used and host huge libraries of peer-reviewed publications. The search will be performed with a search string. That will allow us to search the databases for papers containing important keywords in the Titles and abstracts.

In Figure 1 the initial results are shown. We can see that 57 papers were found on IEEE, 48 on ACM, 36 on ScienceDirect and 33 on SpringerLink.

The advanced search features on these databases were used to fully utilize the search string. Extraction of the results were either done by built-in functions or by hand.

The syntax of the search string that will be used is as follows:
(("Abstract":Test* AND "Abstract":automation) AND (("Abstract":flight OR "Abstract":aviation) AND "Abstract":simulat*))
Different variations of this string will be used on the databases, since support for the syntax or wildcards does not exist on all four. SpringerLink, for example, had to use a heavily modified search string in the form of:
((“Test automation”) AND ("Flight simulator"))
Then some special options had to be chosen to narrow down the search as well. Simulation and modelling were chosen as a subdiscipline, and Conference paper was chosen as content type.

3.5.2 Merging and irregularity removal

Duplicate search results from the different databases will be located and removed. If an older edition of a paper is found, the newer will be kept. Papers that do not fit the search criteria will also be removed. The criteria in question demand that the papers are conference proceedings, journal or workshop. Also, any papers not written in English were to be removed.

3.5.3 Application of selection criteria

After the merging, the papers will have to pass the set selection criteria that help us determine if the results can answer the research questions. Therefore, both inclusion and exclusion criteria were defined before the mapping study.

The inclusion criteria for primary studies are as follows:

- Covers test automation approaches for flight simulation AND
- Written in English AND
- Published in a peer-reviewed journal, conference, or workshop of computer science, computer engineering, or software engineering.

In addition, if several papers present the same test automation approach only the most recent one will be included.

The exclusion criteria for primary studies are as follows:

- Does not cover test automation approaches for flight simulations OR
- Not written in English OR
- Not published in a peer-reviewed journal, conference, or workshop of computer science, computer engineering, or software engineering.

In addition, if several papers present the same test automation approach, only the most recent one will be excluded.

If the selected papers do fulfil the criteria, they will be selected for data extraction; otherwise they were discarded as not relevant.

3.5.4 Snowball sampling

The next step done will be forward and backwards snowballing. The aim here is to complement the primary studies found left after screening.

Forward snowballing consists of finding other papers that cite the paper that is currently being snowballed. Google scholar is a good tool that gives a collected list of all papers that cite the current one [8].

Backward snowballing consists of going through the reference list in the target study and finding relevant papers that have been used. Here services like google scholar comes in good use in finding these papers.

When a relevant paper has been found, the title and abstract should be screened. If that is inconclusive, important sections of the paper may also be screened. If any of these fit in with the selection criteria, the paper can be used to complement the study [8].

3.6 Data extraction

Here the purpose is to obtain the relevant information from the primary studies. The data extraction is presented in table 2.

Table 2: Data extraction table

Data Item	Value	Additional notes
General		
Data extractor name		
Data extraction date		
Study identifier		
Bibliographic reference (title, authors, year, journal/conference/workshop name)		
Author affiliations and countries		
Publication type (journal, conference, or workshop)		
Test automation related		
RQ1: approaches (methods, algorithms, techniques, frameworks)		
RQ2: Purpose of test automation		
RQ3: Level and scope of implementation		
RQ4: Evaluation method		

The table is divided into two parts, general information and test automation-related information. Next, we will review the two parts starting with the general side.
The data extraction table will be used to gather the required information, in this case name of the extractor as well as date of the extraction, and also a simple identifier is given. Information about the study is then extracted, such as title, names of authors, year of publication and the name of the publication (conference, journal or workshop

name). What are the authors affiliated with and what is the publication type (Conference proceeding, journal or workshop)? After the general questions are answered, we will move over to part two of the table.

The second part focuses on the research questions. Thus, in this part we will review the research questions.

First RQ1 will be extracted which is covering the test automation approaches in the study. Then we will extract data relevant to RQ2, the purpose of the approaches in RQ1. After RQ2 we will answer RQ3, which is level and scope of the found approaches. In RQ4, the evaluation methods for the approaches will be extracted and the final point of interest will be RQ5, which will be cataloguing the publication years of each study.

3.7 Data synthesis

The next step after the data extraction is the data synthesis where the extracted data is analyzed and summarized in a representable way. The data and results from the analysis will be presented both in graphs, tables and in text form where trends and outliers will be detailed.

4. Conducting the systematic mapping study

In this chapter, we will describe the execution of the study. This will be based on what was covered in chapter 3.

4.1 Search and selection process

4.1.1 Initial search

As stated in chapter 3.5.1, the libraries used in this study are IEEE, ACM, SpringerLink and ScienceDirect. Here we utilized the search string that was created for the purpose of finding studies that cover the topic of this study. Initially, four different search strings were planned to be used, but during the initial search process

it was decided that one search string would suffice. The search string had to be adapted to each library's rules to be able to produce results.
The search string used was presented in chapter 3.5.1. The search resulted in 174 studies. In table 3, the number of studies from each database can be viewed.

Table 3: Studies after first search

Source	Amount
IEEE	57
ACM	48
SpringerLink	33
ScienceDirect	36

4.1.2 Title and abstract screening

This phase was done in tandem with the merging phase. A title and abstract screening process was done to filter out studies that did not fit the search criteria at all. Several studies could be removed on a title analysis alone, but most studies required screening of the abstract as well. In this phase, keywords and sentences were looked for and if it could be deemed that the study did not cover the desired topic, the study was revoked and the next study was screened. If the study was on topic or possibly on topic, during the title and abstract screening, it was put forward for further screening. In this phase, 41 studies were removed leaving 133 for the next phase.

4.1.3 Merging and impurity removal

In the next phase, duplicates and other irregularities were removed. Since multiple databases were used with one search string, the likelihood of duplicate studies is substantial. For this purpose, a program called JabRef was utilized to organize and filter out duplicate studies. Studies written in other languages were also looked for, none were found. This phase removed 44 studies, leaving 89 studies.

4.1.4 Application of selection criteria

During the whole process so far, we have utilized the criteria set in the inclusion and exclusion criteria as well as the research questions to gauge the suitability of the found studies. During the initial screening, a large part of the studies was discarded due to not being related to the topic. The remainder of the studies fit into the inclusion criteria in most ways and a full text screening was needed for them. An initial approach of searching for specific keywords, such as automation and testing, was chosen to find suitable sections in the studies to screen. If the paper was deemed useful it was put forward, if the paper was deemed inconclusive the whole text was screened and if no mention of test automation was mentioned the paper was rejected. During the full text screening, it was noticed that the majority of the papers contained the correct keywords of automation and testing but in the wrong context. Most papers were focused on testing of autopilots, which was close to the main topic of this study but not what was looked for. This caused some concern, because at this point over half of the studies had been rejected.

After consulting with the thesis supervisor about the search string suitability, the decision was made to finish the remaining papers. In the end, eight out of the 89 papers left for the full text screening were acceptable.

4.1.5 Snowballing

The next step after this was snowballing. Both forward and backward snowballing was chosen, and was done on the eight remaining studies. It was decided this was the best approach, since none of the other studies were relevant and would most likely not yield any results. The first approach was backwards snowballing which consists of going through the references in the studies. Here four additional studies were found.

The next step was to perform the forward snowballing. In this step, we looked at citations for the studies and an additional four studies were added.

At this point, we had 16 primary studies. The supervisor was consulted again and permission to freely search for additional studies was given. This search was performed through Google Scholar and an additional four studies were found, which led to the final tally of studies being 20.

The 20 studies can be found in the following table, S1 or Study One would point to reference 4 which would be Automation in Experimentation with Constructive Simulation by Hodicky et al. [9].

Table 4: The primary studies

Study identifier	Reference no
S1	[9]
S2	[10]
S3	[11]
S4	[12]
S5	[13]
S6	[14]
S7	[15]
S8	[3]
S9	[16]
S10	[17]
S11	[18]
S12	[19]
S13	[20]
S14	[21]
S15	[22]
S16	[23]
S17	[24]
S18	[25]
S19	[26]
S20	[27]

Figure 2 Word cloud of titles and abstracts of the 20 primary studies

In figure 2, we can see the most common words in the titles and the abstracts in the 20 studies that were used. We can conclude that software was the most used word; it appeared 55 times in the 20 papers, flight software came second with 21 mentions.

4.2 Data Extraction

When the final 20 studies were selected the task of extracting useful data started. For this the table that was shown in table 2 was used to help gathering relevant data. Authors and the venues the studies were published in were documented as well as publication years and what affiliations the authors have. After the easy data had been extracted, the slightly more complex part of finding data for the research questions started.

Here it was decided that we will focus on one paper at a time so that documenting becomes simpler. Some of the research questions proved more difficult than others to answer, RQ1 was rather simple to answer, since it was very straightforward. The same can be said for RQ2; the purpose for the test automation was usually stated in clear text. RQ3 went easy too; here it was a question of figuring out the implementation level of the test automation approach that was used. The most difficult task was to find proper answers for RQ4 which was how the test automation approach should be evaluated. Not every paper had a clear evaluation method for the test automation approach, most likely since the topic of most papers was not solely focused on test automation.

The simplest research question to answer was RQ5, which was to collect the publication years and present them to learn when research has been done in the subject.

5. Results

In this chapter, we present the results from the vertical and horizontal analysis of the extracted data.

5.1 Research Question analysis

The purpose of the vertical analysis is to provide quantitative results regarding the research questions.

5.1.1 Result analysis of RQ1

This chapter covers the analysis of Research Question 1, “What approaches exist for test automation for flight simulators?”

As we can see in figure 2, there are a variety of different approaches. Some approaches were used in multiple studies, and some were unique to their study. Some studies did not mention a specific approach, they only mentioned a very broad framework or model; hence, some results were vague.

We can see that most approaches were only used once, for example, smoke testing was only mentioned in one study, while unit testing was used in three studies.

Figure 3 is a graph showing the occurrence of different approaches in each study. We can see that half the studies include more than one method for test automation.

Out of the 50 percent that include more than one method, 20 percent have four methods, and two and three methods both sit on fifteen percent each. This can be seen in table 5.

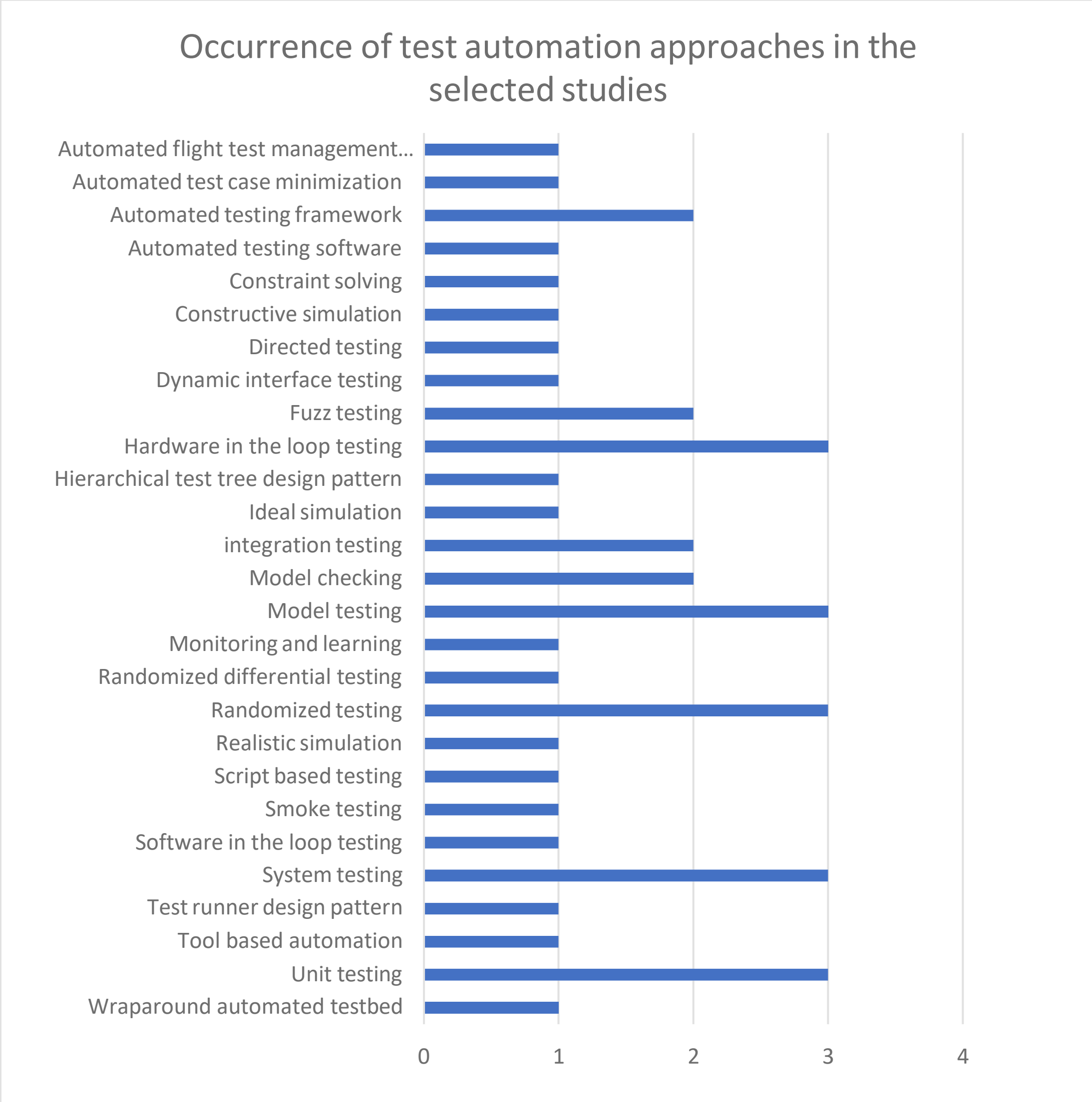

Figure 3: Occurrence of test automation approaches in the selected studies

We can conclude from figure 3 that the most used approaches were system testing, model testing, randomized testing, unit testing and hardware-in-the-loop testing. These approaches occurred in studies S3, S8, S9, S10, S11, S12, S13, S17, S18, S19 and S20. In table 3, the individual approaches are listed; here we can see their names and in what studies they appear as well as the number of studies they appear in.

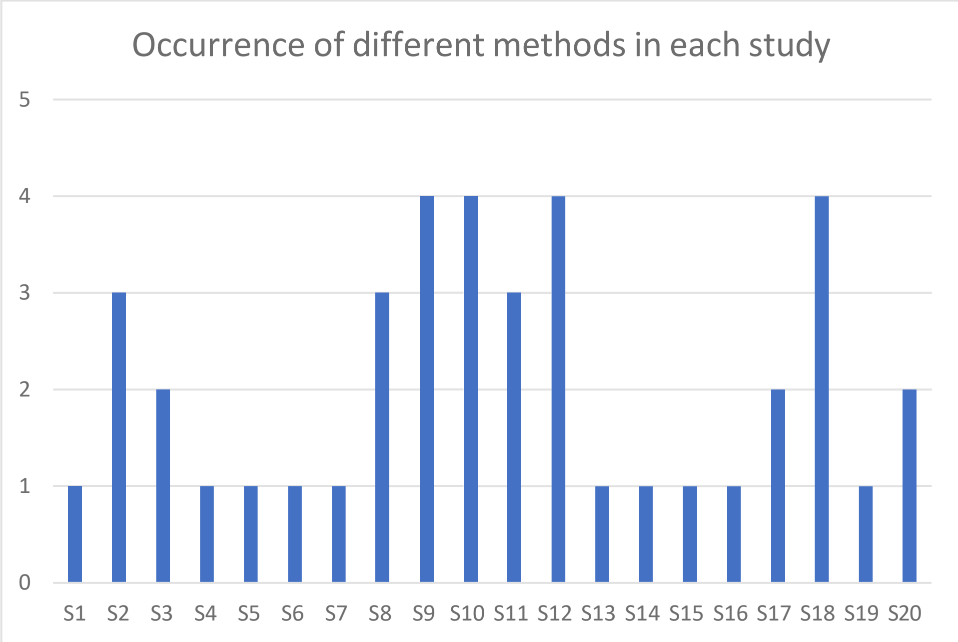

Figure 4: Occurrence of different methods in each study

Table 5: Number of approaches per study

Number of approaches	**Number of studies**	**Percentage**
1	10	50 %
2	3	15 %
3	3	15 %
4	4	20 %

Table 6: Types of approaches.

Approach	**Study IDs**	**Amount**
Constructive simulation	S1	1
Tool based automation	S2	1
Hardware in the loop testing	S3, S10, S20	3

Dynamic interface testing	S4	1
Wraparound automated testbed	S5	1
Fuzz testing	S6, S16	2
Automated flight test management system	S7	1
Unit testing	S8, S9, S18	3
Model checking	S11, S17	2
Randomized testing	S11, S12, S17	3
Model testing	S13, S17, S19	3
Automated testing software	S14	1
Automated testing framework	S15, S20	2
Integration testing	S8, S18	2
Automated test case minimization	S2	1
Script based testing	S3	1
System testing	S8, S9, S18	3
Software in the loop testing	S10	1
Constraint solving	S11	1
Randomized differential testing	S2	1
Test runner design pattern	S9	1
Ideal simulation	S10	1
Directed testing	S12	1
Hierarchical test tree design pattern	S9	1
Realistic simulation	S10	1

Monitoring and learning	S12	1
Smoke testing	S18	1

5.1.1.1 Definition of approaches

In this section, we will briefly cover the approaches and group them together in families of approaches.

Constructive simulation is a simulation where simulated people operate simulated systems. It is being manipulated by real operators, but the outcome is decided by the simulation itself. Constructive simulation is part of the Live, Virtual and Constructive simulation way of classifying simulations by the United States Department of Defence [9].

Tool-based automation is a software tool that helps in automating software testing. This is a tool that enables us to define tasks and then run them with minimal interaction from the user. There are different classes of tools, for example, codeless tools and code-based tools. A codeless tool is a tool that requires no coding, while a code-based tool requires writing code to work [28].

Hardware-in-the-loop (HIL) testing is an approach that includes real hardware either fully or partly in the testing. In the study that contains hardware-in-the-loop testing, different hardware parts are involved in the testing to create as realistic as possible a testing environment [29]. HIL testing is usually done later in the development phase, since it is more expensive and complex than Software-in-the-loop testing (SIL). Both methods are usually used together in different stages of testing.

Dynamic interface testing is a testing approach that specifically focuses on rotorcraft in adverse conditions, such as strong winds and seas. This sort of testing is used to simulate dynamic environments, such as ships in rough seas where a helicopter might land. [30].

Wraparound automated testbed is a setup of connected software and hardware to enable automated testing [13]. In the case of S5, the wrap around testbed contains various ground support equipment that are designed to test the power system electronic box in real time. Other components include a simulated solar array and a battery simulator [13].

Fuzz testing is an automated software technique that is based on automatically generating random input to find software failures [14]. In this case, it would mean outcomes other than the expected ones. In study S6, fuzz testing was applied on the SUCHAI flight software. Random input to functions, modules and commands were sent [14].

Automated flight test management system is a program that focuses on applying interdisciplinary state-of-the-art technology in AI, control theory, and systems methodology to problems of operating and flight-testing high-performance aircraft [15].

Unit testing tests individual units or groups of related units. Usually a test is written to test a specific section of a program, for example, the login function or a submit button [31]. An example of a common unit test is Junit which is a testing framework for the Java programming language. Another example used in one of the studies is Doruk, which is a Python-based testing framework quite similar to xUnit frameworks such as JUnit [3].

Model checking is an automated technique for verifying finite-state systems, for example, communication protocols [32]. An example of model checking would be testing that an elevator stops on the second floor if the second-floor button is pressed. Model checking is not a testing method, but sometimes used as an alternative to one.

Randomized testing is a black box testing method where operations and parameters are randomly generated. Random testing does not attempt to exclude already visited states or reach states. With random testing biased results are avoided [19].

Model-based testing is a form of testing where models are used to represent desired behaviour of a system [33]. Model-based testing comes from a model that describes how the program being tested will function. The model is then used to create automatic test cases.

Automated testing software is a general software that automates testing.

Automated testing framework is an umbrella term for frameworks for automated testing; for example, the Test and Operation Services Framework mentioned in study S15 is an example of an automated testing framework [22].

In integration testing all modules are combined to form a working program. Interaction between modules is emphasized [34]. Integration testing evaluates the compliance of a component or system with specified requirements. Integration testing is used together with unit and system testing, usually after unit and before system testing.

Test minimization is techniques to minimize cost in terms of execution time and resources [35]. For example, removing redundant and obsolete tests is a form of test minimization. If a requirement is satisfied by one of the tests, then that test is classified as essential and is kept [35].

Script-based testing includes scripts that automate the testing process. It is a set of instructions that will be performed during its runtime [11]. For example, a short program written in Java that tests a function is a test script.

System testing includes tests made on the systems level. System testing is used to test the full product. Since the full product is being tested, everything from user experience to how the tested system is interacting with peripherals are tested. System testing belongs to the black box testing category.

Software-in-the-loop testing is a method of testing and validating code in a simulated environment [17]. SIL testing is usually done in tandem with Hardware-in-the-loop testing, HIL is usually done later in the development while SIL is done in the early stages, since it is cheaper and simpler to perform compared to HIL.

Constraint solving is the process of finding a solution through constraints that hold conditions that the variables must satisfy [18]. Constraint solving is a technique used in constraint-based testing.

Randomized differential testing works by comparing the behaviour of a system to another implementation that is similar [10]. Differential testing works by providing the same input to a series of similar application. In the study that uses differential testing, a fault was injected into the hardware simulation layer, then a POSIX operation was chosen randomly. The operation is applied to both tested and reference system, return values and error codes are comparted as well as file systems and, lastly, invariants are checked [10]

Test runner design pattern is a test script that was implemented for testing in study S9 [16]. This script was used for stress testing the flight software for the Van Allen Probes mission.

Ideal simulation is a simulation using a minimum set of blocks to build a control loop; some blocks are not activated [17]. The simulation allows an easy switch between different approaches depending on the test run; it provides a simple framework for a first proof of concept. [17]

Directed testing is a combination of constraint solving and random testing to explore all paths through a program, first randomly choosing an initial path and then repeatedly finding input to finish the program [19].

Hierarchical test tree design pattern is a test scripting architecture used in study S9 [16]. In the S9 study, it is described as “effective at streamlining the process of integrating test scripts developed by separate teams into a single, automated regression test” [16]. An example of how it works is presented in the study. A tree with four nodes that is three nodes deep is present. We have a parent node and three child nodes. The layout is node A that leads to node B that branches out to nodes C and D. Node B runs a child script that is executed from left to right from node C to D. After C and D have been run, node B will execute any local tests in node B. [16]

Realistic simulation is a more detailed simulation than ideal simulation [17]. In study S10, its Simulink model "considers a high-fidelity space environment, sensors and actuator models" [17]. Compared to ideal simulation, realistic simulation may be considerably slower but it provides a more detailed simulation environment [17].

Monitoring and learning is a program where its execution is compared to a specification to see if it violates any desired properties. Another aspect is to learn the behaviour of the system [19]. "In monitoring, a program execution is compared with a specification, to see if it violates any desired properties. Monitoring may be indifferent as to how the execution is produced (and thus combined with one of the above testing techniques). An additional application of monitoring is to learn the behaviour of a system, rather than enforce desired behaviour. In this case a model (typically some type of automaton) is generalized from several executions of the system that exhibit desired behaviour." [19] This is not a testing method but is sometimes used instead of one.

Smoke testing is the initial testing process where the software is checked for its test-ready status [25]. In S18, smoke testing is performed by test engineers following the release of every new version. Issues are found early in the testing phase. S18 seeks to automate the manual smoke testing into an automated version [25].

5.1.2 Result analysis of RQ2

This chapter covers Research question 2. "What is the purpose of the test automation covered in the study?"

In short, most of the papers have a common goal: automate the testing to save time, since manual testing of software and hardware is time-consuming and tedious work that in most cases can be automated. There are, however, other purposes stated for test automation that were used in some of the papers. These features are varied and range from simulation automation to generation of data.

In figure 5, we can see the proportion of the stated testing purposes in the sampled studies.

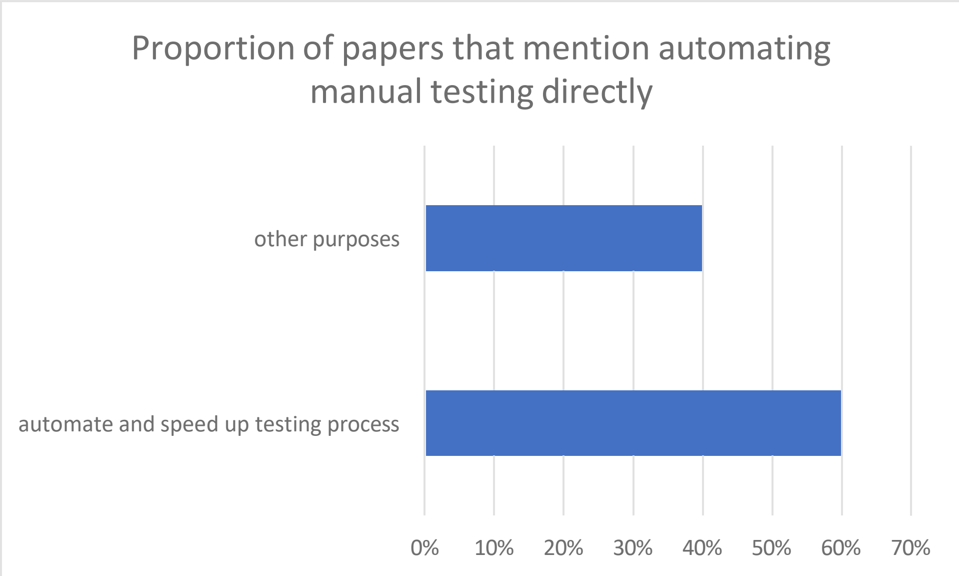

Figure 5: Proportion of directly stating automation

In Table 7, we can see the spread of the studies on what they cover.
The studies that cover test automation expressly to speed up the testing process (S1, S2, S8, S9, S12, S13, S14, S15, S17, S18, S19, S20) consist of twelve studies, while the other category (S3, S4, S5, S6, S7, S10, S11, S6) includes eight studies. Then we can also find 6 studies that have more than one purpose (S1, S2, S12, S13, S14, S17). All those studies have one thing in common, which is they all also belong in the group expressing the desire to speed up the testing process.

Table 7: Table showing which studies contain direct reference to test automation.

Intent	Study IDs	Count
Speeding up	S1, S2, S8, S9, S12, S13, S14, S15, S17, S18, S19, S20	12
other	S3, S4, S5, S6, S7, S10, S11, S6	8

More than one purpose	S1, S2, S12, S13, S14, S17	6

In table 8, we will go through the other stated purposes of test automation.

Table 8: The other purposes for test automation in the studies.

purpose	Study IDs	Count
automation of design	S1	1
automation of simulation	S1	1
generate data automatically	S1	1
automate collection and processing of data	S1	1
automation of analysis	S1	1
automation of indicators	S1	1
Randomized differential testing	S2	1
testing of the controller	S3	1
enhancing and enlarging the flight test knowledge base	S4	1
testing the PSE in real time	S5	1
finding unexpected failures in the flight software	S6	1
development of a rapid prototyping flight research facility	S7	1
evaluating algorithms	S10	1
minimising faults in the file system	S11, S12	2
testing fidelity in flight sims	S13	1

evaluation of functionality	S14	1
generate detailed test reports	S14	1
study and analyse the robustness of the software	S16	1
AI tester	S17	

Table 8 shows that most of the purposes are unique to their respective studies, however, there is one purpose that does appear in two studies: the purpose of minimizing faults in the file system. These two occurrences are from two studies that cover the same project but with different focuses (S11, S12).

The data implies that the common goal for most test automation cases is to spend less time doing manual testing. This can be alleviated by making different solutions that can easily and without fault automate the testing process and generate readable reports where the tester can check the testing results and easily fix any found bugs or faults or send them to the ones responsible for the software or hardware in question.

5.1.3 Result analysis of RQ3

This chapter covers research question 3, "What is the level and scope of implementation of test automation for flight simulators?"

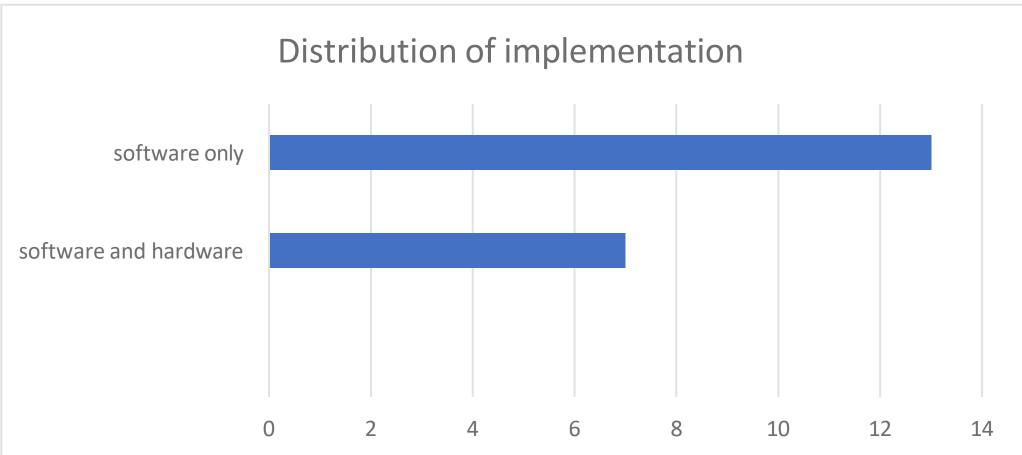

Figure 6: Distribution of implementation

In figure 6, we can see the implementation distribution. Twenty studies were included and out of those seven contained both software and hardware implementation.
The biggest group was the one containing software only implementation which was used in thirteen studies.

It can be concluded that most studies prefer software-based solutions over a combination of software and hardware implementation. The difference is not big, but it tells us that the cheaper alternative is more popular, and most testing can be done on the software level. In the end, it depends on the purpose and scope of the project involved in the studies, whether we can get a working system with only software level testing or whether we also have to include hardware level testing. Some of the studies involved testing on satellites or drones and there the hardware implementation (S3, S10, S15, S19 and S20) and a smaller subset that focused on other hardware (S2, S7).

5.1.4 Result analysis of RQ4

This section covers the analysis of research question four, "How are existing test automation evaluated?"

Most papers did not directly mention evaluation methods, it depends on what the papers covered. Some papers covered how tests were evaluated, while others covered how the whole project in the study was evaluated. Some did not really mention any of the above mentioned. Table 9 shows what the different studies focused on. The studies have been grouped together into nine different groups depending on how the evaluation methods were described in the papers.

The biggest group is the group that focuses on testing the automation approaches. This group consists of 12 studies. This is a diverse group of methods, but they all have one thing in common: tests were run to evaluate performance or find errors. The second biggest group is the comparing test data group. This group consists of six studies where the evaluation method was to compare testing data to simulation data.

There were also some studies that had unique evaluation methods or were not defined at all. These can be seen below as well and a short description of them will follow.

Unclear evaluation method, no direct evaluation was defined, two studies were grouped here.

One study had an empirical study done to evaluate the effectiveness of the automation approach.

Feedback, evaluation method based on feedback to the designer, one study had this as the sole evaluation method.

Time taken; one study had a clearly defined goal of time taken to complete as an evaluation method.

Table 9: Evaluation methods

Study ID	Focus	Study ID	Focus
S4, S6, S9, S13, S19, S20	Comparing test data	S2, S6, S7, S8, S9, S10, S11, S12, S14, S15, S16, S17	Run tests
S1, S3	Unclear evaluation method	S5	Feedback
S8	Study done	S18	Time taken

In figure 7, we can see a visualization of the evaluation groups. As was mentioned earlier, the biggest group is just over half of the studies and the second group consists of 26% of the studies. The rest of the studies remain at 9% and 4-5 %.

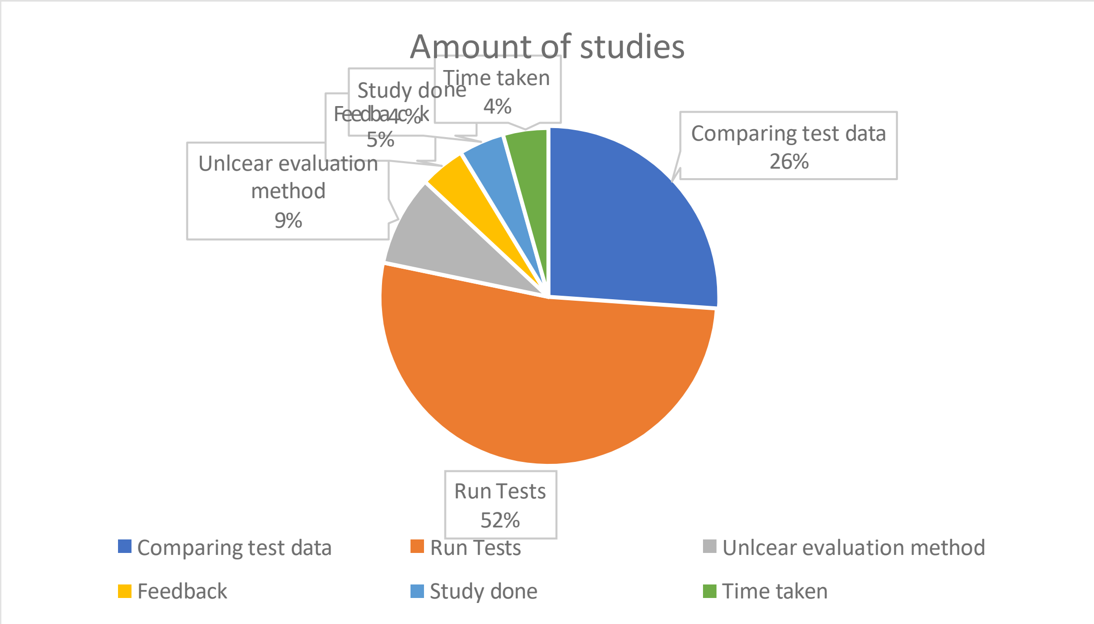

Figure 7: Visualization of groups

5.1.5 Result analysis of RQ5

This section will cover the analysis of Research question 5. What are the publication trends for studies covering Test automation in flight simulators?

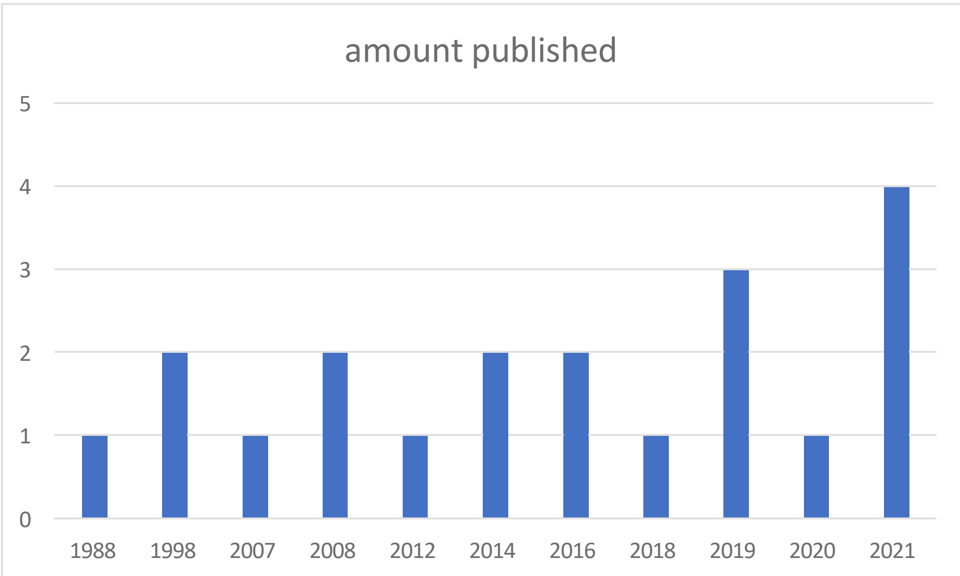

Figure 8: Amount published per year

Figure 8 shows the publication years and how many papers were published in that year. The first paper was published in 1988, then it took ten years until two more were published. In 2007, a paper was published, and after that, one or two papers have been published every other year until the late 2010s and early 2020s where a rise in publications occurred. It can be concluded that the field has progressed especially in the last fifteen years.

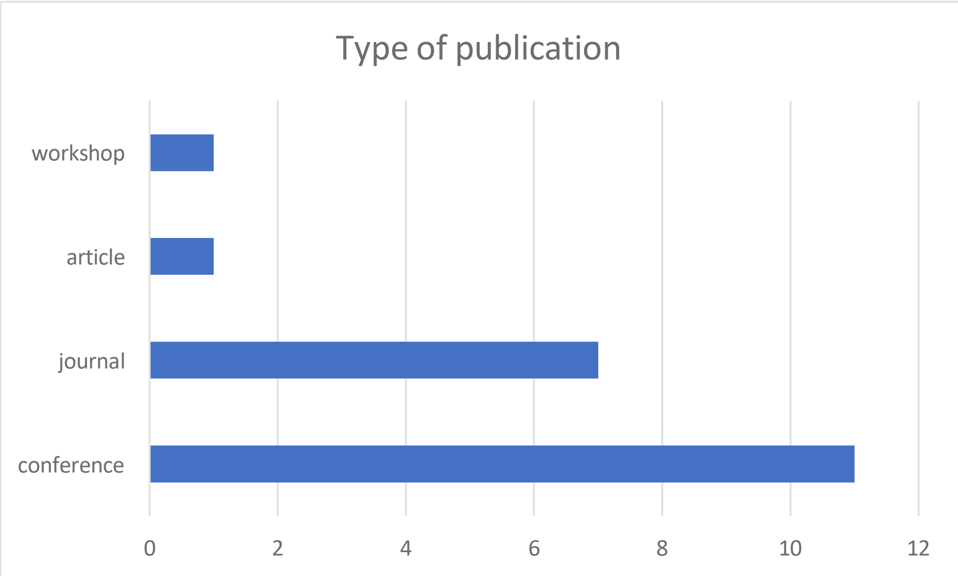

Figure 9: Type of publication venues

Figure 9 shows that most of the studies, or 11, are from conferences. Seven studies were published in journals, while articles and workshops had one each published.

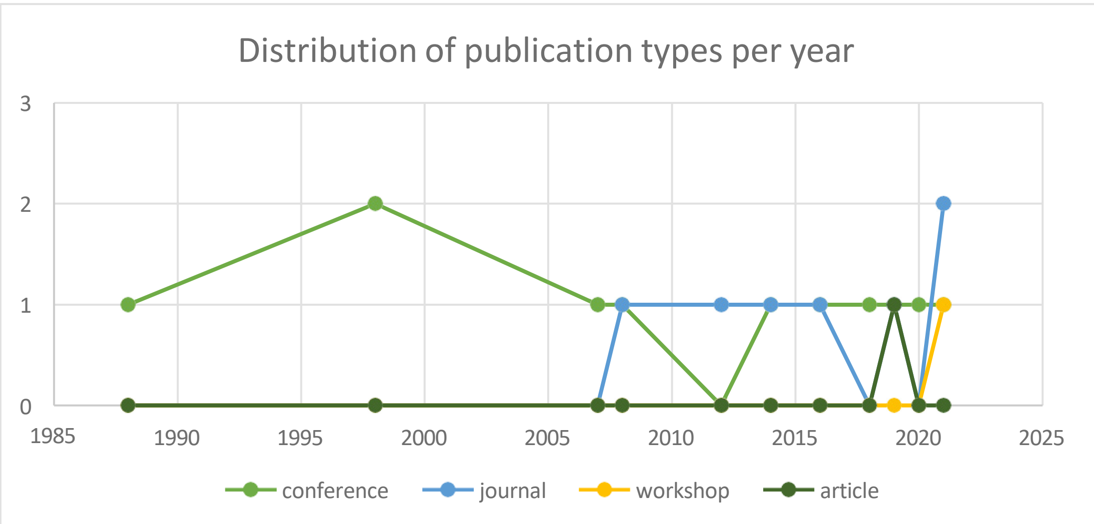

Figure 10: Distribution of publication types per year

Figure 10 shows when the different publication types were published. Conference proceedings are the publication type that has been used over most years, from 1988 to

2021, while journals come second. The first journal was published in 2008 and then has been steadily published up to 2021. The workshop and article publications were published once, the workshop publication in 2021 and the article in 2019. A conclusion can be drawn that conference proceedings and journals are the most common way of publishing research papers in the test automation field and only very recently other venues have started to emerge in this field.

6. Discussion

This systematic mapping study was conducted for the purpose of mapping the field of test automation in flight simulators. In order to describe and interpret the findings, this chapter will discuss the findings in the previous chapter.

Regarding test automation approaches (RQ1), it seems no approach exists that does it all, but there are a couple of approaches that are used more than others. These approaches are general testing approaches that are widely used for testing purposes; for example, unit testing is a widely used testing method. There are five approaches that are used the most. These are used in three studies each. The second-most used approaches were used twice in four different studies and the remaining approaches were used once. This leaves the most popular approaches with 18.52 percent of the 27 approaches that were used in the 20 studies of this work. The second-most used approaches include about 14.81 percent and the remaining 18 single use approaches 66.67 percent. We can conclude that no catch-all solution exists regarding test automation for flight simulators, but some approaches are slightly more popular than others. It can be argued that the sample size in this study is rather low, but it seems most papers are not focused specifically on test automation; it is most of the time a topic that is not mentioned.

Regarding the purpose of test automation (RQ2), we can conclude that the most common goal is to automate and speed up the testing process, with about 60 percent of the studies having that as the goal for their test automation. The remaining 40 percent is categorized as other purposes. This category contains a wide variety of differing purposes or hard-to-define purposes. It is safe to say that the majority aims for the same goal as regards test automation. A common theme among many of the studies was that manual testing is slow and expensive, especially concerning hours used on it, hours that could be used for more pressing matters. This presses home the need for test automation, since it releases people from tedious and time-consuming tasks. It is obviously not always feasible to automate all testing, but in the majority of cases it can and should be done to save time and minimize user errors.

Regarding implementation (RQ3), we simplified the studies into what sort of implementation they were using, software or software and hardware, for their test automation. Thirteen studies are using a software implementation only, while seven are using a combination of hardware and software. This is a rather even split with some favourability leaning towards software only implementation. This makes sense, since the majority of projects regarding flight simulators would not need any sort of hardware implementation but, in some cases, it seems to be necessary. Another reason to focus more on the software level implementation is that it would be cheaper to implement compared to making tests for hardware implementations.

Regarding evaluation methods (RQ4), we can conclude that this was a difficult question to find answers to, since almost all studies mainly focused on something else than test automation, so the evaluation methods that were listed were most of the time not related to test automation evaluation. In the end, some evaluation methods for the topic of this study were able to be found in the studies.

We ended up with six categories for evaluation methods, one being clearly bigger than the others. This method was a general group for evaluation methods based on running tests; 52 percent of the studies were in this group. The second biggest of the groups was a group that consisted of comparing data. This group included 26 percent of the studies. The remaining studies all included one or two studies.

Regarding publication trends (RQ5), the majority of the primary studies, 11 studies were published at conferences. The second biggest contributor is journals at 7 published studies and workshops and articles at one each. More evenly spread was year of publication, with some years being more productive than others, although a growing trend can be noticed. Since 2008, publications per year have increased and years with more than one published paper have become more common, culminating in

2021 with four published papers. The first paper was published in 1988 and it took ten years for the next papers to be published in 1998. This and the fact that there have been almost yearly publications since 2007 points to the field growing in popularity but also that flight simulation and aviation in general are becoming more accessible, especially with unmanned aircraft such as drones. This has also increased the availability for flight software such as simulators.

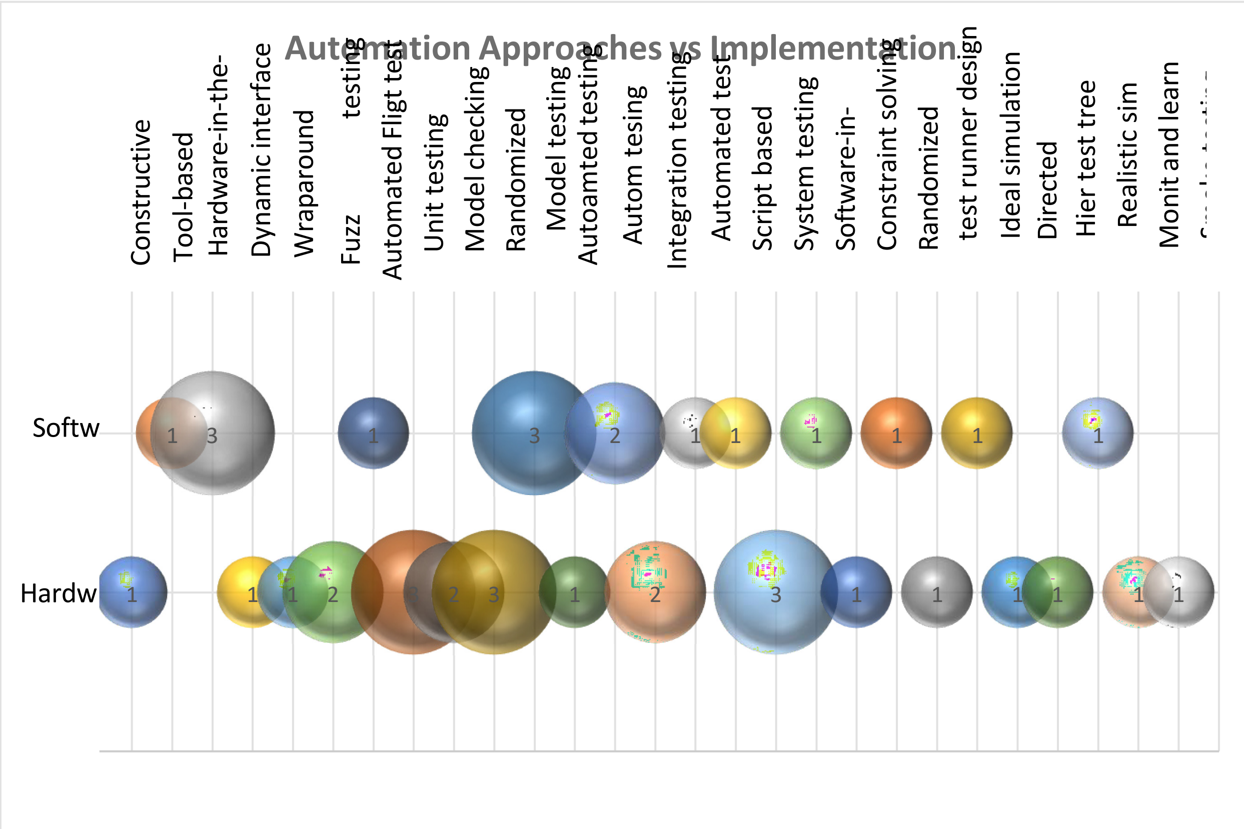

Figure 11: Approaches and implementation groups

In Figure 11, the amount of test automation approaches in relation to the implementation of the test automation approaches is visualized.

The horizontal line is the test automation approaches. It can be noted that the distribution is relatively even between the test automation approaches. Gaps can be seen between the approaches; this clearly gives an overview of which approaches where used for what implementation. For example, label 5 which stands for Wraparound automated testbed is a Software and Hardware implementation approach. This approach is used in study S5 where both software and hardware are used to conduct automated testing [13].

7. Threats to validity

The first major threat to validity of the results in this paper is related to the coverage of the literature used in the study. To diminish the chances of this being a threat a search strategy based on the guidelines for systematic mapping studies and systematic literature reviews [7] [6] has been followed. The search terms were taken from the RQs and were used in four major digital research databases.

The second threat would be the selection of the primary studies. Out of 174 papers found in the initial search, only 20 were used in the final paper. There is a risk that papers that covered the topic was missed, and to mitigate this risk extensive filtering was performed. Title and abstract screening removed 68 studies from the pool. After this, 106 studies remained and were screened again, and 53 studies were removed. The remaining studies were screened fully, and 33 studies were deemed not suitable. The remaining studies were now screened, and snowballing was performed, and the final 20 studies were ultimately used in this review.

The third threat would be the extracted data. Is the data extracted for the review classified correctly? Does it belong in the topic of test automation? This threat was diminished by researching the data extracted and validating that it is indeed fitting to the scope of the study.

8. Conclusions

In this paper, we presented a Systematic Mapping Study (SMS) of test automation approaches for flight simulators. Systematic Literature Review (SLR) guidelines were applied to design a search strategy. The initial search returned 174 studies from four major digital libraries. After removal of duplicates and quality filtering with the help of the inclusion/exclusion criteria as well as snowballing, 20 primary studies were selected. These 20 studies were used as source material for the conducting of this SMS. The objective of the SMS was to provide a comprehensive, unbiased overview of the state of test automation approaches for flight simulators. To do this we chose to pose five research questions (RQ) that would help us in this endeavour. The following research questions to be answered were: What existing approaches for test automation are there? What is the purpose of said approaches? What is the implementation of said approaches? How are they evaluated and, finally, what are the publication trends regarding research in the field?

The results presents a picture of the current situation.
RQ1 shows that there is not one defining approach for this field rather that the test automation approach is defined on a case-by-case basis. There were some methods that are more popular than others, but they are widely used in general.
RQ2 shows that the purpose in most cases is to save time and speed up the process. This seems to make sense as otherwise why would one automate the testing? The other purposes were usually unique to the studies they were used in and did not give a universal picture on the matter.

RQ3 focused on the implementation of the test automation, and it shows that the studies focus on two areas, software implementation and a mix between software and hardware implementation. The software implementation is the more popular solution. This is since most of the studies did not have any sort of hardware other than the computers the software is used.

RQ4 focused on evaluation methods. Here two groups in the six-group classification that was used stand out with 12 studies focusing on running tests to evaluate. The second biggest evaluation methods are comparing test data to sample data.

RQ5 focused on the distribution of the papers. Here we look at when and where the studies have been published. The first study was published in 1988 and the last one in 2021. We can see a clear trend that after 2007 the publication rate has grown, and studies have been published almost yearly, with a few exceptions, 2021 being the year with the most publications. With the growing trend in mind, the field will most likely grow and we will see more studies done on test automation approaches for flight simulators.

9. Sammanfattning

Den här avhandlingen titel är *En systematisk kartläggningsstudie över testautomatiserings metoder för flygsimulatorer.*

9.1 Introduktion

Mjukvarutestning är en tidsödande process som kan undvikas genom automatisering. Större projekt för framtagning av mjukvara för flygsimulatorer kan både ta upp mycket arbetstid och arbetskraft. Mjukvarutestningen kan därför gynnas av standardisering och automatisering för att höja mjukvarukvaliteten.

För att få en överblick av var forskningen står i dag och vilka metoder som används för testning av mjukvara, har en systematisk kartläggningsstudie utförts. De frågor som ska besvaras i den här avhandlingen är:

- Vilka metoder används för testautomatisering för flygsimulatorer?
- Hur används de testautomatiseringsmetoder som är i bruk?
- På vilken nivå implementeras testautomatiseringsmetoderna?
- Hur bedöms de testautomatiseringsmetoder som är i bruk?
- Hur ser publiceringstrenderna ut inom forskningen?

Den här studien har genomförts genom en systematisk kartläggning av testautomatiseringsmetoder. Tidigare forskning som är relevant för den här avhandlingen behandlas i kapitel 2. Därefter i kapitel 3 presenteras systematisk kartläggning som forskningsmetod. Kapitel 4 redovisar för hur den här studien har genomförts. I kapitel 5 presenteras studiens resultat, medan kapitel 6 består av en diskussion av resultat. I kapitel 7 förs en kritisk diskussion av den här studiens trovärdighet. Avslutningsvis sammanfattas studiens slutsatser i kapitel 8.

9.2 Forskningsmetod

Forskningsmetoden i den här avhandlingen är en systematisk kartläggning. Systematisk kartläggning som forskningsmetod kräver att det finns ett behov av forskning inom området, att bestämda forskningsfrågor besvaras, och att forskningen ger ett definierat slutresultat.

För att de här kriterierna ska uppnås har en forskningsplan gjorts, som har följts under forskningsarbetet.

Den här studien har genomförts i tre olika faser: sökning och urval, informationsinsamling, samt analys.

I den första fasen, sökning och urval av primärkällor, tar man i bruk de söksträngar som blivit framtagna under planeringsskedet, och framgår i forskningsplanen. Söksträngarna används i de valda digitala biblioteken. Härefter slås dubbletter av sökresultatet samman och irrelevanta fynd tas bort. Urvalskriterier tillämpas för att hitta primärkällor, sedan används snöbollsmetoden för att komplettera listan över dessa. När de här stegen är avslutade har man en lista över primärkällor som är redo för informationsinsamling.

Informationsinsamlingsfasen består av att primärkällorna granskas. I det här skedet tas relevant information till vara för att användas i nästa fas. Slutligen analyseras den insamlade informationen och sammanställs för att kunna besvara forskningsfrågorna. Det här skedet utgör analysfasen.

9.3 Den systematiska kartläggningen

Undersökningen utfördes genom att välja ut de primärkällor som bäst lämpades för att besvara forskningsfrågorna. Primärkällorna, som utgör materialet i den här studien, hittades via de elektroniska biblioteken IEEE Xplore, ACM, ScienceDirect och SpringerLink. Sökningen bidrog till 174 artiklar.

Urvalsprocessen utfördes sedan i fyra faser. Första fasen bestod av titel och abstraktgenomgång, där 133 av 174 artiklar accepterades. Den andra fasen bestod av textgenomgång av dessa 133 artiklar, där endast 8 av dem godkändes.
Den tredje fasen bestod av att genom snöbollsmetoden utöka antalet relevanta artiklar. Den här metoden består av att man systematisk går igenom de källor som huvudartikeln hänvisar till, eller att man söker artiklar som har hänvisat till den valda

huvudartikeln. Med hjälp av snöbollsmetoden ökade antalet artiklar, och resulterade i 16 relevanta artiklar.

För att ytterligare utöka resultatet så utfördes en manuell sökning med hjälp av Google Scholar där ytterligare fyra artiklar hittades. Slutresultatet blev 20 artiklar som utgör primärkällorna i den här studien.

Den fjärde fasen bestod av bedömning och datainsamling av de 20 utvalda primärkällorna. Datainsamlingen skedde med hjälp av ett datainsamlingsformulär som tillämpades på de artiklar som valdes inom ramen för kartläggningen.

9.4 Resultat

Den här studiens fokus har varit att besvara den första forskningsfrågan: ”Vilka metoder används för testautomatisering för flygsimulatorer?’’. Resultat i den här studien påvisar är att processen för testautomatisering av flygsimulatorer utgör ett brett område där många metoder kan tillämpas. I den här studien har 27 olika testautomatiseringsmetoder identifierats.

För att besvara den andra forskningsfrågan, ’’Hur används de testautomatiseringsmetoder som är i bruk?’’, påvisar resultatet att av dessa 27 var det fem metoder som användes i tre unika projekt. Utöver det så fanns det fyra metoder som användes i två projekt var, de övriga metoderna användes enbart i enskilda projekt. De mest använda metoderna är välkända testningsmetoder som dagligen används inom programmering. Testautomatisering används främst för att snabba upp och automatisera testningsprocessen. En röd tråd identifierades i flera artiklar där manuelltestning av mjukvara anses vara tidskrävande och arbetsdrygt, vilket automatisering kan lösa. De andra användningsändamålen var unika för sina projekt så de grupperades som en egen kategori under namnet övriga ändamål.

För att besvara forskningsfrågan om vilken nivå testautomatiseringsmetoderna implementeras på har man i den här studien valt att fokusera på mjukvara och hårdvara. Resultatet tyder på att en klar majoritet av testautomatiseringsmetoderna, som framgår i materialet, är implementerade på endast mjukvarunivå. Av de 20 primärkällor som använts framgår det att i 13 av dem är testautomatiseringsmetoderna implementerade på mjukvarunivå och 7 på hårdvarunivå.

Gällande den fjärde forskningsfrågan, '' Hur bedöms de testautomatiseringsmetoder som är i bruk?'', har primärkällorna grupperats i sex kategorier. Av dessa kategorier sticker två ut på basen av mängden artiklar. I den största gruppen, där tolv artiklar kan placeras, bedöms metoderna genom olika tester. Den näststörsta kategorin fokuserar på att jämföra testdata med modelldata. I den här kategorin ingår sex artiklar. De kvarstående fyra kategorierna består av en eller två studier var.

Den femte, och sista, forskningsfrågan fokuserar på publiceringsdata gällande tidigare forskning, när och var har artiklarna publicerats. Den äldsta artikeln är från 1988 och den färskaste från 2021. En klar trend kan däremot ses i publiceringsåren. Efter 2007 har artikelpubliceringen bara ökat. Före 2007 har endast tre artiklar publicerats, varav två artiklar 1998 och en 1988. Det mest produktiva året var 2021 med 4 artiklar publicerade. Det här visar att intresset för det här ämnet växer. Vilket kan bero på att specifikt drönare är mera tillgängliga än tidigare.

9.5 Avslut

I den här avhandlingen presenteras en systematisk kartläggningsstudie av testautomatiseringsmetoder för flygsimulatorer. Syftet med kartläggningen var att ge en heltäckande, opartisk översikt över tillståndet för testautomatiseringsmetoder för flygsimulatorer.

För att göra detta ställdes fem forskningsfrågor: *Vilka metoder används för testautomatisering för flygsimulatorer?*; *Hur används de testautomatiseringsmetoder som är i bruk?; På vilken nivå implementeras testautomatiseringsmetoderna?; Hur bedöms de testautomatiseringsmetoder som är i bruk?;* samt *hur ser publiceringstrenderna ut inom forskningen?*

Resultaten i den här studien ger en bild av hur situationen för ämnet ser ut förtillfället. Som svar för fråga ett kan man konstatera att det finns flera metoder att utföra testautomatisering och ingen av dem är mera populär än någon annan. Däremot så används vissa metoder lite oftare än andra.

Syftet för att använda testautomatisering handlar i de flesta fall om att spara tid och påskynda processen. De metoder som endast använts i unika projekt har inte kunnat ge en generell bild av varför man valt att använda testautomatisering.

Fråga tre fokuserade på implementeringen av testautomatiseringen, och resultaten visar på att artiklarna fokuserar på två områden, mjukvaruimplementering och en blandning mellan mjukvaru- och hårdvaruimplementering. Mjukvaruimplementeringen är den populärare lösningen eftersom de flesta av studierna inte använde sig av någon extra hårdvara.
Gällande utvärderingsmetoder utmärker sig två grupper i den sexgruppsklassificering som användes då tolv artiklar använde sig av tester för att utvärdera testautomatiseringsmetoderna. De näst största utvärderingsmetoderna är att jämföra testdata med modelldata.
Trenden för publicering av tidigare forskning inom ämnesområdet visar att den första studien publicerades 1988 och den sista studien 2021. Vi kan se en tydlig trend efter 2007 då publiceringstakten ökat och studier har publicerats nästan årligen med några få undantag. Året med flest publikationer visade sig vara.

Med den växande trenden i åtanke, kan man dra slutsatsen att intresset för forskning inom området med största sannolikhet kommer fortsätta växa. Vilket innebär att man i framtiden kommer att se fler studier göras om testautomatiseringsmetoder för flygsimulatorer.

References

[1] P. Raulamo-Jurvanen, M. Mäntylä and V. Garousi, “Grey literature review, tool selection, software test automation, test automation tool,” in *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, Karlskrona, Sweden, 2017.

[2] A. Afzal, D. S. Katz, C. L. Goues and C. S. Timperley, “Simulation for Robotics Test Automation: Developer Perspectives,” in *2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST)*, 2021.

[3] V. Garousi, S. Tasli, O. Sertel, M. Tokgoz, K. Herkiloglu, H. F. E. Arkin and O. Bilir, “Automated Testing of Simulation Software in the Aviation Industry: An Experience Report,” *IEEE Software,* vol. 36, p. 63–75, July 2019.

[4] V. Garousi and M. V. Mäntylä, “When and what to automate in software testing? A multi-vocal literature review,” *Information and Software Technology,* vol. 76, pp. 92-117, 2016.

[5] S. C. Lee and V. G. Duffy, “Use of Simulation Technology in Transportation Training: A Systematic Literature Review,” in *HCI International 2021 - Late Breaking Papers: HCI Applications in Health, Transport, and Industry*, 2021.

[6] K. Petersen, S. Vakkalanka and L. Kuzniarz, “Guidelines for conducting systematic mapping studies in software engineering: An update,” *Information and Software Technology,* vol. 64, pp. 1-18, 2015.

[7] B. Kitchenham and S. M. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” vol. 2, 2007.

[8] C. Wohlin, “Guidelines for Snowballing in Systematic Literature,” in *EASE '14: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, London, 2014.

[9] J. Hodicky, D. Prochazka and J. Prochazka, “Automation in Experimentation with Constructive Simulation,” in *Modelling and Simulation for Autonomous Systems*, Springer International Publishing, 2019, p. 566–576.

[10] A. Groce, G. Holzmann and R. Joshi, “Randomized Differential Testing as a Prelude to Formal Verification,” in *29th International Conference on Software Engineering (ICSE'07)*, Minneapolis, MN, USA, 2007.

[11] A. Ptak and K. Foundy, “Real-Time Spacecraft Simulation and Hardware-in-the-Loop Testing,” in *Proceedings. Fourth IEEE Real-Time Technology and Applications Symposium (Cat. No.98TB100245)*, Denver, CO, USA, 1998.

[12] D. Carico, “Rotorcraft flight test automation-the saga continues,” in *1998 IEEE Aerospace Conference Proceedings (Cat. No.98TH8339)*, Snowmass, CO, USA, 1998.

[13] T. Parker, “Solar Probe Plus (SPP) Wrap Around Automated Testing,” in *2016 IEEE AUTOTESTCON*, Anaheim, 2016.

[14] T. Gutierrez, A. Bergel, C. E. Gonzalez, C. J. Rojas and M. A. Diaz, “Systematic Fuzz Testing Techniques on a Nanosatellite Flight Software for Agile Mission Development,” *IEEE Access,* vol. 9, p. 114008–114021, 2021.

[15] M. Hewett, D. Tartt, E. Duke, R. Brumbaugh, R. Antoniewicz and A. Agarwal, “THE DEVELOPMENT OF AN AUTOMATED FLIGHT TEST MANAGEMENT SYSTEM FOR FLIGHT TEST PLANNING AND MONITORING,” in *Proceedings of the 1st International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Tullahoma, 1988.

[16] J. Finnigan, “A Scripting Framework for Automated Flight SW Testing: Van Allen Probes Lessons Learned,” in *2014 IEEE Aerospace Conference*, Big Sky, MT, USA, 2014.

[17] J. Kiesbye, D. Messmann, M. Preisinger, G. Reina, D. Nagy, F. Schummer, M. Mostad, T. Kale and M. Langer, “Hardware-In-The-Loop and Software-In-The-Loop Testing of the MOVE-II CubeSat,” *Aerospace,* vol. 6, p. 130, 1 December 2019.

[18] A. Groce, G. Holzmann, R. Joshi and R.-G. Xu, “Putting Flight Software Through the Paces with Testing, Model Checking, and Constraint-Solving,” in *International Workshop on Constraints in Formal Verification*, 2012.

[19] A. Groce, K. Havelund, G. Holzmann, R. Joshi and R.-G. Xu, “Establishing flight software reliability: testing, model checking, constraint-solving, monitoring and learning,” *Annals of Mathematics and Artificial Intelligence,* vol. 70, p. 315–349, 18 March 2014.

[20] C. Wang, J. Han, G. Li and H. Jiang, “Flight Simulator Fidelity Evaluation Automated Test System Analysis,” in *2008 International Workshop on Education Technology and Training & 2008 International Workshop on Geoscience and Remote Sensing*, Shanghai, China, 2008.

[21] Y. Zaidi, N. Fitz-Coy and R. Van Zyl, “RAPID, AUTOMATED TEST, VERIFICATION AND VALIDATION FOR CUBESATS,” in *67th International Astronautical Congress (IAC)*, Guadalajara, Mexico, 2016.

[22] S. Busch, C. Horch and F. Schäfer, “A Flexible Test and Operations Automation Framework for Agile Satellite Engineering Workshop on Simulation and EGSE for Space Programmes (SESP) 30,” 2021.

[23] T. Gutierrez, A. Bergel, C. E. Gonzalez, C. J. Rojas and M. A. Diaz, “Toward Applying Fuzz Testing Techniques on the SUCHAI Nanosatellites Flight Software,” in *2020 IEEE Congreso Bienal de Argentina (ARGENCON)*, 2020.

[24] H. Sartaj, “Automated Approach for System-level Testing of Unmanned Aerial Systems,” in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2021.

[25] A. Yildirim, E. Berker and M. Kayakesen, “SYSTEM LEVEL TEST AUTOMATION IN UAV DEVELOPMENT,” in *2018 IEEE AUTOTESTCON*, National Harbor, 2018.

[26] C. Wang, J. He, G. Li and J. Han, "An Automated Test System for Flight Simulator Fidelity Evaluation," *JOURNAL OF COMPUTERS,* vol. 4, no. 11, 2009.

[27] X. Dai, C. Ke, Q. Quan and K.-Y. Cai, "RFlySim: Automatic test platform for UAV autopilot systems with FPGA-based hardware-in-the-loop simulations," *Aerospace Science and Technology,* vol. 114, 20 July 2021.

[28] P. Nagarani and R. VenkataRamanaChary, "A tool based approach for automation of GUI applications," in *2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12)*, Coimbatore, India, 2012.

[29] M. Bacic, "On hardware-in-the-loop simulation," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, 2005.

[30] J. Nelson, "Dynamic Interface Testing of Unmanned Aircraft Systems," in *STO-MP-SCI-269*, Saint Inigoes, 2015.

[31] P. Runeson, "A survey of unit testing practices," *IEEE Software ,* vol. 23, no. 4, pp. 22-29, 2006.

[32] E. M. Clarke, "Model checking," in *Foundations of Software Technology and Theoretical Computer Science*, Berlin, 1997.

[33] S. Dalal, J. A, K. N, L. J.M., L. C.M., P. G.C. and H. B.M, "Model-based testing in practice," in *Proceedings of the 1999 International Conference on Software Engineering (IEEE Cat. No.99CB37002)*, Los Angeles, 1999.

[34] H. Leung and L. White, "A study of integration testing and software regression at the integration level," in *Proceedings. Conference on Software Maintenance 1990*, San Diego, 1990.

[35] R. Sing and M. Santosh, " Test Case Minimization Techniques : A Review," *International Journal of Engineering Research & Technology (IJERT),* vol. 2, no. 12, 2013.

[36] D. D. H. R. R. Hill, "The art and science of live, virtual, and constructive simulation for test and analysis," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology,* vol. 11, no. 2, pp. 77-89, 2013.

[37] V. Garousi and M. V. Mäntylä, “A systematic literature review of literature reviews in software testing,” *Information and Software Technology,* vol. 80, pp. 195-216, 2016,.