TU WIEN Informatics

Enhancing Fault Diagnosis in Safety-Critical Cyber-Physical Systems

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktorin der Technischen Wissenschaften

by

Drishti Yadav, M.Tech.
Registration Number 12035748

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dr. Ezio Bartocci

The dissertation has been reviewed by:

Prof. Domenico Bianculli

Prof. Giulia Pedrielli

Vienna, 30th September, 2024

Drishti Yadav

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Declaration of Authorship

Drishti Yadav, M.Tech.

I hereby declare that I have written this Doctoral Thesis independently, that I have completely specified the utilized sources and resources and that I have definitely marked all parts of the work - including tables, maps and figures - which belong to other works or to the internet, literally or extracted, by referencing the source as borrowed.

Vienna, 30. September 2024

Drishti Yadav

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

To my parents
Usha Yadav *and* ***Alakh Ram Yadav***

यन्मातापितरौ वृत्तं तनये कुरुतः सदा ।
न सुप्रतिकरं तत्तु मात्रा पित्रा च यत्कृतम् ॥

- श्रीमद्वाल्मीकिरामायणे अयोध्याकाण्डे
एकादशोत्तरशततमस्सर्गः (९ श्लोक)

“*The deeds done by mother and father for their children constantly…, there is no requital to these actions performed by the parents.*”

- Shrimad Valmiki Ramayana, Ayodhya Kanda,
Chapter [Sarga] 111, Verse [Shlok] 9

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Acknowledgements

I wholeheartedly dedicate this thesis to my beloved parents, whose unwavering love and sacrifices have sculpted the person I am today. Through every trial and triumph, in joy and sorrow, your endless support has been my anchor. Your belief in me, even in the simplest of accomplishments, fills me with gratitude and humility. To my dear father, though you have transcended the mortal realm, your presence remains palpable, guiding me through life's journey. I carry your love and teachings in my heart, always. Love you Mummy, Love you Papa!

I am also deeply thankful to the Almighty for bestowing upon me the gift of life and for showering me with boundless blessings. Whenever I have sought guidance, support, or love, I have found solace in the divine presence that surrounds me. Furthermore, I extend my heartfelt appreciation to my sister, *Dr. Shilpee Yadav*, whose unconditional love, support, and encouragement have been the cornerstone of my PhD journey. Simply put, this dissertation owes its existence to the indispensable support of my parents, the divine, and my sister.

Embarking on the journey towards a doctoral thesis entails navigating a challenging path fraught with numerous obstacles. I am profoundly grateful for the invaluable guidance and mentorship provided by my advisor, *Univ.Prof. Dr. Ezio Bartocci*. His unwavering support, profound expertise, and professional acumen have been invaluable assets, guiding me through both moments of triumph and adversity. He has not only made this academic endeavor achievable but also guided the research process, imparting in me the ability to nurture my own potential. Thank you so much, *Ezio*!

I am equally indebted to my collaborators, who have served as invaluable mentors and partners in my academic journey. *Dr. Dejan Ničković*, Senior Scientist at AIT Austrian Institute of Technology, Austria, and *Prof. Leonardo Mariani*, University of Milano-Bicocca (UniMiB), Milan, Italy, have been pivotal figures in my PhD pursuit. Their never-ending support, patience, and encouragement have been the bedrock of my scholarly endeavors, propelling me forward on the path to academic excellence. I am immensely grateful for their guidance and mentorship, which have been instrumental in shaping my academic trajectory. Thank you, *Leo*! Thank you, *Dejan*! Without a grain of doubt, *Ezio*, *Leo*, and *Dejan*, the trio, have been the finest mentors I could have ever asked for.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

I am deeply honored to have undertaken a research stay at the Department of Informatics, Systems, and Communication DISCo, UniMiB, under the guidance of *Prof. Leonardo Mariani*. I also express my heartfelt gratitude to TU Wien for awarding the *KUWI* grant, which facilitated my research stay abroad. Additionally, I extend my thanks to the *Doctoral College Resilient Embedded Systems*, hereafter DC-RES (which is run jointly by the TU Wien's Faculty of Informatics and the University of Applied Sciences Technikum Wien) for funding the publication of my research work. I also want to extend my gratitude to the academic and administrative staff at DC-RES and CPS (Cyber-Physical Systems) research group at TU Wien for their continuous assistance, guidance, and cooperation during my academic journey.

I am grateful to the members of the Proficiency Evaluation Committee, my advisor *Univ.Prof. Dr. Ezio Bartocci*, *Prof. Karl M. Göschka* as the National Expert and *Ao.Univ.Prof. Andreas Steininger* as the Chairperson, for their invaluable feedback and suggestions. Furthermore, I extend my thanks to *Prof. Domenico Bianculli* and *Prof. Giulia Pedrielli* for their diligent review of my thesis and for their insightful comments and suggestions.

I am immensely grateful for the opportunity to have travelled to various countries around the world and present my research work at prestigious conferences including (1) ISSRE 2022 in Charlotte, North Carolina, USA; (2) ESEC/FSE 2022 in Singapore, and (3) ICST 2023 in Dublin, Ireland. Additionally, I had the privilege of attending the Informatics Europe Summer School on Informatics Education Research (SCHIER) 2024, supported by IBM Research Europe and hosted at the IBM Innovation Studio & Think Lab Zurich, in Rüschlikon, Switzerland. The summer school offered unique and exciting moments, including visits to IBM's Quantum Computing Lab and Nanotechnology Center—absolutely fascinating! Furthermore, I truly appreciate the chance to participate as a student volunteer at ISSTA/ECOOP 2024 in Vienna and share my research at the Doctoral Symposium, where the insightful feedback I received was highly valuable. All these experiences not only allowed me to expand my network but also provided valuable insights and perspectives. Additionally, the chance to explore new places and cultures during these trips added an extra dimension of enjoyment to the entire experience.

My tenure at TU Wien has been enriched by the camaraderie and support of my exceptional research group and friends. To *Lilly*, *Mahum*, *Elisa*, *Eleonora*, *Michele*, *Monika*, *Luigi*, *Francesco*, *Ignacio*, *Axel*, *Andreas*, *David*, *Ennio*, *Stefan* and *Emad*, I extend my heartfelt gratitude for making my stay at TU Wien feel like home. Your steadfast support and friendship have brought immense strength, comfort, and joy. I extend my heartfelt gratitude for the cherished memories of our '*Pasta Thursdays*' and '*Cake Fridays*', our delightful trips, shared lunches, Christmas Parties, the TU Ball 2024, and the joyous occasions of indulging in chocolates, snacks, or cakes without any reason. Our conversations, spanning topics both profound and mundane, have enriched my life beyond measure. Thank you all for these treasured moments. Also, big thanks to *Eleonora* and *Mahum* for joining me on that wild and totally unforgettable skydiving adventure!

I am also grateful to all my friends who have accompanied me on this journey, enriching my life with laughter, camaraderie, and invaluable learning experiences. Special thanks to my dear friend, *Niveditha*, for her steadfast presence during times of need, invaluable emotional support and for treating me to delightful Indian cuisines. A heartfelt thanks to her family, especially her husband *Abilash*, for their kindness and encouragement.

On top of that, the friends I have made through DC-RES have been a tremendous support. I especially want to give a big applause to *Lilly*, who is not just a great friend but also my go-to guide in the new Viennese adventure. Your help and support during those early days in Vienna were priceless, and I cannot thank you enough for it! A big shoutout to *Mahum* for her friendship, endless support and positivity. Our memorable trips and fun-filled adventures would not have been the same without her. Huge thanks to *Mahum* and *Hugo* for helping me ace the Formal Methods exam. Massive appreciation to *Luigi* for hooking me up with GitLab and walking me through the ropes. And huge thanks to *Mahyar* for the amazing chats about life and those mouthwatering Iranian sweets. Also, heartfelt appreciation to *Raghda* and *Najla* for all the laughter and great conversations we have shared together. To all the DC-RES PhD students, your support is deeply appreciated.

I would like to express my heartfelt thanks to all my relatives, family friends, uncles, and aunts who always cheer me on for even the smallest achievements. My deepest gratitude goes to my uncles *Mahesh*, *Santosh*, and *Ganesh* for their unwavering belief in me and their enduring love and support. I am also thankful to my uncle and aunt, *Pramod* and *Kiran*, for their abundant love, blessings, and encouragement. Special thanks to my aunt *Shweta*, uncle *Rajendra*, and my lovely cousin *Mahek* for their love, blessings, delightful meals, and warm hospitality during my visits to Denmark, especially during our festivals. I also want to express my gratitude to the new family I have found in Denmark – my aunt's friends: *Rashida*, *Tanya*, *Veena*, *Anju*, *Arpita*, and their families. Their love, warmth, and flavorsome Indian food made me feel like I was back home in India. Thank you all!

My deepest love and gratitude are extended to all my teachers and mentors, whose wisdom and guidance have shaped my academic journey in profound ways. Grateful acknowledgments to *Dr. R. N. Patel*, my Bachelor's mentor, and *Dr. Om Prakash Verma*, my Master's advisor, for their encouragement and guidance. To every person I have encountered along the way, I am indebted for the invaluable lessons learned, even in the simplest of interactions.

Lastly, to my dear family, whose relentless support and encouragement have been my guiding light, I express my deepest appreciation. Your love and encouragement have sustained me through every trial and tribulation. I eagerly anticipate reuniting with you all after more than four years, promising to cherish every moment spent together. Thank you for being the pillars of strength in my life!

Thank you, everyone.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

TU Bibliothek Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
WIEN Your knowledge hub The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Abstract

Cyber-Physical Systems (CPSs) present increasingly complex challenges, particularly within safety-critical domains. Ensuring the correctness of safety-critical CPSs is crucial for maintaining their viability and full operational capacity, as undetected failures can be not only costly but also life-threatening. The timely detection and diagnosis of faults are essential in the CPS development cycle, given the prohibitively high costs of rectifying faults post-hardware integration. Consequently, thorough testing of CPSs becomes paramount, underscoring the necessity for advanced fault diagnosis procedures that can accurately detect, explain, and rectify faults.

In this thesis, we develop innovative tools and methods tailored for CPSs, aimed at facilitating swift and accurate fault diagnosis. Our contributions encompass fault injection, global optimization, search-based testing for fault localization, mutation testing against formal properties, and signal feature coverage-based testing. This holistic approach offers significant advantages over existing methods, ensuring thorough testing across various scenarios.

The proposed solutions have been evaluated using publicly available benchmarks from diverse domains, demonstrating the applicability and effectiveness of our research. The findings of this thesis advance the knowledge in CPS testing, paving the way for more reliable and safe CPSs in the future.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Kurzfassung

Cyber-Physical Systems (CPSs) stellen immer komplexere Herausforderungen dar, insbesondere in sicherheitskritischen Bereichen. Die Gewährleistung der Korrektheit sicherheitskritischer CPS ist von entscheidender Bedeutung für die Aufrechterhaltung ihrer Funktionsfähigkeit und vollen Betriebskapazität, da unentdeckte Ausfälle nicht nur kostspielig, sondern auch lebensbedrohlich sein können. Angesichts der unerschwinglich hohen Kosten für die Behebung von Fehlern nach der Hardwareintegration ist die rechtzeitige Erkennung und Diagnose von Fehlern im CPS-Entwicklungszyklus von entscheidender Bedeutung. Daher ist eine gründliche Prüfung von CPS von größter Bedeutung, was die Notwendigkeit fortschrittlicher Fehlerdiagnoseverfahren unterstreicht, mit denen Fehler genau erkannt, erklärt und behoben werden können.

In dieser Arbeit entwickeln wir innovative Werkzeuge und Methoden, die auf CPSs zugeschnitten sind und darauf abzielen, eine schnelle und genaue Fehlerdiagnose zu ermöglichen. Unsere Beiträge umfassen Fehlerinjektion, globale Optimierung, suchbasierte Tests zur Fehlerlokalisierung, Mutationstests anhand formaler Eigenschaften und auf der Abdeckung von Signalmerkmalen basierende Tests. Dieser ganzheitliche Ansatz bietet erhebliche Vorteile gegenüber bestehenden Methoden und gewährleistet gründliche Tests in verschiedenen Szenarien.

Die vorgeschlagenen Lösungen wurden anhand öffentlich verfügbarer Benchmarks aus verschiedenen Bereichen bewertet, was die Anwendbarkeit und Wirksamkeit unserer Forschung belegt. Die Ergebnisse dieser Arbeit erweitern das Wissen über CPS-Tests und ebnen den Weg für zuverlässigere und sicherere CPSs in der Zukunft.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Contents

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

List of Publications

This thesis draws upon the research papers published in International Conference Proceedings and Journals. Across all of these papers, the authors are listed alphabetically.

- Drishti Yadav. 2024. From Fault Injection to Formal Verification: A Holistic Approach to Fault Diagnosis in Cyber-Physical Systems. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2024), Vienna, Austria, September 16-20, 2024*. ACM, New York, NY, USA, 1896–1900. https://doi.org/10.1145/3650212.3685552.
- Ezio Bartocci, Leonardo Mariani, Dejan Nickovic, and Drishti Yadav. 2022. FIM: fault injection and mutation for Simulink. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2022, Singapore, Singapore, November 14-18, 2022*, Abhik Roychoudhury, Cristian Cadar, and Miryung Kim (Eds.). ACM, USA, 1716–1720. https://doi.org/10.1145/3540250.3558932.
- Drishti Yadav. 2021. Blood coagulation algorithm: A novel bio-inspired metaheuristic algorithm for global optimization. *Mathematics*, 9(23), 3011. https://doi.org/10.3390/math9233011. ***Received the "Feature Paper" recognition.***
- Ezio Bartocci, Leonardo Mariani, Dejan Nickovic, and Drishti Yadav. 2022. Search-based Testing for Accurate Fault Localization in CPS. In *IEEE 33rd International Symposium on Software Reliability Engineering, ISSRE 2022, Charlotte, NC, USA, October 31 - Nov. 3, 2022*. IEEE, USA, 145–156. https://doi.org/10.1109/ISSRE55969.2022.00024.
- Ezio Bartocci, Leonardo Mariani, Dejan Nickovic, and Drishti Yadav. 2023. Property-Based Mutation Testing. In *IEEE Conference on Software Testing, Verification and Validation, ICST 2023, Dublin, Ireland, April 16-20, 2023*. IEEE, USA, 222–233. https://doi.org/10.1109/ICST57152.2023.00029.

In addition to the above mentioned published works, this thesis also includes material from the following unpublished work, which is currently under submission:

- Ezio Bartocci, Leonardo Mariani, Dejan Nickovic, and Drishti Yadav. 2024. Signal Feature Coverage and Testing for CPS Dataflow Models. *Under Submission.*

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Furthermore, the following publication was released during my PhD research, but it is unrelated to the thesis content:

- Drishti Yadav. 2021. Criteria for good qualitative research: A comprehensive review. *The Asia-Pacific Education Researcher*, 31(6), 679-689. https://doi.org/10.1007/s40299-021-00619-0.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

CHAPTER 1

Introduction

In an era defined by relentless innovation and technological advancement, the impact of Science and Technology on society has been profound and far-reaching. From the intricate machinery powering our daily lives to the vast network of interconnected systems shaping our collective future, the emergence of Cyber-Physical Systems (CPSs) stands as a testament to humanity's relentless pursuit of innovation and progress [OJL+22]. These systems, seamlessly integrating the digital and physical realms, have revolutionized countless aspects of society, from transportation and manufacturing to infrastructure and healthcare. Artificial Intelligence (AI), Robotics, Drones, and a multitude of interconnected devices form the foundation of CPSs, harmonizing a complex interplay of data and operations to enhance efficiency, safety, and convenience [Um19, SS22].

At the forefront of this technological revolution lie pillars such as autonomous driving, smart factories driving Industry 4.0, and the paradigm shift in urban living epitomized by smart mobility and buildings. For example, recent strides in technological advancements within CPSs have garnered considerable momentum, evidenced by initiatives like the Austrian "*5G Playground*," [5G 24] which is pioneering innovative solutions in intelligent transportation systems, robots and smart cities. Also, initiatives like the "*Austrian Action Programme on Automated Mobility*" [FM24] and "*Digibus Austria*" [oTG24] spearhead the development of autonomous vehicle technology, promising safer and more efficient transportation solutions for tomorrow's cities. These groundbreaking and transformative technologies are not exclusive to developed nations but are also being embraced by developing countries. For instance, India's "*Smart Cities Mission*" [MoHUAGoI24] seeks to harness CPSs to optimize urban infrastructure and enhance quality of life. Besides, the

©2024 Copyright held by the owner/author(s). Parts Reproduced, Reused, Reprinted, with permission, from Drishti Yadav. 2024. From Fault Injection to Formal Verification: A Holistic Approach to Fault Diagnosis in Cyber-Physical Systems. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2024), Vienna, Austria, September 16-20, 2024*. ACM, New York, NY, USA, 1896–1900. https://doi.org/10.1145/3650212.3685552

TU WIEN Bibliothek Your knowledge hub

seamless integration of CPSs stands as a cornerstone in India's endeavors towards space exploration, notably demonstrated in historic missions like "*Chandrayaan-3*" [ISRO24b] and "*Aditya-L1*" [ISRO24a] by India's space agency: Indian Space Research Organisation.

Cyber-Physical Systems (CPSs), by definition, transcend the boundaries between the digital and physical realms, embodying a symbiotic relationship between hardware and software. Beyond the rudimentary definition lies a complex interplay of sensors, actuators, AI algorithms, and communication networks, orchestrating a symphony of data and actions to achieve predetermined goals [Lee15]. More specifically, CPSs represent a cutting-edge fusion of computational prowess, communication capabilities, and control mechanisms, embodying highly collaborative, intelligent and automated systems [YZH20]. Unlike conventional systems, CPSs operate at a level of complexity that demands sophisticated design methodologies. These systems are tasked with orchestrating intricate interactions not only within their own subsystems but also with the external environment. In essence, CPSs transcend conventional engineering boundaries, focusing on the convergence of cyber and physical worlds to create dynamic, interconnected systems capable of adapting and responding to real-world challenges with unprecedented efficiency and precision [PZL12].

However, with technological progress comes heightened complexity. The proliferation of interconnected devices, coupled with the exponential growth of data and computational power, has ushered in an era where CPSs operate at unprecedented scales and intricacies. Sources of complexity abound, from the heterogeneity of system components to the dynamic and unpredictable nature of real-world environments. Amidst this complexity, the imperative for CPSs to be safe and reliable cannot be overstated. Mission-critical and safety-critical applications such as autonomous vehicles and medical devices hinge on flawless operation, with the slightest error carrying potentially catastrophic consequences. In essence, the following incidents stand as poignant reminders of the gravity of the situation and the possible repercussions of failure, encompassing both financial ramifications and human casualties.

1. *Northeast Blackout (2003)*: The Northeast blackout of 2003, caused by a software bug in the alarm system of the control room at FirstEnergy Corporation in Ohio, resulted in the loss of power for over 50 million people in the northeastern United States and parts of Canada[1]. This CPS failure led to widespread disruptions, economic losses, and compromised public safety.

2. *SpaceX CRS-7 Mission Failure (2015)*: In June 2015, SpaceX's Falcon 9 rocket carrying a Dragon spacecraft exploded shortly after liftoff during the CRS-7 mission to resupply the International Space Station (ISS). The failure resulted in the loss of valuable cargo and dealt a blow to space exploration efforts[2].

3. *Tesla Autopilot Crashes*: Incidents involving Tesla vehicles using the Autopilot feature have resulted in collisions and fatalities. Notable incidents include a 2016

[1]*Source*: Final Report on the August 14, 2003 Blackout
[2]*Source*: SpaceX CRS-7 Mission Failure Investigation Report

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

crash where a Tesla Model S in Autopilot mode collided with a tractor-trailer, resulting in the death of the Tesla driver[3].

4. *Uber Self-Driving Car Crash (2018)*: In March 2018, an Uber autonomous vehicle struck and killed a pedestrian in Tempe, Arizona. The vehicle was in self-driving mode at the time of the accident[4].

5. *Boeing 737 MAX Crashes (2018-2019)*: The crashes of Lion Air Flight 610 in October 2018 and Ethiopian Airlines Flight 302 in March 2019, both involving Boeing 737 MAX aircraft, resulted in a total of 346 fatalities. Investigations revealed flaws in the aircraft's Maneuvering Characteristics Augmentation System (MCAS), a CPS designed to enhance flight stability[5]. Following the occurrence of these two fatal crashes, Boeing encountered significant penalties, legal actions, and incurred substantial costs, amounting to billions of dollars, in efforts to enhance the safety of its products and restore its public image.

These examples serve as stark reminders of the need for robust design and testing to ensure the safety and reliability of CPSs across various domains, lest they lead to further loss of life, economic turmoil, and disruptions to essential services.

In order to address the risks associated with CPS failures, it is crucial to implement comprehensive design and Verification and Validation (V&V) procedures. With the advent of intelligent CPSs, verification needs to be conducted across different levels, encompassing both hardware and software components, and system-wide functionalities. Consequently, to streamline the development process and alleviate the complexities encountered by engineers during the design phase, novel frameworks have been established. These frameworks, including formal methods, simulation-based testing, and hybrid approaches, offer promising avenues for enhancing the effectiveness of V&V processes in ensuring the reliability and safety of CPSs. Among these approaches, *Model-Based Design* (MBD) has emerged as a pivotal strategy, with researchers and practitioners increasingly embracing model-based development as a cornerstone of this endeavor [JCL11]. By adopting a model-centric framework, engineers can simulate and analyze CPS behavior in a controlled environment under various conditions, facilitating early identification and resolution of potential issues before deployment, thus mitigating risks and enhancing system robustness. The benefits of this approach are manifold, ranging from early error detection and reduced development costs to streamlined collaboration and decision-making.

More precisely, MBD is a methodology that employs simulation to comprehend the behavior of either a forthcoming or an existing physical system. This approach, facilitated by model-based development tools, expedites the development of smarter connected systems by enabling engineers to simulate intricate products as systems-of-systems across the entire development life cycle [ARM17, TTT21]. Contrary to traditional design methodologies, MBD offers a distinct approach. Instead of relying on complex

[3]*Source*: National Highway Traffic Safety Administration (NHTSA) Investigation Report
[4]*Source*: National Transportation Safety Board (NTSB) Report
[5]*Source*: Final Report on Lion Air Flight 610, Final Report on Ethiopian Airlines Flight 302

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

structures and extensive software code, designers utilize MBD to craft plant models with sophisticated functional attributes using continuous-time and discrete-time building blocks. These models, when integrated with simulation tools, foster rapid prototyping, software testing, and verification [NM18, PGZ+23].

The importance of MBD for CPSs becomes particularly evident given the plethora of tools available for development. Numerous MBD tools support various facets of CPS development, each offering unique capabilities. For instance, SysML[6] and AADL[7] provide comprehensive modeling languages tailored for CPS design, while LabVIEW[8] offers a versatile platform for developing and deploying measurement and control systems. MATLAB™/Simulink®[9] from MathWorks®, widely favored by researchers, facilitates model-based development and simulation, enabling fail-safe design strategies for CPSs. Additionally, Modelica[10] enables multi-domain modeling and simulation, and dSPACE[11] provides real-time simulation and testing solutions. Besides, Ansys SCADE Suite[12] offers capabilities for graphical modeling, simulation, verification, and automatic code generation, ensuring the reliability and safety of CPS software. In addition to these tools, the academic community has devised various hybrid automata modeling frameworks to effectively capture the complexities of CPSs. These frameworks, including *HyVisual* [LZ06], *Petri Nets* [DA94], *PowerDEVS* [BK11], and *Shift* [DGV96], offer versatile approaches to modeling CPS dynamics. Collectively, these tools and frameworks empower engineers and researchers to efficiently design, simulate, and validate CPSs, underscoring the pivotal role of MBD in advancing CPS technologies. Among the arsenal of model-based development tools, MATLAB™/Simulink® stands out as a ubiquitous choice, offering a comprehensive suite of functionalities tailored to the intricacies of CPS design. Its robust simulation capabilities, coupled with extensive libraries and intuitive interfaces, make it indispensable for engineers and researchers navigating the complexities of modern CPSs.

It is imperative to acknowledge that achieving a bug-free state in a CPS is challenging. The heterogeneity of CPS components gives rise to various classes of faults, each demanding meticulous debugging and testing. These faults, whether permanent or transient, contribute to the complexity of CPS and necessitate rigorous debugging processes. As development progresses, the cost of error detection and correction escalates at each stage, underscoring the importance of ensuring system correctness as early as possible. Timely fault detection holds utmost importance, as addressing faults after hardware integration proves to be excessively costly. This has prompted the adoption of "*shift-left*" testing, emphasizing the need to address V&V concerns early in the development lifecycle and advocating for early intervention to preempt potential issues. By adhering to this

[6] https://sysml.org/
[7] https://insights.sei.cmu.edu/library/architecture-analysis-and-design-language-aadl-tool/
[8] https://www.ni.com/en/shop/labview.html
[9] https://in.mathworks.com/products/simulink.html
[10] https://modelica.org/
[11] https://www.dspace.com/en/ltd/home.cfm
[12] https://www.ansys.com/products/embedded-software/ansys-scade-suite

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Wien Bibliothek Your knowledge hub

approach, test suites are positioned to be robust, effectively identifying defects before they are introduced into the production environment. Consequently, adopting the shift-left mindset facilitates the timely detection and resolution of issues, contributing to the overall quality and reliability of the software product.

V&V techniques, particularly with the adoption of *Shift-left testing* (aka *Design-time testing*), are pivotal in ensuring a CPS's adherence to design constraints and performance expectations. These techniques, whether exhaustive or simulation-based, serve as essential checkpoints throughout the development process, helping to identify and rectify potential issues early on. However, the heterogeneous and intricate nature of CPSs introduces a myriad of theoretical and practical challenges to the verification process.

1.1 Research Gaps and Problem Statement

As discussed above, ensuring the rapid and accurate detection and diagnosis of faults is paramount in guaranteeing the viability and full operational capacity of safety-critical CPSs. The consequences of undetected failures in such systems can be both costly and life-threatening. Therefore, comprehensive testing is indispensable in CPS development to ensure fault-free operation and rectify faults before product release. In verifying the safety aspects of CPSs, it is essential that fault diagnostics are highly advanced, capable of precisely identifying the location and nature of faults. This advanced capability facilitates efficient fault rectification, contributing to the overall reliability and safety of the system.

Despite the increasing complexity of CPSs and advancements in test methodologies, there remains a notable gap in the provision of automated and effective tools and methodologies for systematic fault-based testing, including fault detection and analysis. This deficiency limits the ability of engineers to conduct comprehensive fault diagnostics of CPSs, thereby jeopardizing their reliability and safety in critical applications. Consequently, *there is a pressing need for the development of efficient tools and methodologies to assist engineers in apt fault diagnostics of safety-critical CPSs*, addressing this critical gap in current CPSs development practices and ensuring their robustness and integrity in real-world scenarios.

Under the umbrella of fault diagnostics and fault-based testing for CPSs, fault injection, optimization-based falsification, fault localization, mutation testing, coverage-based testing and formal verification form a cohesive network of activities, as illustrated in Figure 1.1. Rather than existing as isolated processes, these activities complement and reinforce each other synergistically, enhancing fault analysis and system testing. Each of these activities is integral, contributing significantly to the evaluation of CPS robustness, reliability, and fault tolerance. *Fault injection* involves deliberately introducing faults or errors into the system to evaluate how it responds under adverse conditions. Once faults are injected, *Fault Localization* (FL) comes into play, aiming to pinpoint the precise

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

location of these faults within the system. This step is essential for understanding the root causes of system failures and facilitating targeted debugging and rectification efforts. *Optimization-based falsification*, aka *Falsification Testing* (FT), employs (meta-)heuristic strategies to systematically search for inputs that violate specified safety properties, enhancing the effectiveness of fault detection. By leveraging (meta-)heuristic strategies, this approach effectively explores the system's behavior space to uncover potential failures. *Mutation Testing* (MT) complements fault injection and localization by assessing the effectiveness of test cases in detecting and addressing faults. It involves introducing small changes or mutations to the system's code or specifications to evaluate the adequacy of existing test cases. This process helps identify weaknesses in the testing approach and ensures comprehensive coverage of potential fault scenarios. Besides, *Coverage-Based Testing* focuses on creating test cases that thoroughly exercise different parts of the system to (i) achieve maximum *coverage*, and (ii) validate the behavior and functionality of the system under various operating conditions, including normal operation and fault scenarios. By generating diverse and comprehensive test cases, engineers can uncover latent faults and vulnerabilities in the system, enhancing its reliability and resilience.

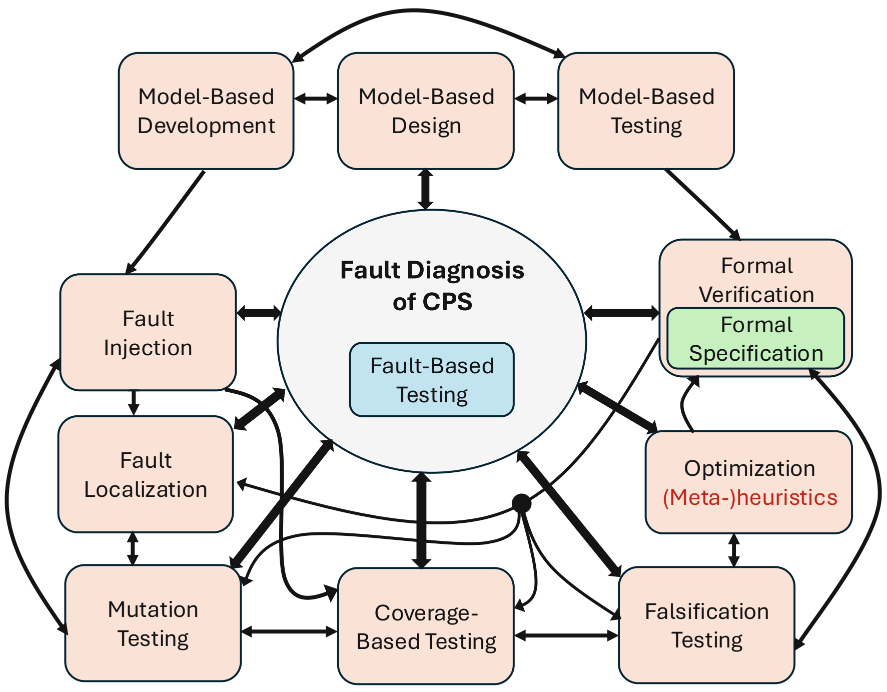


Figure 1.1: Overview of Fault Diagnosis of CPSs.

Formal Verification and *Formal Specifications* are also crucial elements in this network. *Formal Specifications* are detailed and precise descriptions of the behavior and properties that a system or component should exhibit. These specifications use formal languages and mathematical notation to define what the system is supposed to do, often in terms of its desired properties and constraints, serving as a benchmark for correctness. *Formal*

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Verification then uses mathematical techniques to prove or disprove whether the system meets these specifications. This process provides a rigorous method for validating the system's adherence to its requirements, complementing other testing activities by ensuring that the system meets formally defined criteria. Altogether, these activities create an integrated and interconnected network of processes that work in tandem to assess the robustness, reliability, and safety of CPSs. By integrating these approaches, engineers can effectively identify, analyze, and mitigate faults, ultimately enhancing the overall quality of CPSs.

In practice, fault diagnostics of CPSs can benefit from a structured methodology that preselects and prioritizes the most suitable techniques for identifying and addressing system faults. Building on the interconnected network of activities (illustrated in Figure 1.1) that can be flexibly applied in various sequences depending on the system's specific needs, a possible methodology would begin with *fault injection* to simulate potential faults and assess the system's response under controlled conditions, providing a basis for further fault analysis and testing. Preselecting fault injection allows the identification of vulnerable components or behaviors in the CPS, which lays the groundwork for the subsequent stages of the methodology. Once fault injection has been carried out, *optimization-based falsification* is employed to generate test inputs that could potentially lead the system into unsafe states. Falsification serves as a bridge between fault injection and other testing activities, enriching the exploration of the system's behavioral space to uncover a broader range of possible failures. After identifying faulty behaviors through fault injection and falsification, the focus is on *fault localization* to precisely identify the location of faults within the CPS, thereby providing a detailed fault map of the system. With potential faults identified and localized, *mutation testing* is used to assess the quality and robustness of the test cases, providing valuable feedback on the comprehensiveness of the testing approach and highlighting areas that may require more focused testing efforts. Building on the outcomes of mutation testing, *coverage-based testing* is employed to ensure maximum exploration of the system's behavior. Coverage-based testing not only seeks to validate the system's functionality under varied conditions but also aims to uncover latent faults that may have been overlooked in previous stages. Finally, *formal verification* serves as the conclusive step in the methodology: This step complements all previous activities by providing a high level of assurance that the system meets its formally defined requirements, thereby confirming its overall robustness and integrity.

The methodology outlined above integrates these activities where each stage builds upon the results of the previous ones. Fault injection sets the stage for exploring system vulnerabilities, optimization-based falsification enriches fault scenarios, fault localization identifies specific issues, mutation testing evaluates the testing adequacy, coverage-based testing ensures thorough validation, and formal verification provides a rigorous check against formal specifications. Note that this methodology is designed to be flexible, allowing the sequence of activities to be adjusted depending on the CPS's characteristics and specific fault diagnosis goals. For example, in some cases, starting with fault injection may be more appropriate to explore the system's response to faults before

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

formal verification. Alternatively, if the system's design is highly complex, performing formal verification upfront can help define the constraints and expectations that guide all subsequent testing efforts.

It is worth mentioning that the quality of a test suite is closely linked to its ability to pinpoint faults (fault localization) and detect them accurately. Additionally, there is a strong correlation between the extent of test coverage and the suite's capacity to identify faults effectively. Moreover, the thoroughness with which a test suite exercises the internal behavior of a system directly influences its ability to reveal faults. Consequently, it is imperative to pursue several strategies: (1) developing tools for automatically injecting faults into CPS models, thereby facilitating fault diagnostics; (2) improving the search-based testing of CPS using advanced meta-heuristic approaches; (3) designing methods to automatically generate test suites that possess high fault-localization capabilities; (4) devising test generation procedures that effectively uncover faults; and (5) creating effective strategies for generating tests that ensure maximum coverage of the system's behavior.

1.2 Scientific Goals and Objectives

This thesis focuses on the development of tools and techniques to verify the safety aspects of CPSs, thereby facilitating efficient fault diagnostics. Below, we delineate the primary research objectives, providing a succinct summary[13] of the current state-of-the-art limitations for each objective. Additionally, for each research objective, we pose a corresponding Research Question (RQ) and articulate the intended scientific goal.

1. **Injecting faults and mutating CPS models**
   Safety-critical CPSs must function reliably under both normal and adverse conditions, where components may exhibit unexpected behaviors. To ensure these systems meet stringent safety and reliability standards, adherence to industrial safety guidelines like ISO 26262 and IEC 61508 is crucial. These standards are designed to validate that the systems conform to necessary safety requirements and are built with fail-safe mechanisms to prevent catastrophic failures. Ensuring safety, a non-negotiable requirement in these systems, often involves evaluating how the system behaves under abnormal conditions, such as the presence of faults [BV07]. Techniques like fault injection and mutation testing are advocated by industrial safety standards to advance the development of CPSs, particularly in safety-critical fields such as the automotive industry [PFK^{+}13]. These methodologies help assess a system's fault tolerance and dependability [ALRL04], enabling a thorough verification of its safety and correctness [EL17].

   Automated and programmatic fault injection mechanisms are essential prerequisites for conducting large-scale mutation testing evaluations effectively. These mechanisms enable the injection of faults of various types into the system model in an automated

[13] In the upcoming chapters, we will provide a comprehensive reference list (*Related Work* section).

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

fashion, i.e., without requiring human intervention during the injection process. For an effective fault injection process, several key requirements must be fulfilled:

- *Automation*: The process should support fully automated fault injection without requiring human intervention, allowing for extensive evaluations.
- *Diversity of Fault Types*: It must allow the injection of a wide variety of faults to comprehensively test the system.
- *Scalability*: The mechanism should be scalable to accommodate complex and large-scale Simulink models typical of CPSs.
- *Seamless Integration*: It should integrate smoothly with existing simulation environments and testing frameworks used for Simulink models, facilitating a streamlined testing workflow.
- *Reproducibility*: Fault injections must be repeatable and consistent to ensure the reliability of the testing outcomes.

Currently, there is a notable absence of a fault injection solution for Simulink that fulfills all identified requirements. This deficiency contributes to the lack of systematic experiments on CPS testing approaches, leading us to the following research question:

RQ1: How to leverage automated and systematic injection of faults in CPS models to allow scalable experiments?

Goal: The objective is to develop a solution for automatically injecting faults and mutations into CPS Simulink models, meeting the specified requirements outlined above. Specifically, our focus is on leveraging existing state-of-the-art tools for fault injection and mutation to create a fault injection mechanism. This mechanism will offer a diverse range of faults and detailed options for fault configuration, enabling scalable[14] *experiments.*

2. **Improving search-based testing**
   Extensive research has been conducted on verifying CPSs through exhaustive simulations and monitoring algorithms. Safety assessments in CPSs are grounded in high-level system requirements expressed as formal temporal logic specifications, such as Signal Temporal Logic (STL) [DM10]. STL facilitates the representation of real-time properties over dense-time and real-valued behaviors of CPS, providing both qualitative and quantitative semantics. The quantitative robustness semantics of STL indicate *how robustly* a property[15] is satisfied or violated.

   In the context of CPSs, optimization algorithms, particularly metaheuristics, are essential for effective testing. These algorithms can systematically navigate the complex

[14] In this context, "scalable" refers to the fault injection process's ability to handle increasing complexity and size of CPS models without a significant loss in performance or effectiveness. The process should be capable of injecting faults into large-scale Simulink models with numerous components, signals, and intricate interactions while maintaining accuracy, efficiency, and automation.

[15] Throughout this thesis, we use the terms specification, formal requirement, (formal) property, and temporal logic formula interchangeably.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

input spaces of CPSs, uncovering faults and ensuring system robustness. Particularly, falsification is an effective method for identifying violations of a specification through counterexamples [ZLA$^+$21]. For automatic test generation in CPS safety evaluation, *optimization-based falsification*, or *search-based testing*, is frequently employed. This approach aims to identify a falsifying input that triggers a violation of the STL property. Despite the use of various metaheuristic strategies in falsification, such as Simulated Annealing [AF12] and Tabu Search [DJKM15], the trade-off between intensification (exploitation) and diversification (exploration), remains a significant challenge. Consequently, the development of new algorithms is crucial to address the unique and varied challenges presented by CPSs, guided by the necessity to not only balance exploration and exploitation in the search process, but also achieve faster convergence, and perform well in both constrained and high-dimensional spaces. By continuously innovating and refining these algorithms, we can significantly enhance the reliability and safety of CPSs in practical applications. This leads us to the following question:

RQ2: How can optimization algorithms be further refined to enhance testing of CPSs in practical applications?

Goal: The objective is to develop a high-level heuristic strategy to efficiently tackle the exploration-exploitation trade-off in search-based testing, also capable of handling high-dimensional search spaces and constrained optimization tasks, as those encountered in advanced CPS controllers.

3. **Accurately localizing faults in CPS models**
   Run-time monitoring techniques are effective for detecting faulty or spurious behaviors in systems by utilizing monitors for STL properties [FMN15]. Numerous efforts have been made to pinpoint faults in a system-under-test (SUT) by analyzing observed violating and falsifying traces. However, fault localization in the presence of multiple faults remains largely unaddressed in the literature. Besides, existing fault localization methods are often time-consuming and resource-intensive. Therefore, there is a need to develop methods that can accurately localize faults with lower costs and greater efficiency. This brings us to the following research question:

   RQ3: How to use the observed behaviors of a system to localize multiple faults (potentially large in number) accurately?

   Goal: The objective is to develop an accurate fault localization method closely integrated with testing, capable of maintaining robustness even when confronted with numerous faults in the SUT, potentially of varying types, while also being cost-effective.

4. **Mutation Testing of CPS models against formal specifications**
   Mutation testing, a recognized software quality assurance technique for evaluating test suites, is proficient at estimating the overall fault-revealing capacity of a test suite. However, its practicality and informativeness diminish when validating software against specific requirements, a common necessity for embedded software subjected to

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

rigorous safety property validation. In such cases, traditional mutation testing proves impractical and inefficient in fault detection. This leads us to the following research question:

RQ4: How can mutation testing be adapted or enhanced to effectively reveal faults in embedded software subjected to rigorous safety property validation?

Goal: The objective is to propose a new approach for mutation testing of CPS models against formal properties. More specifically, we aim to provide a formal definition for killing mutants with respect to properties. In addition, for CPS models, we focus on designing a test generation strategy to generate test cases that can kill the mutants with respect to formal properties effectively.

5. **Defining new notion of coverage for testing CPS models**
   Within the domain of embedded software testing, practitioners commonly assess the effectiveness of test suites by examining their code coverage and fault-revealing capabilities. Various metrics, such as input coverage, output coverage, and structural coverage, are employed to measure coverage adequacy. Structural coverage metrics, widely utilized in both research and industry [FA11, YLW06, IPJF19], offer insights into how thoroughly a test suite exercises system elements, thus identifying potential shortcomings in test cases. However, studies suggest that solely relying on structural coverage criteria may be insufficient for fault detection in software models and programs [GRS+16, IH14, NA09, SP10]. Particularly for CPS dataflow models, the interconnected nature of elements in these models means that executing a test case to cover one element may trigger the execution of numerous others. Consequently, depending solely on these metrics for CPS dataflow model testing may yield limited benefits, failing to provide a comprehensive understanding of how well the system's internal behavior has been tested.

   To adequately test CPS models, it is crucial to utilize more sophisticated and refined coverage metrics tailored to capture the intricate behaviors and interactions inherent in these models. By refining coverage metrics in this manner, we can ensure that the testing efforts adequately explore the full spectrum of system functionalities and potential failure modes. This leads us to the following research question:

   RQ5: How can coverage metrics be refined and advanced to effectively capture the nuanced behaviors and interactions within CPS models?

   Goal: The objective is to design a new coverage metric that thoroughly exercises the internal behavior of a CPS model. Additionally, we aim to (i) design a test generation strategy that maximizes our coverage metric, and (ii) investigate the effectiveness of the generated test cases in revealing faults.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

1.3 Methodology

The research outlined in this thesis predominantly adopts the Design Science Research (DSR) methodology [Hev07] to address the aforementioned research questions. Design science is a research approach centered on crafting and validating prescriptive knowledge. Particularly suited for practical engineering challenges, it furnishes frameworks for devising and assessing solutions to specific problems.

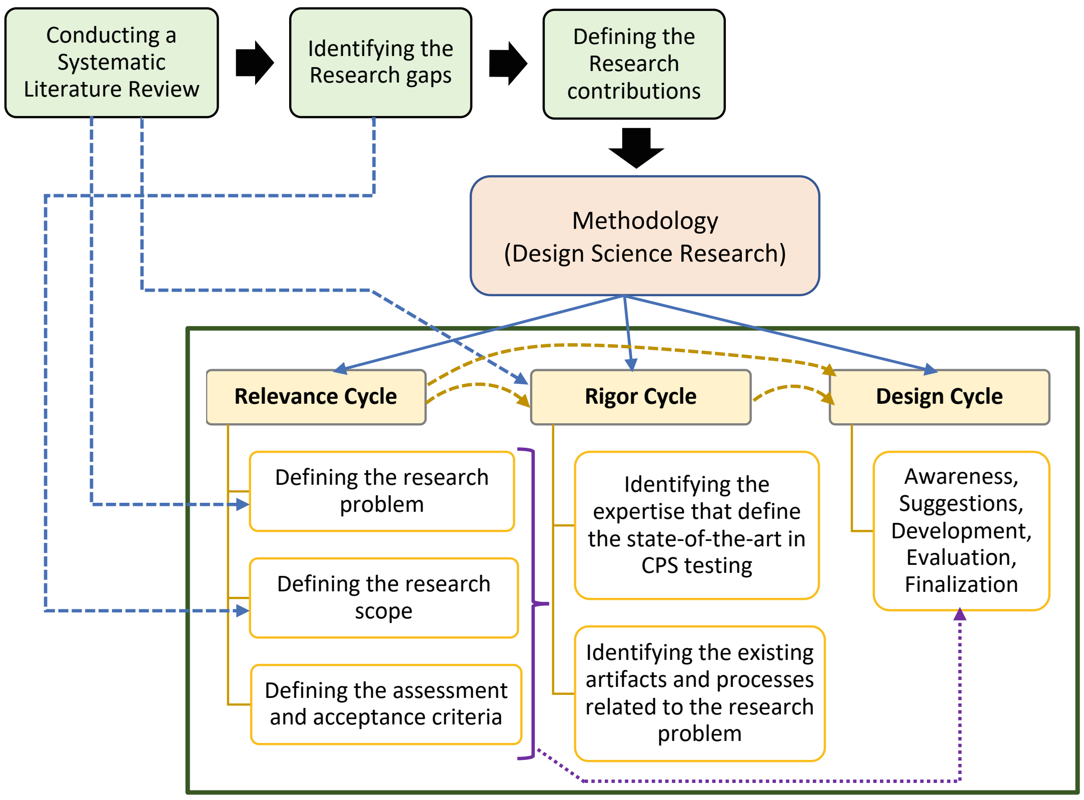

Conducting a Systematic Literature Review
Identifying the Research gaps
Defining the Research contributions
Methodology (Design Science Research)
Relevance Cycle
Rigor Cycle
Design Cycle
Defining the research problem
Defining the research scope
Defining the assessment and acceptance criteria
Identifying the expertise that define the state-of-the-art in CPS testing
Identifying the existing artifacts and processes related to the research problem
Awareness, Suggestions, Development, Evaluation, Finalization

Figure 1.2: Steps of research development methodology (Design Science Research).

Figure 1.2 illustrates the steps of the research development methodology. We addressed the problem identification and motivation in Section 1.1, and the objectives for the envisioned solution in Section 1.2. These activities were informed by Systematic Literature Reviews (SLRs) [KBB$^+$09], allowing for a comprehensive understanding of the current state-of-the-art and related work in the field. Our SLR on fault diagnostics and fault-based testing of CPSs, explored the extensive works on topics including, but not limited to: CPSs, V&V, testing, falsification, fault- and mutation-analysis, STL, fault identification and diagnosis.

Additionally, we outlined the artifacts that are planned to be designed and developed in the research in Section 1.2. Their contribution, encompassing tools and methods, to the body of knowledge in the realm of CPS testing, is encapsulated in Section 1.4. To assess our developed tools and methods, we employed Simulink benchmarks sourced from

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

different venues as our experimental subjects. Further information on these case studies and our experiments is provided in the subsequent chapters.

The prototypes and tools for the work detailed in Chapters 3-7 were developed using CPS models across different platforms and MATLAB versions. Additionally, we utilized different tools, including those for monitoring STL specifications, to implement our approaches. Each chapter will provide specifics regarding the platforms and tools used in the implementation process.

Lastly, the dissemination of the findings is addressed in "List of Publications". Generally, each response to the RQ yields a valuable artifact (either tools or methods), which is shared through publications in scientific conferences and journals.

1.4 Thesis Contributions

This thesis endeavors to construct a fault-based testing framework customized for safety-critical CPSs, streamlining and enhancing fault diagnosis in these systems. Specifically, it targets CPS Simulink models with safety requirements articulated using Signal Temporal Logic (STL). The thesis introduces innovative tools and methodologies that surpass conventional approaches, enabling thorough CPS testing. Illustrated in Figure 1.3, the thesis unfolds with the following distinctive contributions:

Enhanced Fault Injection and Mutation Framework for Simulink Models. One of the principal contributions of this thesis is the introduction of an open-source tool named **FIM** for systematic **F**ault **I**njection and **M**utation in Simulink models. Unlike previous approaches, FIM offers automated and programmable fault injection capabilities, eliminating the need for manual intervention during the injection process. FIM offers a broader range of faults and mutations for CPS Simulink models and streamlines automated fault injection. Additionally, FIM is highly scalable, generating numerous mutants quickly via configuration files. Moreover, it enables testers to control fault block activation and parameter adjustment, enhancing rigorous testing of the SUT against failures to assess its fault tolerance.

Introducing Blood Coagulation Algorithm: A Bio-inspired Global Optimizer. This thesis presents a novel bio-inspired optimization algorithm inspired by the biological process of blood coagulation, called the **B**lood **C**oagulation **A**lgorithm (BCA), which surpasses other state-of-the-art optimizers in terms of exploration, exploitation, avoidance of local optima, and convergence. As the thesis unfolds, we will illustrate BCA's efficacy in navigating high-dimensional search spaces, handling constrained optimization tasks, and facilitating the optimization-based falsification (i.e., search-based testing) of CPS models. These qualities position BCA as a competitive solution for addressing real-world optimization challenges, including those encountered in CPS testing.

Improving Fault Localization with Search-based Testing. As another significant contribution, this thesis presents a novel search-based testing method for precisely localizing faults in CPS Simulink models. The method leverages two test executions

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

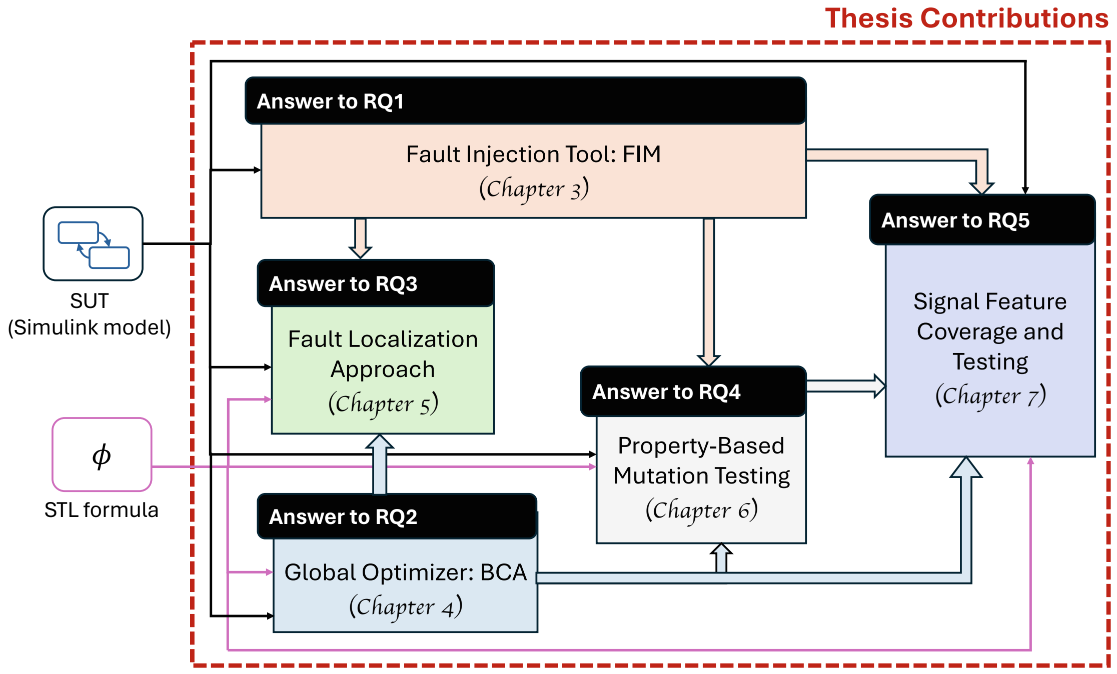


Figure 1.3: Overview of the Thesis: Mapping Contributions to Research Questions and Dependencies.

with contrasting outcomes (one passing and the other failing) to provide precise details about the fault location, thereby reducing the scope and effort required for debugging. Furthermore, our method demonstrates the capability to effectively localize multiple faults while enhancing fault localization cost and computational efficiency, as detailed later in this thesis.

Enhancing Test Suite Evaluation with Property-Based Mutation Testing. This thesis introduces **P**roperty-**B**ased **M**utation **T**esting (**PBMT**), a novel extension to traditional mutation testing method, to evaluate the effectiveness of test suites in relation to properties, a previously unexplored area in mutation-based testing literature. Additionally, this research proposes a search-based approach for generating test cases to automatically identify mutants relevant to PBMT experiments, supported by empirical findings for CPS Simulink models.

Introducing Signal Feature Coverage for Enhanced CPS Testing. Another key contribution of this thesis is the introduction of the innovative concept of *signal feature coverage*, a coverage criterion specifically designed for CPS dataflow models. Signal feature coverage is based on standard signal features and involves exercising the internal signals of a CPS dataflow model concerning their various time and frequency domain characteristics. Furthermore, this thesis proposes a search-based test generation strategy aimed at maximizing this coverage. As elaborated upon in subsequent chapters of the thesis, we will evaluate the performance of existing test generation strategies in terms

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

of signal feature coverage and demonstrate how test suites designed to maximize this coverage can enhance fault detection.

1.5 Thesis Organization

This thesis presents significant contributions (outlined in Section 1.4) aimed at enhancing fault diagnosis in safety-critical CPS. The contents of the thesis stem from four *published* conference papers [BMNY22a, BMNY22c, BMNY23, Yad24], one *published* journal paper [Yad21] and one *unpublished* work (which is under submission and review), reflecting research conducted between the years 2020 and 2024. In this introductory chapter, we have delved into the context and motivations behind this thesis. The subsequent chapters are structured as follows:

Chapter 2. This chapter lays the groundwork for subsequent chapters, introducing foundational concepts essential for the analyses and implementations ahead. We begin by exploring the basics and syntax of Signal Temporal Logic (STL). Following this, we delve into a detailed examination of CPS Simulink models, covering their structure and functionalities. Additionally, we establish foundational definitions pertinent to mutation testing. Finally, we explore methods for conducting mutations on Simulink models. The material presented in this chapter comes from [BMNY22c] and [BMNY23].

Chapter 3. In this chapter, we introduce our fault injection and mutation tool, referred to as **FIM**, which holds a central role in this thesis. We begin by elucidating the rationale behind developing FIM. Next, we delve into the conceptual challenges that informed and influenced the features of FIM, followed by an exploration of its architecture. We illustrate the tool's application through a case study, present evaluation findings, review related work on fault injection and mutation, and conclude by summarizing our contributions. The content of this chapter is primarily derived from [BMNY22a], with the exception of certain details regarding tool usage. Note that FIM is the foundational tool for fault seeding and mutant generation in the subsequent chapters (i.e., Chapters 5, 6 and 7).

Chapter 4. In this chapter, we introduce the *Blood Coagulation Algorithm* (BCA), our bio-inspired optimization algorithm, which stands as another significant contribution of this thesis. We delve into the inspiration behind the algorithm and its workings, followed by a comprehensive assessment of BCA's performance across various mathematical functions, real-world engineering design challenges, and falsification task involving CPS Simulink model. The chapter closes with a summary of the contributions and findings. All the content in this chapter is sourced from [Yad21]. Note that the BCA optimizer will be utilized for handling optimization tasks in the following chapters (i.e., Chapters 5, 6 and 7).

Chapter 5. In this chapter, we introduce a *search-based testing* method aimed at precisely localizing faults within CPS Simulink models, constituting another fundamental contribution of this thesis. We tackle fault localization in scenarios involving both explicit specifications (i.e., when formal properties are present) and implicit specifications (i.e.,

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

when formal properties are absent). Following this, we present the results of our evaluation on CPS Simulink models, illustrating the effectiveness of our approach compared to a state-of-the-art fault localization technique. The chapter concludes with an examination of related research and a summary highlighting the primary contributions and findings. All information in this chapter comes from [BMNY22c].

Chapter 6. In this chapter, we delve into our novel approach to mutation testing, called *Property-Based Mutation Testing (PBMT)*, which evaluates software against formal properties, adding a new dimension to traditional mutation testing. We start by explaining the motivation behind PBMT, followed by formally defining how to kill a mutant with respect to a property. Next, we explore a search-based test generation strategy aimed at creating mutation-adequate test cases. Then, we present the evaluation results obtained from applying our approach to Simulink models, comparing them with state-of-the-art testing techniques and discussing the insights obtained from our experiments. Finally, we wrap up the chapter with a discussion on related work and a brief summary of our contributions and findings. The content in this chapter is sourced from [BMNY23].

Chapter 7. In this chapter, we introduce *signal feature coverage*, a new coverage criterion for evaluating the thoroughness of test suites for testing CPS dataflow models. We also present *feature-coverage testing*, a search-based test generation strategy specifically designed to maximize signal feature coverage. Additionally, we apply our feature-coverage testing approach to various sets of target signals and their respective features, including scenarios where CPS models are tested against STL specifications. We then present our evaluation results through experiments with various Simulink models and comparisons to current testing methods. Lastly, we discuss the related work and conclude the chapter with a concise summary of our contributions and findings. The content of this chapter is unpublished and currently under submission.

Chapter 8. This chapter serves as the conclusion of this thesis, recapitulating the primary outcomes and research findings, and proposing future research avenues aimed at advancing fault diagnosis in CPSs.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

CHAPTER 2

Background

In this chapter, we provide essential background information necessary for understanding the rest of this thesis. Additionally, we introduce the notations and fundamental concepts utilized consistently throughout this work.

2.1 Signal Temporal Logic (STL)

In recent years, researchers have employed temporal logic formalisms to express safety properties for the verification of safety-critical CPSs. Signal Temporal Logic (STL) [MN13] is a prominent formalism for specifying temporal properties of dense-time real-valued behaviors in hybrid systems, which include both continuous and discrete dynamics, such as those found in safety-critical CPSs. The formal syntax of STL is defined as follows:

$$\Phi := f(\mathbf{x}(j)) > 0 \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \Box_I \Phi \mid \Diamond_I \Phi \mid \Phi_1 \mathcal{U}_I \Phi_2$$

In this context, the formula $f(\mathbf{x}(j)) > 0$ represents a *signal predicate*, where $\mathbf{x}(j)$ denotes the value of signal $\mathbf{x}$ at time instant j, and f is a function mapping the signal domain $\mathcal{D}$ to $\mathbb{R}$. The interval $I \subseteq \mathbb{R}_{\geq 0}$ specifies an arbitrary time period. The propositional logic operators $\neg$ and $\wedge$ follow standard logical semantics, with $\neg$ indicating logical negation and $\wedge$ indicating logical conjunction. Additional temporal operators are defined as follows:

©2022 IEEE. Parts Reproduced, Reused, Reprinted, with permission, from Ezio Bartocci, Leonardo Mariani, Dejan Ničković and Drishti Yadav, "Search-based Testing for Accurate Fault Localization in CPS," 2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE), Charlotte, NC, USA, 2022, pp. 145-156, https://doi.org/10.1109/ISSRE55969.2022.00024

©2023 IEEE. Parts Reproduced, Reused, Reprinted, with permission, from Ezio Bartocci, Leonardo Mariani, Dejan Ničković and Drishti Yadav, "Property-Based Mutation Testing," 2023 IEEE Conference on Software Testing, Verification and Validation (ICST), Dublin, Ireland, 2023, pp. 222-233, https://doi.org/10.1109/ICST57152.2023.00029

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

- The *always* operator denoted by $\Box$: $\Box_I \Phi$ signifies that Φ must be true for every sample within the interval I.
- The *eventually* operator denoted by $\Diamond$: $\Diamond_I \Phi$ means that Φ must be true *at least once* during the interval I.
- The *until* operator denoted by $\mathcal{U}$: $\Phi_1 \mathcal{U}_I \Phi_2$ means that Φ_1 must hold continuously within the interval I until Φ_2 becomes true.

The Boolean satisfaction semantics, also known as *qualitative semantics*, of STL provides a boolean evaluation of the property Φ. If the signal predicate is satisfied, its Boolean satisfaction is $\top$; otherwise, it is $\bot$. We use the operators $\mathcal{U}$, $\Diamond$, and $\Box$ to represent $\mathcal{U}_I$, $\Diamond_I$, and $\Box_I$ when the interval I is $[0, \infty)$.

In addition to qualitative semantics, STL provides *quantitative robust semantics* [DM10], which enables the computation of the degree to which the traces generated by a system satisfy the property Φ when executed against a test input. More in details, the quantitative robust semantics of STL offer a *robustness degree*, indicating the distance between the observed behavior and the set of behaviors specified by Φ. For an STL formula Φ, the *robustness* quantifies the satisfaction of a trace w with respect to Φ. This is expressed as a real-valued quantity $\rho : R(w, \Phi)$, such that: (1) $\rho > 0 \Rightarrow w \models \Phi$, and (2) $\rho < 0 \Rightarrow w \models \neg\Phi$. For a comprehensive understanding of the quantitative robust semantics of STL properties, we refer the reader to the work by Donze and Maler [DM10].

2.2 CPS Simulink Models

In the preceding chapter, we discussed the prevalent use of the MathWorks™ Simulink® environment in the MBD of CPSs [Mat22e,Mat22b]. Simulink enables non-software engineers to craft intricate systems, compile them into low-level code, and simulate the models to observe their performance with various test inputs. During the MBD of a CPS in the MathWorks™ Simulink® environment, engineers construct a Simulink model $\mathcal{M}$ representing the system as a block diagram. Below, Figure 2.1 presents an example of a Simulink model along with its key components. A Simulink model $\mathcal{M}$ comprises the following components:

- A collection of *blocks* $\mathcal{B}$: These serve as the fundamental units within a Simulink model. The functional behavior of each block delineates the cause-effect relationship, defining its operational role.
- A series of *ports* $\mathcal{P}$: Blocks receive data through their *input ports* and transmit it via their *output ports*. It is worth noting that blocks may also include *trigger*, *action*, *enable*, and *reset* ports, which we omit from our examination.
- An array of *lines* $\mathcal{L}$: These represent connections that indicate the flow of data or signal propagation from one block to another. The block transmitting data is termed the *source* block, while the receiving block is the *target* block. Every line $l \in \mathcal{L}$ is distinctly identified by the output ports of its source block and the input ports of its target block.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

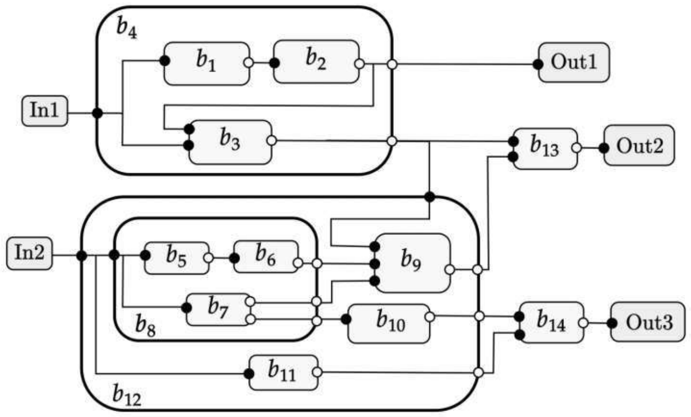


Figure 2.1: Example of a Simulink model. Black nodes indicate input ports while white nodes indicate output ports. Blocks b_4, b_8, b_{12} are hierarchical; remaining blocks are atomic. ©2022 IEEE.

Each of the mentioned components is assigned a unique numeric identifier within the model, referred to as its *handle*. This handle serves as a pointer to access and modify the corresponding component within the model. Typically, a model $\mathcal{M}$ retrieves inputs from designated input blocks or sources (e.g., as illustrated by the *In1* and *In2* blocks in Figure 2.1) and generates outputs through designated output blocks or sinks (e.g., the *Out1*, *Out2*, and *Out3* blocks in Figure 2.1). Additionally, $\mathcal{M}$ comprises a set of state variables $\mathcal{V} = \{\mathcal{V}_I, \mathcal{V}_H, \mathcal{V}_O\}$ s.t. $\mathcal{V}_I$, $\mathcal{V}_H$ and $\mathcal{V}_O$ represent the input, hidden or internal and output state variables respectively, and a set of signals $\mathcal{S}$, where signals are defined as the mapping $\mathcal{S} \colon \mathcal{L} \to V$.

Additionally, blocks in a Simulink model can be classified into two types: *atomic* and *hierarchical*. An atomic block does not contain any other blocks within it, whereas a hierarchical block encompasses atomic blocks as well as other hierarchical blocks. A block is considered the *parent* of one or more blocks when it encapsulates them as its subsystems. In Figure 2.1, the *top-level* model $\mathcal{M}$ is at hierarchy level 1. Furthermore, b_1 to b_3, b_8, and b_9 to b_{11} are children of $\mathcal{M}$ at hierarchy level 2, while b_5 to b_7 are children of $\mathcal{M}$ at hierarchy level 3.

Moreover, Simulink provides numerous *libraries* containing built-in blocks that streamline the design and rapid development of CPS. Users also have the option to create and integrate custom libraries and blocks tailored to their specific requirements, including user-defined functionalities and customizable parameters.

Upon completing the CPS design in Simulink, users proceed to *compile* and *simulate* the model $\mathcal{M}$ using a specified test case t, which represents an input to the system. We define a function *Simulate* that takes $\mathcal{M}$ and t as inputs, executes the simulation of $\mathcal{M}$, and returns traces of all input-internal-output (IIO) signals as the final output. Assuming the model employs fixed-length sampling, representing a broad category of Simulink models, a trace consists of a sequence of (timestamp, value) pairs, with a constant period between

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

consecutive samples. We denote all traces generated for $\mathcal{M}$ for a given test case t as $out(\mathcal{M}, t)$. Symbolically, $out(\mathcal{M}, t) = Simulate(\mathcal{M}, t)$. Generally, a *trace* refers to the progression of the states of $\mathcal{M}$ over discrete time steps, from $q = 0$ to $q = q_T$, where the finite time horizon $q_T > 0$. The values in a simulation trace are recorded in chronological order of signal evolution, ranging from $q = 0$ to $q = q_T$.

2.3 Mutation Testing

In this section, we introduce the foundational principles and background information regarding regular mutation testing (MT).

MT is based on the following two foundational assumptions [ABD+79, DLS78]: (1) the *Competent Programmer Hypothesis* which posits that “programmers create programs that differ from the correct one mostly by small syntactic errors” [BMNY23], and (2) the *Coupling Effect* which suggests that “complex faults are coupled to simple faults in such a way that a test data set that detects all simple faults in a program will detect a high percentage of the complex faults” [GJG17]. Numerous studies have explored these hypotheses, demonstrating that MT results can reliably predict a significant portion of high-priority real bugs [PIFJ21, JJI+14, ABLN06, ABL15]. While not every bug is coupled with mutants, MT remains a valuable tool for assessing the quality of a test suite.

Now, let us delve into the fundamental concepts of MT.

Definition 2.3.1 (*Mutation operator*)**.** “A *mutation operator* is a source-code transformation that introduces a modification in the program-under-test. More rigorously, given a program $\mathcal{P}$, a mutation operator op is a function that takes as inputs $\mathcal{P}$ and a location k inside $\mathcal{P}$ and creates a syntactic alteration of $\mathcal{P}$ at location k, if the location can be mutated with op.” [BMNY23]

Definition 2.3.2 (*Mutant*)**.** “For a given program $\mathcal{P}$ and a set of mutation operators $\mathcal{O} = \{op_1, op_2, ..., op_n\}$, a *mutant* p is the result of the application of a mutation operator $op \in \mathcal{O}$ to $\mathcal{P}$ at a specified location k. A mutant created by the application of only one mutation operator to $\mathcal{P}$ is known as *First Order Mutant* (FOM). The application of multiple mutation operators to $\mathcal{P}$ results in a *Higher Order Mutant* (HOM) [JH09].” [BMNY23]

For a test suite $\mathcal{T}$ and a test t in $\mathcal{T}$, we denote $t \models p$ when the test passes on the mutant p, and $t \not\models p$ when the test fails on the mutant p. The output generated by p with t is represented by $O(t, p)$, and $\mathcal{T}_U^p$ denotes the universal set comprising every possible valid test case for p.

Definition 2.3.3 (*Killed Mutant*)**.** “A mutant p is said to be *killed* by $\mathcal{T}$ if at least one test case t in $\mathcal{T}$ fails when exercising p, i.e., $\exists t \in \mathcal{T} : t \not\models p$.” [BMNY23]

Definition 2.3.4 (*Live Mutant*)**.** “Mutants that do not lead to the failure of any test case $t \in \mathcal{T}$ are said to be *live* or survived. Formally, p is said to be live if $\forall t \in \mathcal{T}$, $t \models p$.” [BMNY23]

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Definition 2.3.5 (*Equivalent Mutant*)**.** A mutant p is considered *equivalent* to the original program $\mathcal{P}$ if they produce identical outputs for every possible input. Formally, p is equivalent to $\mathcal{P}$ if $\forall t \in \mathcal{T}_U^{\mathcal{P}}, O(t,p) = O(t,\mathcal{P})$. Essentially, no test case can differentiate between an equivalent mutant and the original program [DPV+20]. It is important to note that detecting equivalent mutants is an undecidable problem.

Definition 2.3.6 (*Invalid Mutant*)**.** "A mutant p is considered *invalid* if it cannot be compiled [VDB+22]. Such a mutant is not included in the mutation coverage." [BMNY23]

Definition 2.3.7 (*Mutation coverage*)**.** The adequacy of a test suite $\mathcal{T}$ can be gauged using the *mutation coverage* (referred to as the *mutation score* $\mathcal{MS}$), which is calculated as the ratio of killed mutants to the total number of valid non-equivalent mutants:

$$Mutation\ coverage = \frac{\#killed\ mutants}{\#valid\ mutants - \#equivalent\ mutants}$$

$\mathcal{T}$ is considered to achieve complete (i.e., 100%) mutation test adequacy if it successfully kills all non-equivalent valid mutants. Achieving full mutation coverage guarantees that $\mathcal{T}$ is both robust against the modeled mutation types and responsive to minor modifications in the program-under-test ($\mathcal{P}$).

Definition 2.3.8 (*Redundant Mutant*)**.** Redundant mutants are not beneficial as they utilize resources without adding value to the testing process since they are killed whenever other mutants are killed. This redundancy can be characterized by *duplicate* and *subsumed* mutants [PKZ+19]. *Duplicate* mutants are equivalent to each other but not equivalent to the original program [OPB21]. On the other hand, *Subsumed* mutants are not equivalent to each other but are killed by the same test cases. The subsumption relation is defined as follows [KAO15]: We say that p_i *subsumes* p_j, denoted $p_i \rightarrow p_j$, if and only if the following two conditions are met:

1. $\exists t \in \mathcal{T}_U^{\mathcal{P}} : t \not\models p_i$. More simply, there is a test case t for which p_i and $\mathcal{P}$ produce distinct outputs, indicating that p_i is not equivalent to $\mathcal{P}$.
2. $\forall t \in \mathcal{T}_U^{\mathcal{P}}$, if $t \not\models p_i$, then $t \not\models p_j$. Essentially, if p_i produces a different output than $\mathcal{P}$ for any possible test case t applied to $\mathcal{P}$, the same holds true for p_j.

In Regular MT, for a test case $t \in \mathcal{T}$ to kill a mutant p, it must satisfy the following three conditions [OP97, OU01]:

1. *Reachability*: the test case t should *reach* the mutated statement in p.
2. *Necessity*: the test case t should *infect* the program state, leading to distinct program states for p and $\mathcal{P}$.
3. *Sufficiency*: the incorrect (or erroneous) program state should *propagate* to the output of p and be checked by t, demonstrating a noticeable variation in the outputs of p and $\mathcal{P}$ for t.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

The aforementioned three conditions are referred to as the *Reach-Infect-Propagate (RIP) model.* The effectiveness of a test case $t \in \mathcal{T}$ in killing a mutant p is influenced by the observability of the program state, resulting in the following two prevalent types of MT:

1. *Weak MT*: A test case $t \in \mathcal{T}$ kills a mutant p if only the first two conditions of the *RIP model* are satisfied.
2. *Strong MT*: A test case $t \in \mathcal{T}$ kills a mutant p iff all three conditions of the *RIP model* are fulfilled.

Automated tests, commonly equipped with an *oracle*, typically check particular segments of the output state by comparing the observed program behavior with the expected one. Nonetheless, if the oracle fails to check the specific portion of the output state that contains the erroneous value, it will not detect the failure. Thus, the oracle should also *reveal* the failure [LO16], according to the *Reach-Infect-Propagate-Reveal (RIPR)* model.

2.4 Faults and Mutations in Simulink

In the preceding section (Section 2.3), we provided a general overview of Mutation Testing. When applying these concepts to Simulink models, the program $\mathcal{P}$ corresponds to the Simulink model, and mutations represent changes made to the model's components or signal paths. In this context, the location k in the program $\mathcal{P}$ where the mutation operator is applied is either a model's component or a signal path.

Simply put, mutations in Simulink models involve making changes to the behavior of a Simulink model. Typically, there are two primary methods to make these alterations:

1. *Line mutations*: Modifying the behavior of signals that travel through lines connecting different blocks (as shown in 'Fault in *line*' in Figure 2.2).
2. *Block mutations*: Altering the behavior of a block, such as changing its functionality (illustrated by 'Faults in block' in Figure 2.2).

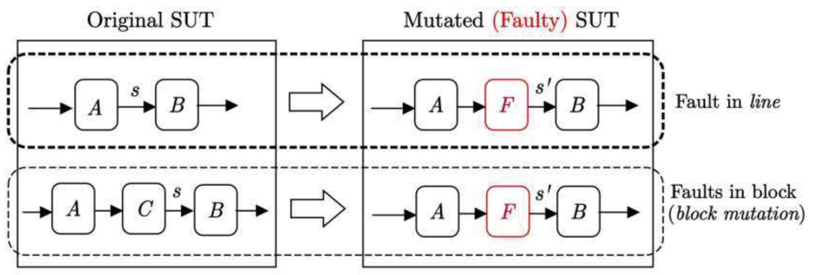


Figure 2.2: Mutations in a SUT (the seeded fault blocks F are highlighted in red). A, B and C are blocks of original SUT. Internal signals s and s' provide knowledge of the fault location. ©2023 IEEE.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

CHAPTER 3

Fault Injection

We present FIM, an open-source toolkit designed for automated fault injection and mutant generation within Simulink models. FIM enables the injection of faults into specific components, accommodating various common fault types and mutation operators. Users can customize parameters to regulate fault actuation timing and persistence. Moreover, additional flags empower users to activate individual fault blocks during testing, facilitating the observation of their impacts on overall system reliability. We offer insights into FIM's design and architecture, along with an evaluation of its performance using a case study derived from the avionics domain.

3.1 Introduction

Safety-critical CPSs are required to maintain reliability even during unexpected scenarios. Consequently, adherence to relevant industrial standards (like ISO 26262 and IEC 61508) is imperative to ensure that such systems meet the necessary safety requirements.

To address the increasing complexity of modern CPS, researchers and industry professionals have embraced model-based design and development coupled with a lightweight verification approach based on simulation and testing. The MATLAB™ Simulink® ecosystem from MathWorks® has emerged as the predominant standard for MBD in the CPS domain. Over the past decade, the formal methods and control theory communities

©2022 Copyright held by the owner/author(s). Parts Reproduced, Reused, Reprinted, with permission, from Ezio Bartocci, Leonardo Mariani, Dejan Ničković, and Drishti Yadav. 2022. FIM: Fault Injection and Mutation for Simulink. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'22), November 14-18, 2022, Singapore, Singapore.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3540250.3558932 have introduced more systematic approaches to simulation-based verification. Notably, falsification testing has gained traction in this context [NSF$^+$10]. This line of research has yielded numerous publications, including contributions featured in prominent conferences such as CAV [AH15, ADD$^+$17, ZHA19, ZLA$^+$21].

While methods like falsification testing have demonstrated success in identifying bugs within CPS designs, it is notable that these methods are often assessed using a limited number of examples and in an ad-hoc manner. A systematic evaluation of a testing strategy typically involves analyzing its ability to detect undesirable system behaviors in the presence of faults [BV07, BBC$^+$16]. This evaluation is typically conducted through fault injection [ALRL04] and mutation testing [JH11], activities recommended by industrial safety standards, particularly in safety-critical domains [PFK$^+$13]. For large-scale mutation testing evaluations to be conducted effectively, it is essential to have a mechanism for automatically and programmatically injecting faults of various types into the system model, without manual intervention during the injection process. Currently, there is a lack of a fault injection solution tailored for Simulink that fulfills all the identified requirements, which explains the absence of systematic experiments in evaluating CPS testing approaches.

To address this issue, we introduce **FIM** (**F**ault **I**njection and **M**utation Engine), an open-source tool designed to inject faults into Simulink models in a manner that meets the specified requirements. FIM offers the following programmable features:

- A comprehensive fault model encompassing sensor, hardware, and network faults, alongside a mutation operator library;
- Systematic fault injection across various segments of a model:
  - Creation of multiple copies of the original model, with each copy containing a single injected fault, or
  - Generation of a single copy of the original model, with all faults injected into that specific copy;
- Option to confine fault injection to specific sections of the model;
- Dynamic activation and deactivation of faults

Chapter organization. Section 3.2 offers an overview of FIM's architecture, core features, and operational flow. The utility of the tool is outlined in Section 3.3. Section 3.4 assesses the tool's efficacy through experimental data derived from an aerospace domain example. In Section 3.5, we delve into the related literature. Finally, Section 3.6 provides concluding remarks.

3.2 Architecture and Implementation of FIM

In the process of designing and implementing our fault injection and mutation framework, we encountered several conceptual challenges that motivated and shaped the features of

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

FIM:

- *Understanding the Specifics of Simulink*: We grappled with the intricacies of Simulink models, delving into the programmatic editing of model components and grasping the nuances of masking. This understanding was crucial for designing a comprehensive fault injection library with parameterized blocks.
- *Exploring Injection Strategies*: We experimented with various injection strategies before settling on the addition of new blocks as the preferred mechanism. This approach offered the highest level of control without compromising the ability to observe and alter variable values.
- *Enabling Large-Scale Experiments*: We prioritized the capability to conduct large-scale experiments, developing an interface that minimizes the need for manual interactions during experiments.
- *Handling a Large Set of Injected Faults*: We explored multiple strategies for managing a large set of injected faults, ultimately supporting two complementary approaches: generating multiple models with one fault each and generating a single model with all faults included.

Now, we walk through the tool architecture and workflow, emphasizing its essential features. Figure 3.1 illustrates the architecture of FIM, with its three primary components: (1) the *Fault Injection Library*, (2) the *Fault Injection Module*, and (3) the *Fault Configuration*.

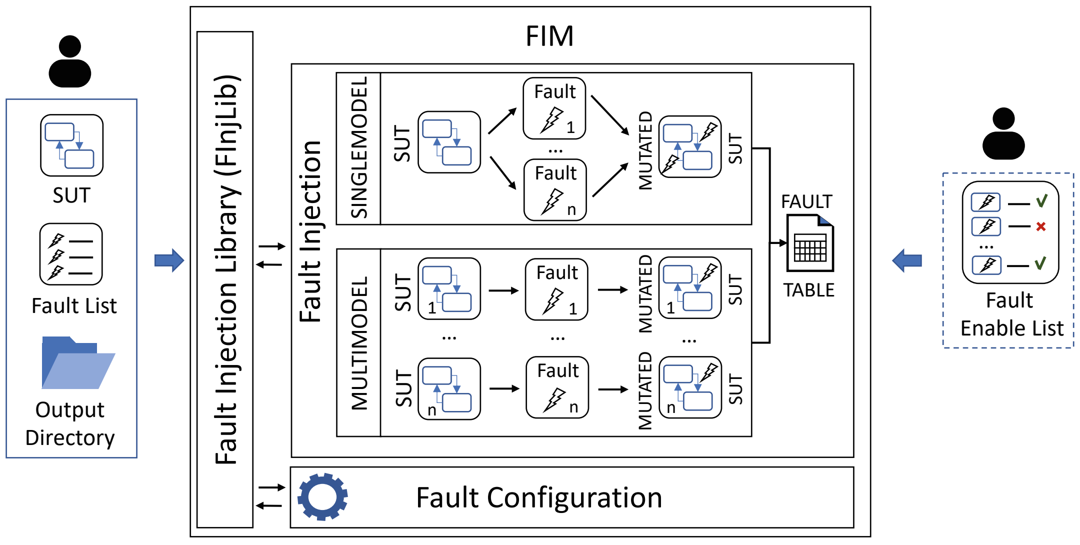


Figure 3.1: Workflow and Structure of FIM.
License: CC BY 4.0

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

3.2.1 Fault Injection Library

The Fault Library in FIM is a specialized Simulink library browser that contains various parameterized blocks designed for different types of faults and mutation operators. Users have the flexibility to expand this library with additional faults and mutation operators as needed. Depending on the fault type specified by the user in the list of desired mutations (referred to as *fault_list*[1]), the insertion or replacement of blocks occurs accordingly. Blocks labeled as "Faults" in Table 3.1 represent the insertion of fault blocks, while those associated with "Block mutations" indicate the replacement of the corresponding block with the specified block mutation operator. Figure 3.2 illustrates an instance of both line mutation and block mutation within the Simulink model of a system.

Table 3.1: Faults and mutation operators.

Type	**Name**	**Description**
Faults (Line mutations)	`Negate`	changes a signal u to $-u$
	`Invert`	inverts a non-zero signal u to 0; if u is 0, makes 0 to 1
	`Stuck-at 0`	makes the signal value zero
	`Absolute`	changes a signal u to $\|u\|$
	`Noise`	adds a band limited white noise to the input signal based on specified fault value [noise power]
	`Bias/Offset`	adds a predefined +ve or −ve offset (bias) value to input
	`Stuck-at`	the signal value stucks at the last correct value before fault occurrence
	`Time Delay`	introduces a delay of specified duration. During the delay, the value of the signal is the same as the last value observed before the time of fault occurrence.
	`Bit Flip`	Bitwise NOT operation on boolean signal
	`Package Drop`	replaces the input by the specified fault value
Block mutations	`ROR`	Relational Operator Replacement
	`LOR`	Logical Operator Replacement
	`S2P`	Sum to Product mutation
	`P2S`	Product to Sum mutation
	`ASR`	Arithmetic Sign Replacement (for 2-Input Sum Block)

License: CC BY 4.0

It is worth noting that FIM supports commonly used faults and mutation operators, based on existing literature. Specifically, fault types such as `Stuck-at`, `Package`

[1]The *fault_list* is a user-specified list detailing the desired faults/mutations.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

drop, Bias/Offset, Noise, Time delay, and Bit flips are adapted from ErrorSim [SMSJ17], MODIFI [SVET10], SIMULTATE [PRWN16], and FIBlock [FMMJ21]. Additionally, faults like Negate, Invert, and Absolute, along with Block mutation operations such as ROR, LOR, and P2S, are derived from SIMULTATE [PRWN16]. Standard arithmetic operator replacements like S2P and ASR are commonly used in mutation testing.

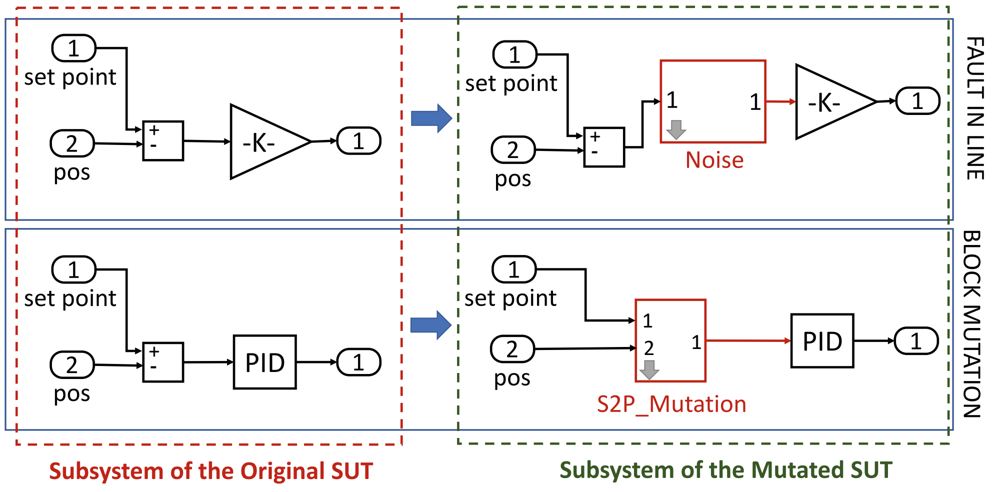


Figure 3.2: Illustration of some faults/mutations in a SUT (the injected fault blocks are highlighted in red).
License: CC BY 4.0

Each fault block in FIM is implemented as a masked subsystem featuring distinct block parameters. A flag named *FIEnableflag* is incorporated into each fault block, allowing users to activate or deactivate it as needed. Since FIM is designed to handle deterministic faults, all fault blocks include the parameter *FaultOccurenceTime* to specify when the fault should be triggered. Additionally, the block parameter *FaultEffect* dictates the duration for which the fault persists.

FIM offers two types of fault effects:

- *Infinite time*: This effect causes the fault block to produce erroneous output from the *FaultOccurenceTime* until the end of the simulation.
- *Constant time*: In this case, the fault occurs at the specified *FaultOccurenceTime* and persists for the duration specified by the block parameter *FaultDuration*.

Furthermore, fault blocks related to Noise, Bias/Offset, and Package Drop include an extra block parameter called *FaultValue*, which allows users to adjust the fault value. Similarly, blocks associated with ROR, LOR, and ASR have an additional parameter called *OperatorNum* to select the relevant operator. The correspondence between various operators and their respective numbers is detailed in Table 3.2.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Table 3.2: List of operators for block mutations.

Mutation	**Operator number : Corresponding Operation**
ROR	$1:>, 2:<, 3:<=, 4:>=, 5:==, 6:\neq$
LOR	1 : AND, 2 : OR, 3 : NAND, 4 : NOR, 5 : XOR, 6 : XNOR
ASR	$1:+-, 2:--, 3:++, 4:-+$

3.2.2 Fault Injection

The *Fault Injection Module* is tasked with injecting faults based on the specified mutations (*fault_list*). In the SINGLE-MODEL scenario, which involves mutations with a single model, FIM duplicates the SUT and injects all desired faults into the copied file. Conversely, in the MULTI-MODEL scenario, which entails mutations with multiple models, FIM duplicates the SUT and injects a single fault into each copied file. This process repeats until all desired faults are injected, resulting in multiple copies of the SUT with one fault injected per copy.

SINGLE-MODEL mode offers two main advantages: (i) it generates and compiles a single model, and (ii) it allows for the examination of the simultaneous effects of multiple faults within a single mutated SUT. On the other hand, MULTI-MODEL mode offers simplicity and smaller individual model sizes.

The *Fault Injection Module* leverages the fundamentals of programmatic editing of Simulink models to inject faults into the mutated SUT. It stores information such as line handles, block handles, and source-destination blocks (along with their respective ports) in a persistent cache for fault injection. During fault injection, it also preserves signal logging information and enhances the model layout to comply with modeling guidelines and improve model readability.

Once all desired faults are injected, the *Fault Injection Module* generates a `fault table` (.xlsx file) containing detailed information about the injected fault blocks. Each injected fault block is assigned a unique name, which is later used by the *Fault Configuration* component to control its activation.

3.2.3 Fault Configuration

Utilizing the generated `fault table`, users can now customize the faults to be utilized in an experiment. FIM enables users to input another configuration file, termed *fault_enable_list*, which is a table (in .csv or .xlsx format) containing fields to define the block(s) to be activated alongside the relevant fault parameters. Following the specifications outlined in the *fault_enable_list*, the *Fault Configuration* component activates the corresponding fault blocks within the mutated SUT and configures them based on the specified fault parameters. The fault block earmarked for activation is identified by its unique identifier (fault number) as listed in the `fault table`.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

3.3 Tool Usage

A specific requirement for FIM is that users must have a licensed installation of MATLAB™ and Simulink®. We have developed FIM in MATLAB™ R2020b, providing a command-line interface that enables users to execute the tool from the command prompt. In both SINGLE- and MULTI-MODEL modes, the *Fault Injection* and *Fault Configuration* tasks are managed by distinct customized functions. Essentially, user-defined configuration files and the output directory control the overall behavior of FIM.

For **fault injection**, FIM utilizes the functions `FISingle` and `FIMulti` for SINGLE-MODEL and MULTI-MODEL scenarios, respectively. Each of these functions requires two arguments: a *configuration file*, defining the set of faults to be injected, and an *output directory*, specifying where the output will be saved. Typically, the *configuration file* contains details about the SUT and the *fault_list*. The *fault_list* is a table that specifies the target fault location and the desired type of fault/mutation. The target fault location can be identified using the names of source, destination, or parent blocks, or by specifying the hierarchy levels of the SUT. This flexibility allows the user to control the fault injection space. Depending on the selected mode (SINGLE- or MULTI-MODEL), FIM generates either a single mutated model with all desired faults injected in one model copy or a set of mutated models, each with one fault injected. Additionally, FIM outputs the generated `fault table` to the user-specified output directory.

After fault injection, the user can review the generated `fault table` to identify which fault blocks to *activate* and *configure*. In the **fault configuration** process, FIM employs the functions `FCSingle` and `FCMulti` for SINGLE-MODEL and MULTI-MODEL cases, respectively. These functions include an additional argument, *fault_enable_list*, which outlines the faults to be activated in the subsequent run and their activation strategy. As output, the selected fault blocks are activated, and the corresponding fault parameters are configured upon executing the appropriate command based on the SINGLE-/MULTI-MODEL mode.

For a *step-by-step demonstration* of our tool, we present a case study based on the *Aircraft Elevator Control System* (AECS) from the avionics-aerospace domain [GM05]. This Simulink model comprises various hierarchical subsystems, such as the Controller, Plant, and Sensors. The model's input variable is the 'Pilot Command,' which governs the positions of the left and right actuators, resulting in two output variables. Additionally, the model includes various types of signals, including real-valued, Boolean, and enumerated (state machine) variables. Mutations in any of these signals have the potential to induce faulty behavior in the model.

Injecting faults. Through the following steps, we systematically inject desired faults into the AECS (our System Under Test, SUT) using our tool, resulting in a mutated SUT. In this example, we focus solely on the SINGLE-MODEL approach, where all mutants are accessible through a single Simulink model. Let us assume our input configuration file is named '`FIMConfigur_sfaircraft.txt`'.

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 3.3 provides an example *fault_list*. The *level_final* column specifies the hierarchy depth to which we restrict our fault injection. The *Src_or_InportName* column indicates that faults are injected into all lines whose Input port/Source is named "Src_or_InportName". The *Dst_or_OutportName* column indicates that all lines whose Output port/Destination is named "Dst_or_OutportName" will be mutated. The *ParentBlock* column specifies the name of the block to select all lines originating from or included within this *ParentBlock*. The *Faulttype_ft* column indicates the type of fault/mutation. Since block mutations (e.g., `ASR, S2P, P2S, ROR, LOR`) operate on blocks rather than lines (signals), we denote "NA" and do not specify any *Src_or_InportName*, *Dst_or_OutportName*, or *ParentBlock*.

Table 3.3: A sample *fault_list* for the AECS.

level_final	**Src_or _Inport Name**	**Dst_or _Outport Name**	**Parent Block**	**Fault type_ft**
Controller	NA	NA	Left Control Laws	Bias/ Offset
Plant/Actuators/Right Outer Hydraulic Actuator	NA	NA	NA	Negate
Plant/Actuators/Left Outer Hydraulic Actuator/Hydraulic Actuator	NA	NA	NA	ASR
Controller/Right Control Laws/IO Control Law	NA	NA	NA	S2P
Plant/Actuators/Right Inner Hydraulic Actuator/Hydraulic Actuator	NA	NA	NA	P2S
Controller/Right Control Laws/ Subsystem	NA	NA	NA	ROR

Let us now assume that we want to save the information of all the mutants in the folder '`Results`'. After setting up the configuration file, we start the fault injection in the MATLAB environment using the following command at the command prompt: `FISingle('FIMConfigur_sfaircraft.txt','Results')`. As soon as the command is entered, FIM initializes, loads the *FInjLib*, creates a copy of the original model and starts injecting faults in the copied file. Note that by default, all the injected faults are turned off. When the command has executed, the mutated SUT (single Simulink model with all mutants) is saved, and the generated `fault table` (see Table 3.4) is stored in the folder '`Results`'. In Table 3.4, Column *My_faulty_model* indicates the name of the mutated SUT. Column *My_faulty_block* indicates the unique name of the fault block added in the SUT. Column *Parentblock* indicates the subsystem within which the fault is injected. Column *Fault_Type* gives the name of the fault or block mutation operator. Columns *SRC_details* and *DST_details* respectively provide the full names of the source

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

and destination blocks corresponding to the line in which the fault is injected. Columns *SRC_port_number* and *DST_port_number* specify the respective port numbers of the source and destination. Such a detailed description of the injected faults will assist the user to identify the exact location of the fault block in the mutated SUT.

The first fault injected into the mutated SUT is `Bias/Offset` s.t. the fault block (Offset1) is placed between the lines connecting the source (pointed by *SRC_details* and *SRC_port_number*) to the corresponding destination. Note that the source/destination information for block mutations (e.g., `ASR`, `ROR`, etc.) are not given by the `fault table` in Table 3.4 since they are faults in blocks and not in signals (or lines). Some mutations (or injected faults) are shown in Figure 3.3.

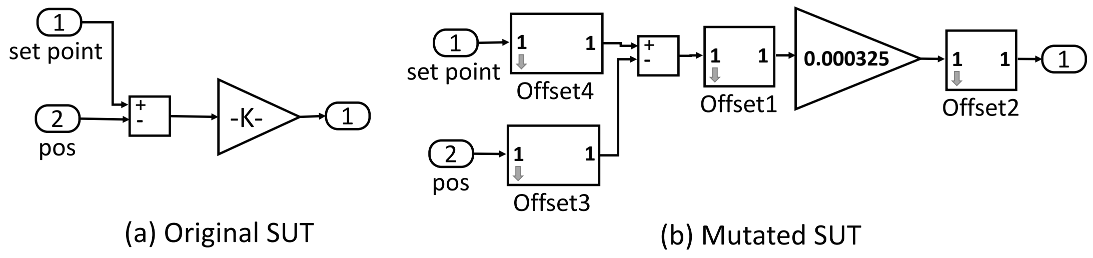


Figure 3.3: Mutations in the 'Controller/Left Control Laws/Direct Link Control Law' subsystem of AECS.

Configuring faults. After fault injection, we can look into the generated `fault table` to identify the fault blocks which we want to turn on. Let us assume that we want to turn on the blocks Offset1 and ArSignReplOperator38 and, configure them for desired fault parameters. We now define the *fault_enable_list* which is a table (see Table 3.5) with the following properties: Column *FaultBlock_Num* specifies the unique number of the fault block in accordance with the `fault table`. Column *Faultvalue_fv* indicates the amplitude of the fault (valid only for `Noise`, `Bias/Offset` and `Package Drop` faults; otherwise "NA"). Column *FaultOccurenceTime_fot* indicates the time of fault occurence. Column *FaultEffect_fe* indicates the fault effect (either 'Infinite time' or 'Constant time'). Column *FaultDuration_fd* specifies the fault persistence time (valid only for 'Constant time' fault effect; otherwise "NA"). Column *FaultOperatorNumber_fo* specifies the operator in case of the block mutations: `ROR`, `LOR` and `ASR`.

Let us assume that we save this *fault_enable_list* in '`Faultlist_Aircraft.csv`' file. The selected fault blocks are switched on and the respective fault parameters are configured after the execution of the following command:
`FCSingle('FIMConfigur_sfaircraft.txt','Results', 'Faultlist_Aircraft.csv')`.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 3.4: Some entries of the `fault table` (based on the *fault_list* in Table 3.3).

My_faulty_model	**My_faulty_block**	**Parentblock**	**Fault_Type**	**SRC_details**	**DST_details**	**SRC_port_number**	**DST_port_number**
Aircraft_copy	Offset1	Aircraft_copy/ Controller/ Left Control Laws/Direct Link Control Law	Bias/ Offset	Aircraft_copy/ Controller/ Left Control Laws/Direct Link Control Law/Sum	Aircraft_copy/ Controller/ Left Control Laws/Direct Link Control Law/Gain	1	1
...	...	...	...	...	...	...	...
Aircraft_copy	ArSignRepl Operator38	Aircraft_copy/Plant/ Actuators/LeftOuter Hydraulic Actuator/Hydraulic Actuator	ASR	-	-	-	-

Table 3.5: A sample *fault_enable_list* for the AECS.

FaultBlock_Num	**Faultvalue_fv**	**FaultOccurence Time_fot**	**FaultEffect_fe**	**FaultDuration_fd**	**FaultOperator Number_fo**
1	100	4	Infinite time	NA	NA
38	NA	2	Constant time	3	3

3.4 Tool Evaluation

To evaluate the performance of our tool, we conducted experiments on the AECS for both the SINGLE-MODEL and MULTI-MODEL modes. These experiments were carried out on a MacBook Pro equipped with an Apple M1 chip, 16 GB RAM, and macOS Big Sur running MATLAB™ R2020b.

The fault injection times, averaged over ten independent runs, for both scenarios are summarized in Table 3.6. The *Target* and *Fault type* columns indicate the target fault location and fault type, respectively. Column n displays the number of faults injected. Please note that (i) n is determined by the structure of the target location, and (ii) the fault generation time is influenced by both the fault type and the target location. Our results demonstrate that the fault injection process is faster and requires less computational resources in the SINGLE-MODEL scenario compared to the MULTI-MODEL case. In particular, with regards to the AECS model, FIM requires an average of 0.92 seconds for injecting a single fault in the SINGLE-MODEL mode, while in the MULTI-MODEL mode, it requires an average of 2.87 seconds for single fault injection.

In summary, based on our fault injection experiments and the resulting findings, we draw the following conclusions:

1. The MULTI-MODEL approach requires more resources, making SINGLE-MODEL the preferred option when efficiency is paramount.
2. However, while SINGLE-MODEL offers efficiency, it may present challenges in terms of model size and complexity, potentially complicating visualization, debugging, and analysis. In contrast, the MULTI-MODEL approach, by creating smaller models, often makes it easier to analyze fault injection results. Each model can be examined individually, which simplifies understanding the impact of individual faults on the system and improves the overall interpretability of the results.
3. In contrast to the MULTI-MODEL scenario, the SINGLE-MODEL setup is advantageous for investigating the interplay between multiple faults, which may potentially be activated at different points in time.
4. To leverage the advantages of both modes, users can initially utilize the SINGLE-MODEL approach by default. If specific faults warrant further investigation, users can switch to the MULTI-MODEL mode to analyze their individual effects, enhancing the interpretability and comprehensibility of the analysis.

3.5 Related Work

A multitude of fault injection tools and techniques are available, tailored to specific fault types, SUT characteristics, and injection methodologies [HTI97, KDN14, NCM16, SVS14, VS05, ZAV$^+$04]. Notably, numerous tools have been developed for fault injection in Simulink models [PRWN16, SVET10, SMSJ17, FMMJ21], albeit with certain limitations and drawbacks. For instance, MODIFI [SVET10] and ErrorSim [SMSJ17] provide

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 3.6: Fault Injection (mutant generation) time (in seconds) for AECS.

Target Location and Fault details		***n***	**Fault Injection time**	
Target	Fault type		SINGLE-MODEL	MULTI-MODEL
‘Controller’ subsystem with parent block ‘Left Control Laws’	Bias/Offset	20	16.38	53.53
Plant/Actuators/Right Outer Hydraulic Actuator	Negate	17	14.05	46.31
Plant/Actuators/Left Outer Hydraulic Actuator/Hydraulic Actuator	ASR	2	3.73	11.02
Controller/Right Control Laws/IO Control Law	S2P	1	1.83	3.58
Plant/Actuators/Right Inner Hydraulic Actuator/Hydraulic Actuator	P2S	3	2.58	7.92
Controller/Right Control Laws/Subsystem	ROR	2	3.05	6.99
Total		45	41.62	129.35

License: CC BY 4.0

limited options for fault types and are not publicly accessible. Meanwhile, SIMULTATE [PRWN16] offers an interactive user interface using Python and MATLAB, which, while user-friendly, may impede scalability for experiments requiring the injection of numerous faults. Additionally, another model-based fault injection method [FMMJ21] supports typical fault injections but lacks automated support for fault block placement within the SUT.

In contrast, FIM surpasses existing tools in several aspects: (i) it encompasses a broader range of faults and mutations (see Section 3.2), (ii) it facilitates automated fault injection through a seamless user interface directly within the MATLAB environment without requiring additional setup, and (iii) it boasts unparalleled scalability in fault injection, allowing experiment design through configuration files for the generation of a vast number of mutants within seconds.

3.6 Conclusion

We introduced FIM, an open-source toolkit designed for the automated injection of faults and generation of mutants in Simulink models. Testers can manage the activation of fault blocks according to their specific tasks and fine-tune fault parameters to thoroughly assess the SUT's fault tolerance. We believe FIM represents a significant advancement in the verification process for safety-critical CPSs. Verification experts and testers can utilize and customize our tool to meet their safety evaluation and testing requirements. Besides, our tool provides testers with the flexibility to update it by (i) modifying the automated scripts for fault injection and configuration, and (ii) expanding the fault library with additional operators, including conditional and cascaded faults.

Data Availability Statement. FIM is openly available at https://gitlab.com/DrishtiYadav/fimtool and is also available on the Zenodo repository [BMNY22b] via https://doi.org/10.5281/zenodo.6554973. Tool demonstration video is available at https://youtu.be/0EJri93Y_Gg. A detailed, step-by-step demonstration of FIM, including specific commands for two usage scenarios, is also available online.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

CHAPTER 4

Blood Coagulation Algorithm: A Global Optimizer

We present a new population-based bio-inspired metaheuristic optimization algorithm named the Blood Coagulation Algorithm (BCA). BCA is inspired by the process of blood coagulation in the human body, leveraging the cooperative behavior of thrombocytes and their efficient strategy of clot formation. These biological principles are modeled to enhance intensification and diversification within a search space.

A thorough comparison with various leading metaheuristic algorithms was conducted using a suite of 23 well-known benchmark functions, showcasing BCA's effectiveness. Our extensive evaluation covers performance and convergence behavior of BCA. The comparative analysis and statistical testing indicate that BCA delivers highly competitive and statistically significant results relative to other prominent metaheuristic algorithms. Additionally, experimental outcomes highlight BCA's consistent performance in high-dimensional search spaces. We further demonstrate BCA's practicality by applying it to solve several real-world engineering problems, including Falsification testing of CPSs.

4.1 Introduction

In recent years, researchers have developed numerous nature-inspired and bio-inspired optimization algorithms, drawing from the collective behaviors of organisms. These

©2021 Copyright held by the author. Parts Reproduced, Reused, Reprinted, with permission, from Drishti Yadav. 2021. Blood Coagulation Algorithm: A Novel Bio-Inspired metaHeuristic Algorithm for Global Optimization. *Mathematics 2021, 9(23), 3011.* https://doi.org/10.3390/math9233011.

algorithms effectively address complex optimization problems across diverse fields, including engineering design, digital image processing, networks and communications, power and energy management, data analysis, machine learning, robotics, and medical diagnosis [FROC+20]. Recently, there has been increased interest in the effective use of evolutionary and swarm intelligence-based algorithms. Their popularity stems from their simplicity, ease of implementation, flexibility, ability to avoid local optima, and gradient-free nature [BLS13].

Nature-inspired metaheuristic[1] algorithms address optimization problems by mimicking natural processes. These algorithms are classified into four categories: evolution-based, swarm-based, physics-based, and human behavior-based, as outlined in Table 4.1.

Population-based metaheuristic optimization algorithms, despite their diversity, share a fundamental characteristic: they all strive to balance diversification (exploration) and intensification (exploitation) [ČLM13].

1. **Diversification** maintains diversity and enables exploration of various promising regions within the search space. Utilizing randomized operators effectively aids in comprehensive and global exploration by applying random perturbations to the search agents (or design variables). Consequently, an efficient optimization algorithm should incorporate sufficient randomization to generate a larger number of solutions spread across the problem landscape during the initial stages of optimization. Without adequate diversification, an optimizer may prematurely converge to local optima.
2. **Intensification** demonstrates the ability to conduct a detailed local search within the promising regions identified during the diversification phase of the problem landscape (i.e., the search space). This characteristic focuses the search process on local areas, specifically around higher-quality solutions, rather than spanning broad areas of the search landscape. Without sufficient intensification, the optimizer may fail to achieve convergence.

Therefore, achieving an appropriate balance between diversification and intensification is crucial for any metaheuristic algorithm to find the global optimum. Without this balance, the algorithm risks getting trapped in local optima and converging prematurely.

Despite the abundance of existing metaheuristic approaches and their applications in various techno-scientific and industrial domains, the question remains: *Is there a need for new optimization algorithms*? The answer is *yes*. The effectiveness of metaheuristic techniques in solving optimization problems hinges on the balance between diversification and intensification [ČLM13]. Each metaheuristic approach employs a unique method to achieve this balance. According to the *No-Free-Lunch* (NFL) theorem [WM97], no single optimization algorithm is universally efficient for all types of problems. This theorem asserts that while a particular optimizer might excel across many problems, it may perform poorly on specific problems, making it ineffective overall. Essentially,

[1]The Greek prefix "*meta*" in "metaheuristics" signifies that these algorithms operate at a "higher level" compared to problem-specific heuristics. Metaheuristics are typically employed for problems lacking an adequate problem-specific solution algorithm [BLS13].

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 4.1: Nature-inspired metaheuristic algorithms.

Category	**Characteristics**	**Examples**
Evolution-based	Inspired by the concept of natural evolution; Emulate the biological evolutionary behaviors such as recombination, mutation, and selection	Genetic Algorithm (GA) [Hol92], Genetic Programming (GP) [Koz94], Differential Evolution (DE) [LS04], Biogeography Based Optimizer (BBO) [Sim08], and Evolutionary Strategy (ES) [Bac91].
Swarm-based	Emulate the social behaviors (e.g., decentralized, self- organized systems) of organisms living in swarms, flocks, or herds; Inspired by the collective natural behavior of living organisms e.g., hunting, food search, mating, etc.	Particle Swarm Optimization (PSO) [KE95],Grey Wolf Optimizer (GWO) [MML14], Whale Optimization Algorithm (WOA) [ML16], Butterfly Optimization Algorithm (BOA) [AS19], Harris Hawk Optimization (HHO) [HMF$^+$19], Cuckoo Search (CS) [GYA13], Moth-Flame optimization (MFO) [Mir15], Flower Pollination Algorithm (FPA) [YKH14], Firefly Algorithm (FA) [GYA11], Salp Swarm Optimization (SSA) [MGM$^+$17], Ant colony optimization (ACO) [Blu05], Krill Herd (KH) [GA12], Improved Elephant Herding Optimization (IEHO) [SSK22], Emperor penguin and Salp Swarm algorithm (ESA) [Dhi21], etc.
Physics-based	Inspired by the physical laws behind natural phenomena	Simulated Annealing (SA) [KGJV83], Gravitational Search Algorithm (GSA) [RNPS09], Black Hole (BH) algorithm [Hat13], Sine Cosine Algorithm (SCA) [Mir16b], Big-Bang Big-Crunch (BB-BC) optimization algorithm [EE06], Water Cycle Algorithm (WCA) [ESBH12], Artificial Electric Field Algorithm (AEFA) [Y$^+$19], etc.
Human behavior-based	Emulate some human behaviors; Inspired by the social behaviors of humans	Interior Search Algorithm (ISA) [Gan14], Mine Blast Algorithm (MBA) [SBEH13], Harmony search (HS) algorithm [GKL01], Imperialist Competitive Algorithm (ICA) [AGL07], Teaching-Learning Based Optimization (TLBO) [RSV11], Soccer League Competition (SLC) algorithm [MR14], Exchange Market Algorithm (EMA) [GB14], Socio Evolution and Learning Optimization Algorithm (SELO) [KKS18], etc.

License: CC BY 4.0

when considering all possible optimization problems, the average performance of all algorithms is equal. Therefore, the NFL theorem supports the continuous development of new metaheuristic optimizers to address a broader range of complex and unsolved problems. This rationale underpins the ongoing effort to create new and powerful nature- or bio-inspired metaheuristic algorithms and to enhance existing ones, either to solve current complex problems more efficiently or to tackle new challenges. This motivation drives our proposal of a novel optimizer to compete with existing algorithms.

More specifically, in the context of CPSs, optimization algorithms, particularly metaheuristics, are crucial for effective testing. These algorithms can systematically navigate the complex input spaces of CPSs, uncovering faults and ensuring system robustness. The development of new algorithms is essential to cater to the unique and varied challenges presented by CPSs, guided by the insights from the NFL theorem and the necessity to balance exploration and exploitation in the search process. By continually innovating and refining these algorithms, we can significantly improve the reliability and safety of CPSs in practical applications.

Contributions. We introduce *Blood Coagulation Algorithm* (BCA), a new bio-inspired metaheuristic optimization algorithm. To our knowledge, no previous study in this context is available in the optimization literature. The core concept of BCA is to mimic the cooperative behavior of thrombocytes during blood coagulation, which leads to hemostasis. Thrombocytes work together, moving towards the injury site to form a stable clot. BCA emulates this process to efficiently solve single-objective optimization problems and locate optima within a complex search space.

The efficiency of BCA is tested on a set of 23 mathematical benchmark functions, including unimodal, multimodal, and fixed-dimensional problems. Additionally, we assess BCA's performance on high-dimensional functions to demonstrate its robustness for larger problems. To further validate its effectiveness and applicability, we apply BCA to six standard engineering design optimization problems and test its efficiency in the falsification of CPSs. The results reveal that BCA is highly competitive with state-of-the-art algorithms, showcasing exceptional performance and optimization capabilities across all benchmark test functions and real-world engineering problems evaluated in this study.

To summarize, the main contributions of this chapter are as follows:

1. Introduction of the bio-inspired *Blood Coagulation Algorithm* (BCA), which mimics the behavior of thrombocytes during blood coagulation.
2. Implementation and evaluation of the BCA using 23 mathematical benchmark functions, demonstrating its competitiveness with state-of-the-art algorithms.
3. Validation of the BCA's effectiveness through application to real-world challenges, including six engineering design problems and a falsification testing problem on a CPS from the automotive domain.

Chapter Organization. Section 4.2 delves into the background and inspiration drawn from the human blood coagulation process, providing essential information along with

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

the mathematical model and computational procedures of the proposed BCA. Section 4.3 describes the testbed of 23 benchmark optimization functions used in this study and details the experimental setup employed for the optimization tasks. The results of these experiments are discussed in Section 4.4. In Section 4.5, the performance of BCA is evaluated using six standard engineering optimization problems. Additionally, Section 4.6 examines BCA's effectiveness in a real-world case study involving the falsification of CPSs. Finally, Section 4.7 concludes the chapter.

4.2 Blood Coagulation Algorithm

In this section, we begin by introducing the inspiration behind the proposed Blood Coagulation Algorithm (BCA). Next, we delve into the mathematical model detailing the intensification and diversification phases of the BCA. Notably, BCA is a population-based, derivative-free optimization method, making it suitable for solving a wide range of well-defined optimization problems.

4.2.1 Inspiration

The proposed BCA draws inspiration from the natural biological process of blood coagulation in the human body. Blood is vital for transporting essential nutrients and oxygen to cells and removing metabolic waste. It is primarily composed of blood cells and plasma. Blood cells include *erythrocytes* (red blood cells), *leukocytes* (white blood cells), and *thrombocytes* (platelets). Thrombocytes play a critical role in coagulation (clotting), the process that transforms blood from a liquid to a gel, resulting in the formation of a blood clot [Cha14]. This coagulation process is essential for hemostasis, which stops blood loss from a damaged vessel and initiates repair. Thus, thrombocytes are essential in hemostasis, sealing ruptured blood vessels and preventing further blood loss.

The blood coagulation mechanism involves the *stimulation*, *linkage* and *aggregation* of thrombocytes, along with the accumulation and maturation of fibrin [GBB20]. Hemostasis begins with vasoconstriction, where the blood vessel wall constricts to reduce blood flow to the injury site and minimize blood loss. Thrombocytes then adhere to the injured blood vessel, forming a soft plug. Subsequently, thrombocytes trigger the final stage of hemostasis: blood coagulation, illustrated in Figure 4.1.

Two distinct biological models have been proposed to explain hemostasis: the *Coagulation Cascade* model and the *Cell-Centric* model. The Coagulation Cascade model, developed in the mid-1960s, was the first widely accepted explanation of the coagulation process [DR64, Mac64]. However, it has significant limitations when compared to the physiological coagulation model and cannot adequately account for all phenomena related to in vivo hemostasis. In the early 2000s, the Cell-Centric model was introduced as an alternative [HMI01]. This model replaces the traditional "cascade" hypothesis, proposing that coagulation occurs on various cell surfaces in four phases: *initiation*, *amplification*, *propagation*, and *termination* [FSDC10]. The four phases of the Cell-Centric model,

TU Bibliothek Your knowledge hub WIEN
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

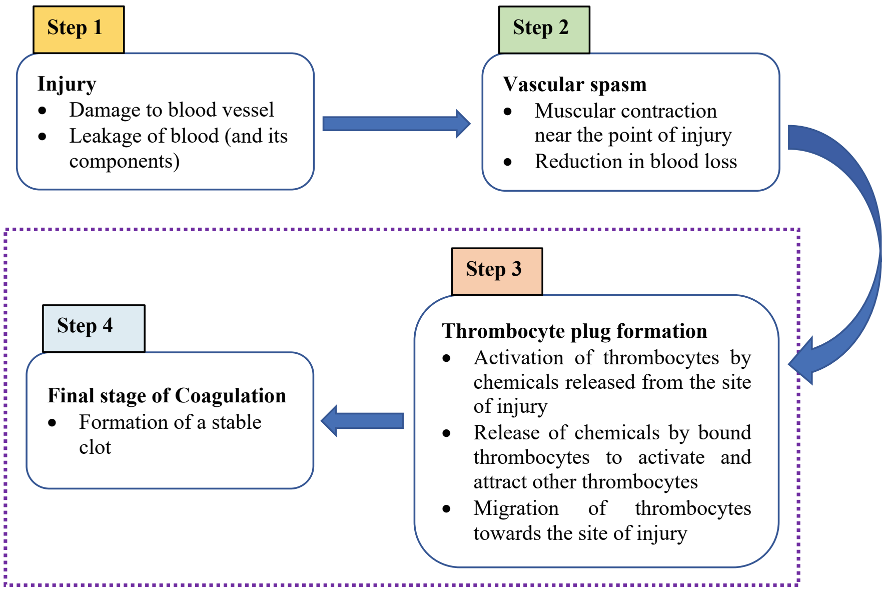

Step 1
Injury
• Damage to blood vessel
• Leakage of blood (and its components)
Step 2
Vascular spasm
• Muscular contraction near the point of injury
• Reduction in blood loss
Step 3
Thrombocyte plug formation
• Activation of thrombocytes by chemicals released from the site of injury
• Release of chemicals by bound thrombocytes to activate and attract other thrombocytes
• Migration of thrombocytes towards the site of injury
Step 4
Final stage of Coagulation
• Formation of a stable clot

Figure 4.1: Schematic of Hemostasis illustrating the general steps of blood clotting. License: CC BY 4.0

which form the foundation of the current theory of coagulation centered on cell surfaces, are outlined below [OL11]:

1. **Initiation phase**: The clotting process begins when tissue factor (TF) from subendothelial cells activates the production of small amounts of clotting factors, including thrombin, which is crucial for coagulation.

2. **Amplification phase**: Once sufficient procoagulant substances are produced, this phase extends the coagulation process from TF-bearing cells to thrombocytes. Thrombin from the initiation phase activates thrombocytes, causing them to adhere and start forming a clot.

3. **Propagation phase**: Activated thrombocytes bind with other essential clotting factors. These interactions, driven by a feedback mechanism, lead to increased thrombin production. This phase only occurs once a certain threshold of thrombin is reached [JB05].

4. **Termination phase**: The coagulation process concludes with the formation of a stable blood clot.

Our proposed algorithm, the Blood Coagulation Algorithm (BCA), is inspired by the cell-centric model of hemostasis for blood coagulation. This model involves the activation of thrombocytes and their migration to the injury site, guided by stochastic chemotactic mechanisms.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

4.2.2 Mathematical Model and Optimization Algorithm

We now outline the steps of the proposed BCA, detailing how the different phases of the blood coagulation process are mathematically modeled. We use a simple mapping to replicate the cell-centric model of hemostasis within the algorithm. Table 4.2 provides a description of the variables used in the mathematical formulation of BCA.

Table 4.2: Description of variables utilized in the mathematical formulation of BCA.

Variable(s)	**Description**
AR	Activation rate
θ	Threshold
P_f	Propagation factor
N_{Pop}	Population size
n	Number of dimensions/variables
$\boldsymbol{x}$	Position vector of the thrombocyte
$\boldsymbol{x}^*$	Position vector of the best thrombocyte obtained so far
$\boldsymbol{x_{rand}}$	Random position vector (a random thrombocyte) selected from the current population
d_{best}	Distance of a thrombocyte from the best thrombocyte
p_1	Uniformly distributed random number in the range [0, 1]
p_2	Uniformly distributed random number in the range [0, 1]
Max_iter	Maximum number of iterations
t	Current iteration
r_1	Uniformly distributed random number in the range [0, 1]

License: CC BY 4.0

4.2.2.1 Initialization phase

The BCA begins by defining the objective function and delineating the solution space. Additionally, the parameters of the BCA are initialized. The optimization problem is formulated in terms of an objective function $f(x)$ as follows:

$$\min_x f(\boldsymbol{x}) \qquad \boldsymbol{x} \in [\boldsymbol{LB}, \boldsymbol{UB}] \tag{4.1}$$

To address an optimization problem using population-based meta-heuristic techniques, the variables are structured as an array. In the BCA, this array is referred to as the *thrombocyte position* (analogous to the *chromosome* in Genetic Algorithms or the *particle position* in Particle Swarm Optimization). For an n-dimensional problem, the *thrombocyte position* is represented by a $1 \times n$ array, mathematically expressed as follows:

$$Thrombocyte\, position, \boldsymbol{x} = [x_1, x_2, x_3, ..., x_n] \tag{4.2}$$

It is important to note that each *thrombocyte position* x_i must lie within a specified range, $[LB_i, UB_i]$, where LB_i and UB_i represent the lower and upper bounds, respectively.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

During the initialization phase, a population of *thrombocyte positions* (i.e., solutions[2]) is generated, as shown in Equation (4.3). These positions are randomly generated (uniformly distributed) and are mathematically represented as a matrix of size $N_{\text{Pop}} \times n$, where the number of rows corresponds to the population size and the number of columns corresponds to the dimensions of the optimization problem. These dimensions are also referred to as the design, decision, or optimization variables.

$$Population\,of\,solutions = \begin{bmatrix} thrombocyte\,position_1 \\ thrombocyte\,position_2 \\ thrombocyte\,position_3 \\ \vdots \\ thrombocyte\,position_{N_{Pop}} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{N_{Pop}} & x_2^{N_{Pop}} & \dots & x_n^{N_{Pop}} \end{bmatrix} \tag{4.3}$$

The values for each decision variable $[x_1, x_2, x_3, \dots, x_n]$ can be represented as real numbers (floating-point values) for continuous problems, or as elements from a predefined set for discrete problems. The cost (or fitness) of a *thrombocyte position* is determined by evaluating the cost function, which is expressed as follows:

$$Cost_i = f(x_1^i, x_2^i, ..., x_n^i), \forall i = 1, 2, 3, ..., N_{Pop} \tag{4.4}$$

The thrombocyte position with the lowest cost (fitness) value is identified as the best solution, denoted as $\boldsymbol{x}^*$. This marks the end of the initialization phase and the beginning of the updating phase in the BCA. During this phase, the algorithm undertakes intensification and diversification tasks to pursue optimal solutions.

4.2.2.2 Updating phase

Figure 4.2 illustrates all the stages of the updating phase in the BCA. To update the positions of the thrombocytes, we utilize the Activation Rate (AR). After an injury, thrombocytes are activated by chemicals released at the injury site. If the thrombocytes are not fully activated, the coagulation process slows down. Once activated, thrombocytes update their positions either based on a randomly chosen thrombocyte or the best thrombocyte, which increases the likelihood of rapidly converging towards the global optimum. Once activated, thrombocytes tend to remain active throughout the coagulation process. Therefore, we use a lower AR value, specifically $AR = 0.1$, in this work. A uniformly distributed random number p_1 in the range $[0, 1]$ is used for comparison with the activation rate AR.

When $p_1 > AR$, the thrombocytes become activated and are prepared to undergo changes in their positions. The positions can be updated based on either diversification (exploration) or intensification (exploitation), as outlined below:

Diversification or Exploration. When the concentration of procoagulants exceeds the threshold (θ), rapid thrombin production occurs, facilitated by the migration of

[2]Throughout this chapter, the terms *solutions* and *thrombocyte positions* are used interchangeably.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

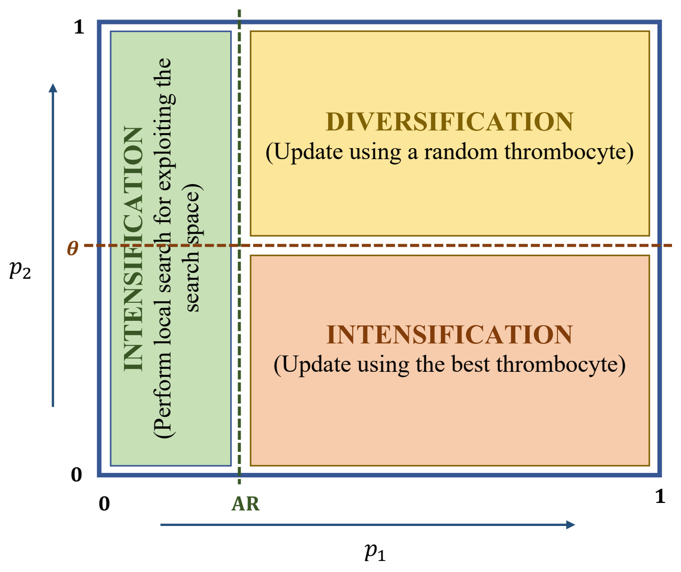

1
0
0
AR
1
p_2
p_1
θ
INTENSIFICATION
(Perform local search for exploiting the search space)
DIVERSIFICATION
(Update using a random thrombocyte)
INTENSIFICATION
(Update using the best thrombocyte)

Figure 4.2: Different updating phases of BCA.
License: CC BY 4.0

a large number of thrombocytes to the injury site. To compare with the threshold θ, we utilize a uniformly distributed random number p_2 in the range $[0, 1]$. When both $p_1 > AR$ and $p_2 > \theta$, the thrombocytes migrate and update their positions. The migration of thrombocytes occurs in a random manner, based on the positions of other thrombocytes. During the diversification phase, the position of a thrombocyte is updated using another thrombocyte randomly selected from the population. Therefore, the conditions $(p_1 > AR) \wedge (p_2 > \theta)$ emphasize diversification and enable the BCA to conduct a global search. The mathematical representation is as follows:

$$d = |C\boldsymbol{x_{rand}}(t) - \boldsymbol{x}(t)| \tag{4.5}$$

$$\boldsymbol{x}(t+1) = \boldsymbol{x_{rand}}(t) - P_f d \tag{4.6}$$

Please note that the values of $\boldsymbol{x}_{\text{rand}}(t)$ in Equation (4.5) and Equation (4.6) are identical; hence, there is no duplication in sampling. Here, C represents a coefficient arbitrarily set as $C = 2r_1$, where r_1 is a uniformly distributed random number in the range $[0, 1]$. We introduce the *Propagation factor* P_f, which serves as a scaling parameter controlling the step sizes of the random walks. This parameter governs the intensity of randomness in the BCA. To expedite overall convergence, the perturbation should gradually decrease. Therefore, the value of P_f is adaptively reduced at each iteration using the following reduction formulation:

$$P_f(t) = 2\left(1 - \frac{t}{Max\_iter}\right) \forall\, t = 1, 2, ..., Max\_iter \tag{4.7}$$

The parameters P_f and C are responsible for enhancing both diversification and intensification throughout the iterations.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

Intensification or Exploitation. When $(p_1 > AR) \wedge (p_2 \leq \theta)$, it highlights a local search and exploitation of the search space. Here, the best thrombocyte position is determined, and the remaining thrombocytes adjust their positions based on their distances from the best thrombocyte. This behavior is mathematically described by the following equations:

$$d_{best} = |\boldsymbol{x}^*(t) - \boldsymbol{x}(t)| \tag{4.8}$$

$$\boldsymbol{x}(t+1) = \boldsymbol{x}^*(t) - \boldsymbol{x'} \tag{4.9}$$

where $\boldsymbol{x'} = P_f \boldsymbol{x}(t) + C d_{best}$.

It is worth noting that we update $\boldsymbol{x}^*$ in each iteration whenever a superior solution is discovered. The most promising solution identified in each iteration is regarded as the best attained solution or the closest approximation to the optimum obtained thus far. Additionally, we observe (as shown in Equation (4.9)) that any thrombocyte can adjust its position within the vicinity of the current best thrombocyte (the best thrombocyte obtained thus far). Consequently, the BCA enables effective intensification (exploitation) of the search space. Furthermore, we assume a 50% probability of selecting either diversification or intensification to update the positions of the thrombocytes during optimization. Therefore, we set $\theta = 0.5$ in this study. We note that depending on the value of p_2, the BCA seamlessly transitions between diversification and intensification.

In cases where $p_1 \leq AR$, the thrombocytes are not yet fully activated and thus not prepared for the propagation phase (i.e., thrombin production). We assume that the thrombocytes are only partially activated, leading to the formation of a platelet plug responsible for primary hemostasis[3]. In this scenario, we update the positions of the thrombocytes by exploiting the current best thrombocyte to exploit the search space further. This is expressed mathematically as follows:

$$\boldsymbol{x}(t+1) = \boldsymbol{x}^*(t) - k P_f d' \tag{4.10}$$

where $d' = |C\boldsymbol{x}^*(t) - \boldsymbol{x}(t)|$ and $k = P_f(C-1)$.

4.2.2.3 Termination phase

The updating phase continues until the termination criteria, such as reaching a tolerance limit (where a specific error rate is achieved), completing the maximum number of iterations (Max_iter), observing no improvement in fitness after a predefined number of iterations, or another suitable condition, is met. In this study, we adopt reaching the maximum number of iterations (Max_iter) as the termination criterion. Upon completion of the updating phase, the termination phase begins, during which the termination criteria are assessed. The BCA outputs the best solution (i.e., thrombocyte position) and its corresponding fitness value.

The Blood Coagulation Algorithm: The three aforementioned phases constitute the entire framework of BCA, as demonstrated in Algorithm 4.1.

[3]Immediately following an injury, thrombocytes promptly create a plug at the injury site; this process is known as primary hemostasis.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Algorithm 4.1: The Blood Coagulation Algorithm (BCA)

```
Input  : Objective function
         $f(\boldsymbol{x}), \boldsymbol{x} = [x_1, x_2, x_3, ...., x_n], n =$ number of dimensions.
Output : the optimal solution $\boldsymbol{x}^*$
Initialize BCA parameters ($N_{Pop}$, Max_iter, $\theta$, $AR$)
Initialize randomly a population of solutions between $LB$ and $UB$
Calculate the cost (fitness) of initial solutions
$\boldsymbol{x}^*$ = The best thrombocyte position (i.e., solution)
while ($t < Max\_iter$) do
    for each thrombocyte $x_i$ do
        Update $P_f$ (i.e., Propagation factor), $C$, $p_1$, $p_2$
        if $p_1 > AR$ then
            if $p_2 > \theta$ then
                Select a random thrombocyte, $\boldsymbol{x_{rand}}$
                Update the position of the current thrombocyte: Equation (4.6)
            else
                if $p_2 \leq \theta$ then
                    Update the position of the current thrombocyte: Equation (4.9)
                end if
            end if
        else
            if $p_1 \leq AR$ then
                Update the position of the current thrombocyte: Equation (4.10)
            end if
        end if
    end for
    Check if any thrombocyte violates the boundary and adjust it
    Evaluate the fitness of each thrombocyte
    Update $\boldsymbol{x}^*$ if better solution is found
    Update iteration counter $t \leftarrow t + 1$
end while
return $\boldsymbol{x}^*$ ;        // the best thrombocyte position (solution)
```

4.3 Optimization Testbed and Experimental Platform

4.3.1 Benchmark Set

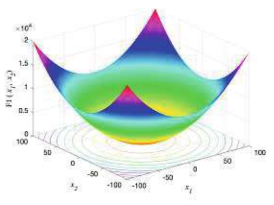
To assess the effectiveness and adaptability of the proposed BCA optimizer, we evaluate its performance on a selection of mathematical functions with known global optima. To achieve this, we utilize a collection of widely studied benchmark functions from existing literature as our optimization test bed. This comprehensive set of benchmark functions comprises two main types:

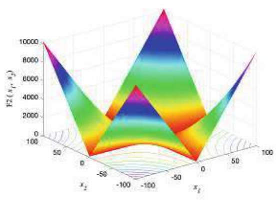
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

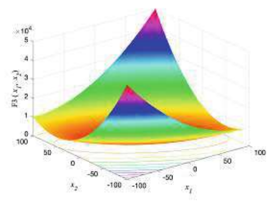
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

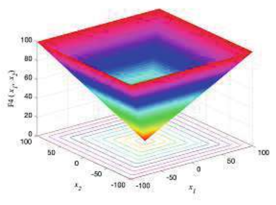
1. *Unimodal benchmark functions*: Unimodal functions possess a singular global optimum. Due to their lack of local optima, they primarily demonstrate intensification (exploitation). These functions are particularly useful for examining the convergence characteristics and the intensity of exploitation exhibited by the proposed algorithm. The unimodal functions (f_1 to f_7) investigated in this study are enumerated in Table 4.3, along with their mathematical formulations.
2. *Multimodal benchmark functions*: Multimodal functions often contain numerous local optima. Consequently, an optimizer must possess the capacity to navigate away from local optima to reach the global optimum. Thus, these functions serve to reveal the diversification (or exploratory) capabilities of BCA and its ability for escaping local optima. The multimodal functions (f_8 to f_{13}), along with their specifications, are outlined in Table 4.3. Additionally, it is noteworthy that fixed-dimension multimodal functions (f_{14} to f_{23}) are also included in this investigation.

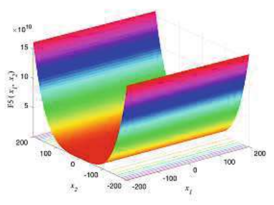
The search landscapes of the objective functions, including unimodal (f_1 to f_7), multimodal (f_8 to f_{13}), and fixed-dimension multimodal (f_{14} to f_{23}) functions, are depicted in Figure 4.3, Figure 4.4, and Figure 4.5, respectively.

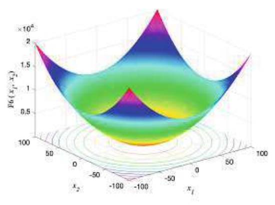

(a) f_1


(b) f_2


(c) f_3


(d) f_4


(e) f_5


(f) f_6

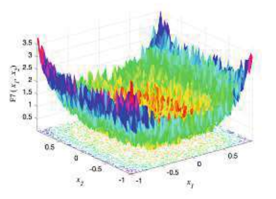

(g) f_7

Figure 4.3: Search space of unimodal functions (2-D view). (License: CC BY 4.0)

4.3.2 Experimental Setup

The proposed algorithm is implemented using the MATLAB® (R2020b) programming platform. Our experiments are conducted on a system equipped with an Apple M1 chip and 16 GB of RAM, running macOS Big Sur. In each experiment, we use a termination criterion for BCA based on the maximum number of iterations (Max_iter), set to 1000. Additionally, a population size of 30 is selected. The activation rate is set to 0.1, and the threshold is set to 0.5. To ensure meaningful statistical analysis and minimize statistical errors, we conduct 30 independent runs for each function. Furthermore, we record both

Table 4.3: List of Benchmark functions.

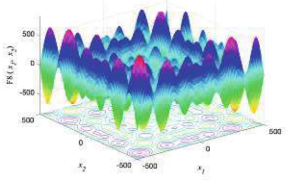
	Mathematical description	**D**	**Range**	f_{min}
Uni-modal	$f_1(x) = \sum_{i=1}^{n} x_1^2$	30	[-100, 100]	0
	$f_2(x) = \sum_{i=1}^{n} \|x_i\| + \prod_{i=1}^{n} \|x_i\|$	30	[-10, 10]	0
	$f_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30	[-100, 100]	0
	$f_4(x) = \max_i \{\|x_i\|, 1 \leq i \leq n\}$	30	[-100, 100]	0
	$f_5(x) = \sum_{i=1}^{n} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30, 30]	0
	$f_6(x) = \sum_{i=1}^{n} [x_i + 0.5])^2$	30	[-100, 100]	0
	$f_7(x) = \sum_{i=1}^{n} ix_i^4 + random[0, 1)$	30	[-1.28, 1.28]	0
Multi-modal	$f_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\|x_i\|})$	30	[500, 500]	$-418.982 \times n$
	$f_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12, 5.12]	0
	$f_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n} \cos 2\pi x_i) + 20 + e$	30	[-32, 32]	0
	$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_1}{\sqrt{i}}\right) + 1$	30	[-600, 600]	0
	$f_{12}(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ where $y_i = 1 + \frac{x_i+1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a < x_i < a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	30	[-50, 50]	0
	$f_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[-50, 50]	0
Fixed-dimension Multimodal	$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	2	[-65, 65]	1
	$f_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	[-5, 5]	0.00030
	$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	-1.0316
	$f_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	[-5, 5]	0.398
	$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)][30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2, 2]	3
	$f_{19}(x) = -\sum_{i=1}^{4} c_i \exp \sum_{j=1}^{3} [a_{ij}(x_j - p_{ij})^2]$	3	[1, 3]	-3.86
	$f_{20}(x) = -\sum_{i=1}^{4} c_i \exp \sum_{j=1}^{6} [a_{ij}(x_j - p_{ij})^2]$	6	[0, 1]	-3.32
	$f_{21}(x) = -\sum_{i=1}^{5} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.1532
	$f_{22}(x) = -\sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.4028
	$f_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.5363

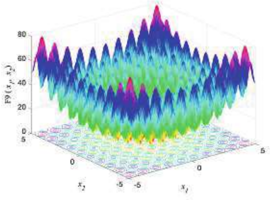
D = the dimensionality of the search space (i.e., number of variables);

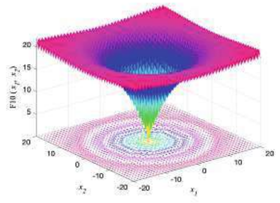
Range = the boundary of the search space [Lower Bound, Upper Bound]; f_{min} = global optimum

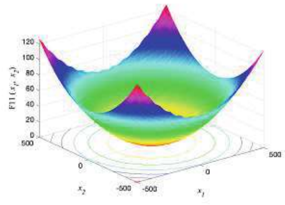
License: CC BY 4.0

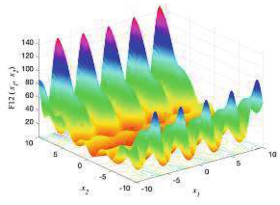
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.


(a) f_8


(b) f_9


(c) f_{10}


(d) f_{11}


(e) f_{12}

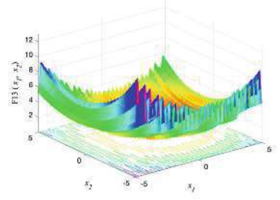
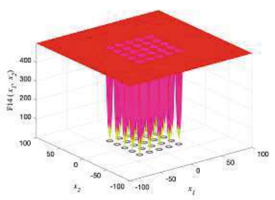
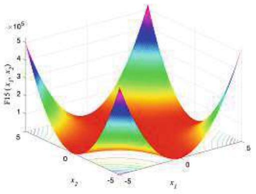
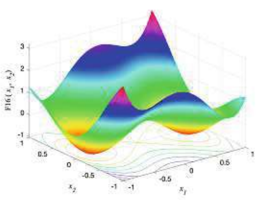
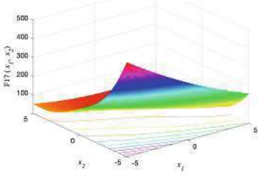

(f) f_{13}

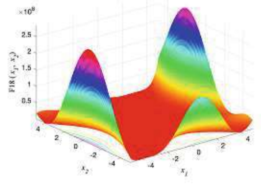
Figure 4.4: Search space of multimodal functions (2-D view). (License: CC BY 4.0)

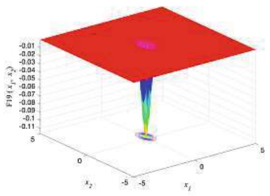

(a) f_{14}

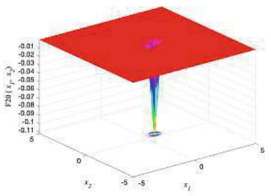

(b) f_{15}

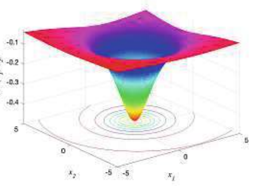

(c) f_{16}

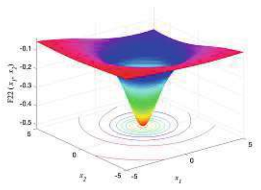

(d) f_{17}


(e) f_{18}


(f) f_{19}


(g) f_{20}


(h) f_{21}


(i) f_{22}

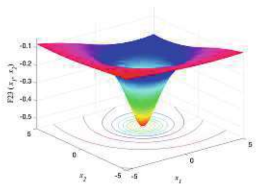

(j) f_{23}

Figure 4.5: Search space of the fixed-dimension multimodal benchmark functions (2-D view). (License: CC BY 4.0)

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

the mean (average fitness value) and the standard deviation (SD) of BCA for each benchmark function.

To demonstrate the effectiveness of the proposed BCA on the benchmark test functions, we conduct a comparative analysis with other established nature-inspired meta-heuristic algorithms. To accomplish this, we assemble a suite comprising 12 state-of-the-art optimizers, as outlined in Table 4.4. The parameter settings for these optimizers are also detailed in Table 4.4. To achieve optimal performance, we configure the parameters according to the values reported in the literature. We assume that the parameters selected for the other methods (as listed in Table 4.4) represent the most suitable choices for the optimization task. To ensure a fair comparison, we set the population size and the maximum number of iterations for each optimizer to 30 and 1000, respectively.

Moreover, to identify significant differences in the results yielded by different optimizers, we additionally conduct the non-parametric Wilcoxon statistical test [DGMH11] with a significance level of 5%. This test complements our experimental simulations and basic statistical analysis.

4.4 Experimental Results and Discussion

For every benchmark function, we conducted 30 simulations for each algorithm, encompassing both BCA and the other algorithms enumerated in Table 4.4, using randomly generated populations. The statistical[4] results, including the mean and standard deviation, are presented in Table 4.5 and Table 4.6.

4.4.1 Intensification and Diversification Capabilities of BCA

The assessment of BCA's intensification (exploitation) capability is conducted using the unimodal benchmark functions (f_1 to f_7). Furthermore, to gauge BCA's effectiveness in exploring diverse regions of the search space (diversification), we employ multimodal benchmark functions (f_8 to f_{23}), which exhibit numerous local optima. The statistical findings, as indicated by the evaluation metrics (mean and SD) presented in Tables 4.5 and 4.6, underscore BCA's competitiveness against other meta-heuristic algorithms. It is noteworthy that BCA outperforms many algorithms across most functions, highlighting its superior performance.

The findings presented in Tables 4.5 and 4.6 demonstrate that the proposed BCA consistently outperforms its competitors across f_1 to f_5 (unimodal test functions) and f_8 to f_{13} (multimodal test functions). Notably, BCA exhibits significantly superior results compared to its counterparts in handling 84.6% of the functions f_1 to f_{13} over 30 dimensions. This underscores the superior performance of BCA relative to other algorithms. Moreover, for the fixed-dimensional multimodal benchmark function f_{14}, the results obtained with BCA are superior and competitive. BCA delivers superior results for

[4]Throughout this chapter, numbers in the form n_1E$\pm n_2$ represent $n_1 \times 10^{\pm n_2}$.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Table 4.4: Parameter settings of various algorithms.

Algorithm	**Parameter**	**Value**
(All algorithms)	Population size	30
	Maximum number of iterations	1000
DE [SP97]	Scaling factor	0.5
	Crossover probability	0.5
PSO [KE95]	Topology	fully connected
	Inertia factor	Linear reduction from 0.9 to 0.1
	c_1	2
	c_2	2
	Velocity limit	10% of dimension range
GA [BDT99]	Type	Real coded
	Selection	Roulette wheel (Proportionate)
	Crossover	Probability 0.8
	Mutation	Gaussian (Probability 0.05)
CS [GYA13]	Discovery rate of alien solutions (p_a)	0.25
GWO [MML14]	Convergence parameter (a)	Linear reduction from 2 to 0
MFO [Mir15]	Convergence constant (a)	[-2 -1]
	Spiral factor (b)	1
FPA [YKH14]	Probability switch (p)	0.8
FA [GYA11]	α	0.5
	β	0.2
	γ	1
BAT [YHG12]	Q_{min} (Minimum frequency)	0
	Q_{max} (Maximum frequency)	2
	A (Loudness)	0.5
	r (Pulse rate)	0.5
GSA [RNPS09]	Gravitational constant	100
	Alpha coefficient	20
AOA [ADM$^+$21]	α	5
	μ	0.5
BBO [Sim08]	Habitat modification probability	1
	Immigration probability limits	[0, 1]
	Step size	1
	Max immigration (I) and Max emigration (E)	1
	Mutation probability	0.005

License: CC BY 4.0

f_{15}, as evident from Tables 4.5 and 4.6. The competitiveness of the results for functions f_{16} to f_{19} is notable, with most approaches yielding favorable results. Consequently, based on the results presented in Tables 4.5 and 4.6, the proposed BCA consistently attains the highest quality solutions for f_{14} to f_{23} (fixed-dimension multimodal) test

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Table 4.5: Comparison of the results obtained by BCA with other state-of-the-art meta-heuristic algorithms. The best results are kept in bold.

F	Metric	BCA	DE	PSO	GA	CS	GWO	MFO
f_1	Mean	**9.80E-28**	1.33E-03	1.83E+04	1.03E+03	9.06E-04	1.18E-27	1.01E+03
	SD	1.70E-22	5.92E-04	3.01E+03	5.79E+02	4.55E-04	1.47E-27	3.05E+03
f_2	Mean	**8.91E-17**	6.83E-03	3.58E+02	2.47E+01	1.49E-01	9.71E-17	3.19E+01
	SD	1.36E-11	2.06E-03	1.35E+03	5.68E+00	2.79E-02	5.60E-17	2.06E+01
f_3	Mean	**7.63E-16**	3.97E+04	4.05E+04	2.65E+04	2.10E-01	5.12E-05	2.43E+04
	SD	2.46E-15	5.37E+03	8.21E+03	3.44E+03	5.69E-02	2.03E-04	1.41E+04
f_4	Mean	**7.01E-10**	1.15E+01	4.39E+01	5.17E+01	9.65E-02	1.24E-06	7.00E+01
	SD	1.77E-09	2.37E+00	3.64E+00	1.05E+01	1.94E-02	1.94E-06	7.06E+00
f_5	Mean	**3.72E+00**	1.06E+02	1.96E+07	1.95E+04	2.76E+01	2.70E+01	7.35E+03
	SD	8.72E+00	1.01E+02	6.25E+06	1.31E+04	4.51E-01	7.78E-01	2.26E+04
f_6	Mean	1.36E-02	1.44E-03	1.87E+04	9.01E+02	3.13E-03	8.44E-01	2.68E+03
	SD	9.83E-03	5.38E-04	2.92E+03	2.84E+02	1.30E-03	3.18E-01	5.84E+031
f_7	Mean	3.95E-04	5.24E-02	1.07E+01	1.91E-01	7.29E-02	1.70E-03	4.50E+00
	SD	4.78E-04	1.37E-02	3.05E+00	1.50E-01	2.21E-02	1.06E-03	9.21E+00
f_8	Mean	**-1.25E+04**	-6.82E+03	-3.86E+03	-1.26E+04	-5.19E+19	-5.97E+03	-8.48E+03
	SD	4.30E+01	3.94E+02	2.49E+02	4.51E+00	1.76E+20	7.10E+02	7.98E+02
f_9	Mean	**1.70E-14**	1.58E+02	2.87E+02	9.04E+00	1.51E+01	2.19E+00	1.59E+02
	SD	3.04E-14	1.17E+01	1.95E+01	4.58E+00	1.25E+00	3.69E+00	3.21E+01
f_{10}	Mean	**8.88E-16**	1.21E-02	1.75E+01	1.36E+01	3.29E-02	1.03E-13	1.74E+01
	SD	2.89E-11	3.30E-03	3.67E-01	1.51E+00	7.93E-03	1.70E-14	4.95E+00
f_{11}	Mean	**3.49E-15**	3.52E-02	1.70E+02	1.01E+01	4.29E-05	4.76E-03	3.10E+01
	SD	1.86E-14	7.20E-02	3.17E+01	2.43E+00	2.00E-05	8.57E-03	5.94E+01
f_{12}	Mean	**4.19E-06**	2.25E-03	1.51E+07	4.77E+00	5.57E-05	4.83E-02	2.46E+02
	SD	5.43E-04	1.70E-03	9.88E+06	1.56E+00	4.96E-05	2.12E-02	1.21E+03
f_{13}	Mean	**1.99E-03**	9.12E-03	5.73E+07	1.52E+01	8.19E-03	5.96E-01	2.73E+07
	SD	5.37E-03	1.16E-02	2.68E+07	4.52E+00	6.74E-03	2.23E-01	1.04E+08
f_{14}	Mean	**9.98E-01**	1.23E+00	1.39E+00	**9.98E-01**	1.27E+01	4.17E+00	2.74E+00
	SD	3.02E-10	9.23E-01	4.60E-01	4.52E-16	1.81E-15	3.61E+00	1.82E+00
f_{15}	Mean	**3.07E-04**	5.63E-04	1.61E-03	3.33E-02	3.13E-04	6.24E-03	2.35E-03
	SD	4.59E-04	2.81E-04	4.60E-04	2.70E-02	2.99E-05	1.25E-02	4.92E-03
f_{16}	Mean	**-1.03E+00**	**-1.03E+00**	**-1.03E+00**	-3.78E-01	**-1.03E+00**	**-1.03E+00**	**-1.03E+00**
	SD	6.32E-06	6.78E-16	2.95E-03	3.42E-01	6.78E-16	6.78E-16	6.78E-16
f_{17}	Mean	**3.98E-01**	**3.98E-01**	4.00E-01	5.24E-01	**3.98E-01**	**3.98E-01**	**3.98E-01**
	SD	9.31E-05	1.69E-16	1.39E-03	6.06E-02	1.69E-16	1.69E-16	1.69E-16
f_{18}	Mean	**3.00E+00**	**3.00E+00**	3.10E+00	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**
	SD	1.28E-04	0.00E+00	7.60E-02	0.00E+00	0.00E+00	4.07E-05	0.00E+00
f_{19}	Mean	**-3.86E+00**	**-3.86E+00**	**-3.86E+00**	-3.42E+00	**-3.86E+00**	**-3.86E+00**	**-3.86E+00**
	SD	3.69E-03	3.16E-15	1.24E-03	3.03E-01	3.16E-15	3.14E-03	1.44E-03
f_{20}	Mean	**-3.32E+00**	-3.27E+00	-3.11E+00	-1.61E+00	-3.32E+00	-3.25E+00	-3.23E+00
	SD	3.33E-01	5.89E-02	2.91E-02	4.60E-01	1.77E-15	6.43E-02	6.42E-02
f_{21}	Mean	**-1.01E+00**	-9.64E+00	-4.14E+00	-6.66E+00	-5.05E+00	-8.64E+00	-6.88E+00
	SD	2.00E+00	1.51E+00	9.19E-01	3.73E+00	1.77E-15	2.56E+00	3.18E+00
f_{22}	Mean	**-1.04E+00**	-9.74E+00	-6.01E+00	-5.58E+00	-5.08E+00	-1.04E+00	-8.26E+00
	SD	2.51E-04	1.98E+00	1.96E+00	2.60E+00	8.88E-16	6.78E-04	3.07E+00
f_{23}	Mean	**-1.05E+01**	**-1.05E+01**	-4.72E+00	-4.69E+00	-5.12E+00	-1.08E+01	-7.65E+00
	SD	2.93E-07	8.88E-15	1.74E+00	3.25E+00	1.77E-15	1.72E+00	3.57E+00

License: CC BY 4.0

functions compared to alternative algorithms.

To gain deeper insights into the intensification and diversification capabilities of BCA, we track the search history of thrombocytes. Figure 4.6 showcases the search history of thrombocytes across some benchmark functions. Notably, we visualize the 2D versions of these functions to illustrate the positions of thrombocytes over successive iterations.

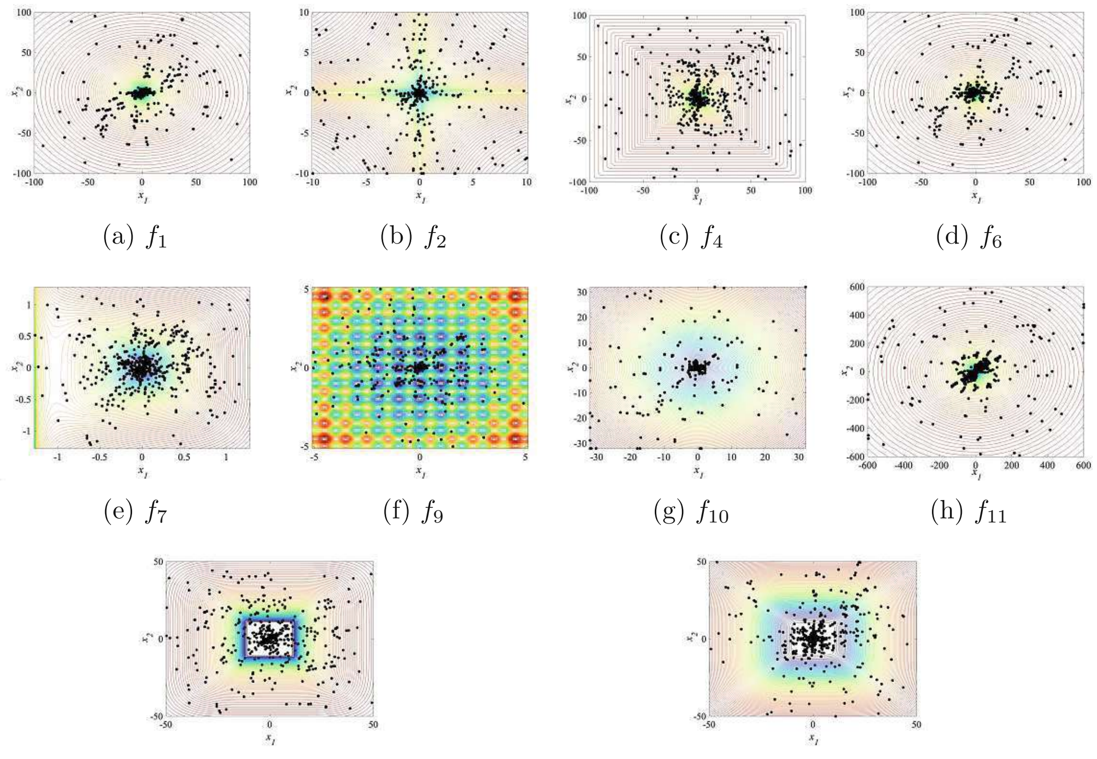
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 4.6: Results obtained with other state-of-the-art meta-heuristic algorithms. The best results are kept in bold. (Additional results of Table 4.5)

F	**Metric**	**FPA**	**FA**	**BAT**	**GSA**	**AOA**	**BBO**
f_1	Mean	2.01E+03	7.11E-03	6.59E+04	6.08E+02	6.67E-07	7.59E+01
	SD	5.60E+02	3.21E-03	7.51E+03	4.64E+02	7.45E-07	2.75E+01
f_2	Mean	3.22E+01	4.34E-01	2.71E+08	2.27E+01	0.00E-00	1.36E-03
	SD	5.55E+00	1.84E-01	1.30E+09	3.36E+00	0.00E-00	7.45E-03
f_3	Mean	1.41E+03	1.66E+03	1.38E+05	1.35E+05	6.87E-06	1.21E+04
	SD	5.59E+02	6.72E+02	4.72E+04	4.86E+04	6.87E-06	2.69E+03
f_4	Mean	2.38E+01	1.11E-01	8.51E+01	7.87E+01	1.40E-03	3.02E+01
	SD	2.77E+00	4.75E-02	2.95E+00	2.81E+00	1.90E-03	4.39E+00
f_5	Mean	3.17E+05	7.97E+01	2.10E+08	7.41E+02	2.49E+01	1.82E+03
	SD	1.75E+05	7.39E+01	4.17E+07	7.81E+02	3.64E-01	9.40E+02
f_6	Mean	1.70E+03	6.94E-03	6.69E+04	3.08E+03	**3.47E-04**	6.71E+01
	SD	3.13E+02	3.61E-03	5.87E+03	8.98E+02	3.47E-04	2.20E+01
f_7	Mean	3.41E-01	6.62E-02	4.57E+01	1.12E-01	**3.92E-06**	2.91E-03
	SD	1.10E-01	4.23E-02	7.82E+00	3.76E-02	3.92E-06	1.83E-03
f_8	Mean	-6.45E+03	-5.85E+03	-2.33E+03	-2.35E+03	-1.22E+04	-1.24E+04
	SD	3.03E+02	1.16E+03	2.96E+02	3.82E+02	1.22E+03	3.50E+01
f_9	Mean	1.82E+02	3.82E+01	1.92E+02	3.10E+01	3.42E-07	0.01E+00
	SD	1.24E+01	1.12E+01	3.56E+01	1.36E+01	3.42E-07	0.00E+00
f_{10}	Mean	7.14E+00	4.58E-02	1.92E+01	3.74E+00	**8.88E-16**	2.13E+00
	SD	1.08E+00	1.20E-02	2.43E-01	1.71E-01	8.88E-16	3.53E-01
f_{11}	Mean	1.73E+01	4.23E-03	6.01E+02	4.86E-01	1.00E-06	1.46E+00
	SD	3.63E+00	1.29E-03	5.50E+01	4.97E-02	1.21E+06	1.69E-01
f_{12}	Mean	3.05E+02	3.13E-04	4.71E+08	4.63E-01	4.28E-06	6.68E-01
	SD	1.04E+03	1.76E-04	1.54E+08	1.37E-01	4.28E-06	2.62E-01
f_{13}	Mean	9.59E+04	2.08E-03	9.40E+08	7.61E+00	3.10E-01	1.82E+00
	SD	1.46E+05	9.62E-04	1.67E+08	1.22E+00	3.10E-01	3.41E-01
f_{14}	Mean	**9.98E-01**	3.51E+00	1.27E+01	**9.98E-01**	**9.98E-01**	**9.98E-01**
	SD	2.00E-04	2.16E+00	6.96E+00	4.52E-16	5.54E-01	4.52E-16
f_{15}	Mean	6.88E-04	1.01E-03	3.00E-02	1.03E-03	3.12E-04	1.66E-02
	SD	1.55E-04	4.01E-04	3.33E-02	3.66E-03	2.64E-04	8.60E-03
f_{16}	Mean	**-1.03E+00**	**-1.03E+00**	-6.87E-01	**-1.03E+00**	**-1.03E+00**	-8.30E-01
	SD	6.78E-16	6.78E-16	8.18E-01	6.78E-16	5.48E-05	3.16E-01
f_{17}	Mean	**3.98E-01**	**3.98E-01**	**3.98E-01**	**3.98E-01**	**3.98E-01**	5.49E-01
	SD	1.69E-16	1.69E-16	1.58E-03	1.69E-16	2.54E-06	6.05E-02
f_{18}	Mean	**3.00E+00**	**3.00E+00**	1.47E+01	**3.00E+00**	**3.00E+00**	**3.00E+00**
	SD	0.00E+00	0.00E+00	2.21E+01	0.00E+00	1.00E-02	0.00E+00
f_{19}	Mean	**-3.86E+00**	**-3.86E+00**	**-3.84E+00**	**-3.86E+00**	**-3.86E+00**	**-3.78E+00**
	SD	3.16E-15	3.16E-15	1.41E-01	3.16E-15	4.29E-04	1.26E-01
f_{20}	Mean	-3.29E+00	-3.28E+00	-3.25E+00	-3.24E+00	**-3.32E+00**	-2.70E+00
	SD	1.95E-02	6.36E-02	5.89E-02	1.51E-01	1.25E+01	3.57E-01
f_{21}	Mean	-5.21E+00	-7.67E+00	-4.26E+00	-8.65E+00	-8.85E+00	-8.31E+00
	SD	8.15E-02	3.50E+00	2.55E+00	1.77E+00	1.25E+00	2.88E+00
f_{22}	Mean	-5.34E+00	-9.63E+00	-5.60E+00	-1.02E+01	-1.04E+01	-9.38E+00
	SD	5.36E-02	2.29E+00	3.02E+00	7.27E-03	2.21E+00	2.59E+00
f_{23}	Mean	-5.29E+00	-9.75E+00	-3.97E+00	-1.01E+01	**-1.05E+01**	-6.23E+00
	SD	3.56E-01	2.34E+00	3.00E+00	1.70E+00	1.02E+00	3.78E+00

License: CC BY 4.0

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.



(a) f_1 (b) f_2 (c) f_4 (d) f_6

(e) f_7 (f) f_9 (g) f_{10} (h) f_{11}

(i) f_{12} (j) f_{13}

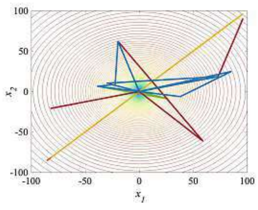
Figure 4.6: Illustration of the search history of thrombocytes. For visualization, we consider 2D version of the benchmark functions.
License: CC BY 4.0

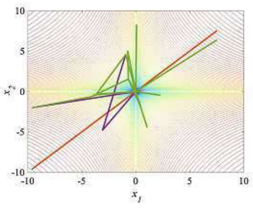
Analyzing Figure 4.6, we observe that the distribution of thrombocytes demonstrates BCA's capability in exploring diverse regions of the search space. Moreover, it effectively exploits the vicinity of favorable areas to converge towards global optimal solutions. The convergence of thrombocytes towards global optima underscores BCA's ability to escape local optima effectively. Additionally, we examine the trajectories of randomly selected thrombocytes (2D visualization) to further elucidate BCA's diversification potential (Figure 4.7). Different colors denote distinct thrombocyte trajectories, revealing their exploration across various regions of the search space. These zig-zag patterns in trajectories signify the diverse searching behavior of thrombocytes. Furthermore, these trajectories terminate in the vicinity of global optima, affirming BCA's capability to navigate away from local optima, particularly in the case of multimodal functions.

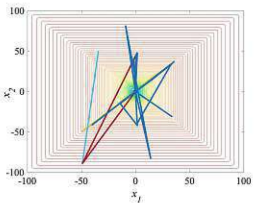
4.4.2 Convergence Analysis

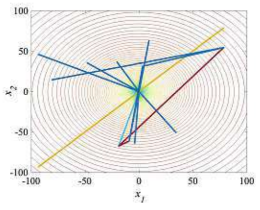
To comprehend the convergence behavior and assess the efficacy of an optimization algorithm, it is vital to achieve a balance between intensification and diversification, mitigating the exploration/exploitation trade-off and avoiding premature convergence

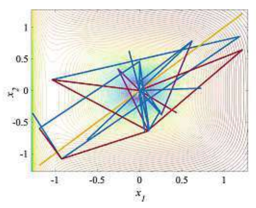
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

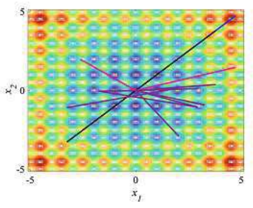

(a) f_1

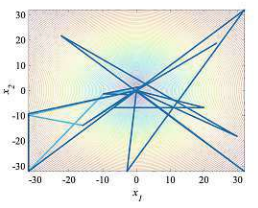

(b) f_2

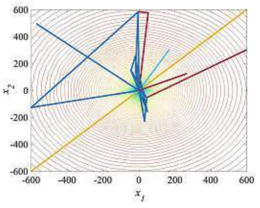

(c) f_4

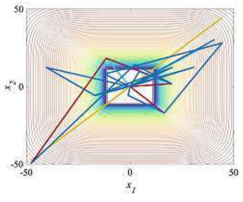

(d) f_6


(e) f_7


(f) f_9


(g) f_{10}


(h) f_{11}


(i) f_{12}

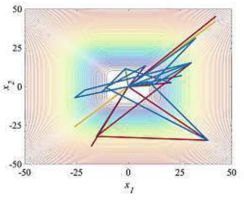
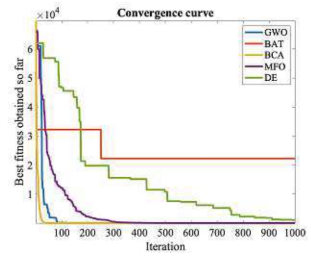

(j) f_{13}

Figure 4.7: Trajectories of some randomly chosen thrombocytes for some of the benchmark functions (We consider 2D version of the benchmark functions).
License: CC BY 4.0

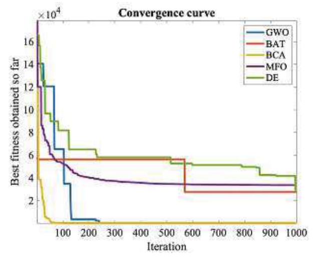
and entrapment in local optima. To evaluate this, we compare the convergence curves of BCA with several other meta-heuristic approaches such as DE, BAT, MFO, and GWO, as depicted in Figure 4.8. These curves illustrate the fitness value of the objective function against the number of iterations. The convergence plots reveal BCA's competitive performance and its ability to demonstrate robust convergence behavior across various benchmark functions when compared with other state-of-the-art optimizers.

In Figure 4.8, the convergence behavior of BCA across function optimization reveals three distinct patterns. Firstly, there is an acceleration in convergence as iterations progress for functions f_1, f_3, and f_9. Secondly, for unimodal functions, rapid convergence is apparent from early iterations, indicating BCA's quick exploitation of favorable solution spaces (mainly f_1, f_3, f_5, f_7). Another noteworthy observation is the quick avoidance of local optima for multimodal functions f_9, f_{11}, and f_{13}. Figure 4.8 also illustrates early convergence to global optima for fixed-dimensional multimodal functions f_{15}, f_{17}, and f_{19}, showcasing BCA's capability in extensive diversification within fewer iterations, coupled with effective intensification in promising solution areas. This behavior is attributed to diverse position-updating mechanisms and a gradual reduction in the propagation factor

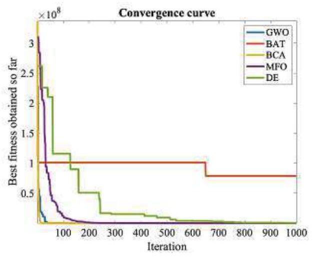
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub



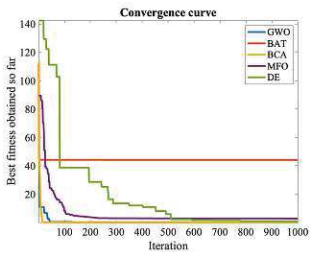
(a) f_1



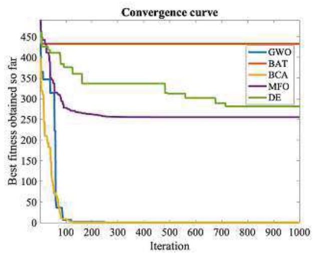
(b) f_3



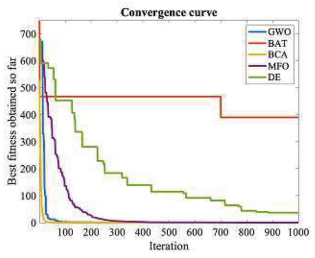
(c) f_5



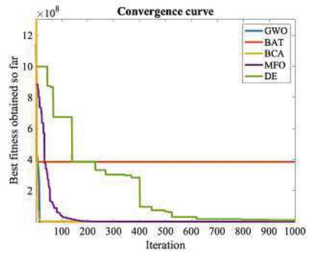
(d) f_7



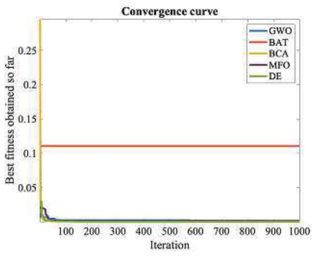
(e) f_9



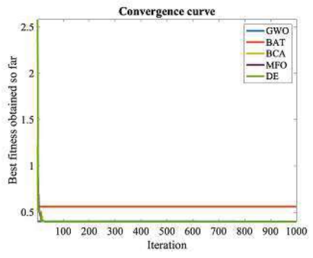
(f) f_{11}



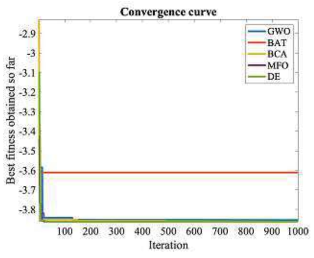
(g) f_{13}



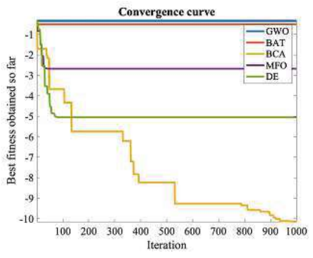
(h) f_{15}



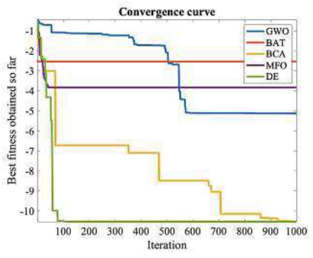
(i) f_{17}



(j) f_{19}



(k) f_{21}



(l) f_{23}

Figure 4.8: Comparison of convergence curves of BCA and few eminent algorithms for some of the benchmark functions.
License: CC BY 4.0

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

(P_f) across iterations. The third observed behavior manifests as convergence towards global optima primarily in later iterations, as evidenced by the curves of fixed-dimensional multimodal test functions f_{21} and f_{23}. This phenomenon likely arises from BCA's initial difficulty in finding optimal solutions, prompting continued exploration of the solution space until convergence to the global optimum is achieved.

Verification of convergence at global optima can also be obtained from the thrombocytes' search history (Figure 4.6) and their trajectories (Figure 4.7). In Figure 4.6, the dense regions in the contour plots of benchmark functions signify convergence of thrombocytes in those areas. Notably, BCA avoids premature convergence, as the optimal solution resides within these dense regions. Additionally, thrombocyte trajectories (Figure 4.7) terminate in regions housing global optima, further affirming convergence. These observations underscore BCA's adept balance of intensification and diversification, facilitating attainment of global optimal solutions. Overall, these analyses affirm BCA's high efficacy in addressing optimization challenges.

4.4.3 Statistical Significance Analysis

In our work, we employ the widely recognized Wilcoxon rank-sum test to statistically assess the efficacy of BCA relative to other optimization algorithms, as outlined in the literature [DGMH11]. Conducted at a significance level of 0.05, this test aims to identify significant differences between the outcomes of BCA and alternative optimizers. The resulting p-values, presented in Table 4.7, signify whether BCA exhibits superiority over its counterparts. Observations of p-values below 0.05 indicate the robust performance of BCA. Analysis of the p-values in Table 4.7 underscores BCA's significant outperformance compared to its competitors, with statistically meaningful differences observed across the majority of the test functions.

4.4.4 Influence of High Dimensionality

To showcase the robustness of BCA across varying problem dimensions, we conduct a scalability assessment. This analysis elucidates how the optimizer performs as problem dimensions increase, shedding light on its efficacy for both low and high-dimensional scenarios. Our evaluation encompasses both unimodal $(f_1 - f_7)$ and multimodal $(f_8 - f_{13})$ problems across dimensions ranging from 30 to 1000. By systematically exploring a wide range of dimensions, we gain insights into BCA's ability to maintain solution quality across diverse problem complexities.

Figure 4.9 depicts the scalability outcomes of BCA across various test functions at different dimensionalities (particularly, 30, 50, 80, 100, 500 and 1000). Analyzing the convergence curves, it is evident that BCA maintains strong performance across high-dimensional environments. Notably, BCA showcases an ability to converge to optimal solutions swiftly, as evidenced by the early iterations where optimal solutions are often attained. The curves also underscore BCA's adeptness in balancing intensification and diversification tendencies in multivariable optimization scenarios. Additionally, we compute the mean

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 4.7: p-values of the Wilcoxon rank-sum test with 5% significance for BCA versus other algorithms for the benchmark functions with 30 independent runs.(p-values ≥ 0.05 are indicated in bold face. NaN indicates “Not a Number” reported by the test.)

F	DE	PSO	GA	CS	GWO	MFO	FPA	FA	BAT	GSA	AOA	BBO
f_1	3.52E-11	2.73E-11	3.62E-11	3.53E-11	3.49E-11	3.29E-11	2.78E-11	3.28E-11	2.06E-11	3.19E-11	3.45E-11	2.87E-11
f_2	2.78E-11	2.67E-11	4.31E-11	2.71E-11	2.91E-11	2.48E-11	3.89E-11	2.91E-11	2.49E-11	2.48E-11	2.28E-11	4.45E-11
f_3	4.31E-11	2.91E-11	3.18E-11	4.37E-11	4.65E-11	4.81E-11	2.98E-11	3.27E-11	3.21E-11	3.46E-11	4.01E-11	2.78E-11
f_4	2.38E-11	2.65E-11	1.27E-11	2.32E-11	2.42E-11	2.20E-11	1.91E-11	1.99E-11	1.94E-11	2.29E-11	2.84E-11	2.56E-11
f_5	2.71E-11	2.71E-11	2.89E-11	2.79E-11	2.68E-11	2.09E-11	3.02E-11	3.00E-11	2.39E-11	2.16E-11	2.36E-11	2.98E-11
f_6	4.27E-11	2.84E-11	2.63E-11	4.20E-11	4.30E-11	4.71E-11	2.03E-11	3.67E-11	3.61E-11	3.28E-11	4.39E-11	5.91E-11
f_7	2.33E-11	2.42E-11	2.63E-11	2.36E-11	2.29E-11	2.67E-11	1.07E-11	3.01E-11	5.28E-11	2.87E-11	2.93E-11	2.77E-11
f_8	2.90E-10	2.79E-11	2.48E-10	2.99E-10	2.85E-10	2.82E-10	2.22E-10	2.76E-10	2.90E-10	2.39E-10	2.96E-10	2.91E-10
f_9	3.16E-12	2.45E-11	1.66E-12	3.11E-12	3.41E-12	3.56E-12	3.00E-12	3.91E-12	2.50E-12	3.71E-12	3.55E-12	3.97E-12
f_{10}	2.87E-12	1.19E-12	1.66E-12	2.86E-12	2.84E-12	2.49E-12	2.37E-12	2.45E-12	1.79E-12	2.05E-12	2.13E-12	4.67E-12
f_{11}	3.01E-11	1.56E-13	1.31E-12	3.05E-11	3.60E-11	3.91E-11	3.61E-11	3.11E-11	2.00E-11	2.78E-11	3.06E-11	2.14E-11
f_{12}	2.81E-11	1.87E-11	2.18E-11	2.05E-11	2.87E-11	2.48E-11	1.76E-11	2.92E-11	1.99E-11	2.47E-11	3.05E-11	2.66E-11
f_{13}	2.28E-11	2.54E-11	2.18E-11	2.82E-11	2.61E-11	2.29E-11	1.55E-11	1.08E-11	3.07E-11	2.81E-11	2.10E-11	2.41E-11
f_{14}	2.39E-09	5.21E-08	**6.32E-02**	6.16E-14	7.27E-08	3.24E-08	**1.81E-01**	1.48E-09	1.37E-12	9.23E-05	1.63E-06	3.25E-06
f_{15}	1.92E-11	7.39E-11	9.67E-11	3.34E-11	8.69E-11	5.29E-11	2.78E-11	3.92E-10	2.56E-11	1.37E-11	1.67E-08	2.82E-09
f_{16}	NaN	1.93E-13	2.35E-12	NaN	NaN	NaN	NaN	NaN	4.61E-10	NaN	3.92E-10	2.66E-12
f_{17}	**1.27E-01**	1.84E-12	3.28E-11	**1.27E-01**	**1.27E-01**	**1.27E-01**	**1.27E-01**	**1.27E-01**	**1.27E-01**	2.71E-12	1.39E-12	1.55E-11
f_{18}	8.29E-11	2.59E-13	4.36E-10	3.28E-08	2.16E-11	1.99E-10	3.29E-12	1.09E-10	2.27E-12	6.82E-13	3.05E-09	3.84E-10
f_{19}	3.05E-11	2.48E-10	2.39E-09	2.29E-11	2.76E-10	1.66E-12	8.69E-11	3.01E-11	2.48E-10	5.48E-09	3.64E-11	3.52E-11
f_{20}	3.42E-04	7.21E-05	3.93E-05	NaN	4.39E-05	2.16E-04	1.07E-05	8.27E-04	2.29E-05	1.01E-04	2.49E-05	3.27E-04
f_{21}	1.64E-07	8.11E-06	3.46E-11	7.37E-10	1.07E-08	1.44E-05	5.28E-10	1.05E-12	2.45E-08	2.91E-08	1.38E-08	2.16E-09
f_{22}	1.34E-11	6.47E-11	1.98E-11	5.71E-11	5.37E-11	3.59E-11	2.46E-11	2.04E-10	2.38E-11	4.28E-12	1.09E-08	1.89E-09
f_{23}	NaN	1.28E-12	2.05E-10	2.51E-11	1.94E-08	4.85E-09	8.37E-08	1.39E-09	4.37E-12	3.61E-08	2.57E-10	3.08E-08

License: CC BY 4.0

and standard deviation (SD) of results over 1000 iterations and 30 runs for dimensions of 30, 100, and 1000. These results, presented in Table 4.8, demonstrate BCA's consistent performance across various dimensions for most test functions ($f_1 - f_{13}$). Overall, these experimental findings underscore the scalability and efficiency of BCA, particularly in handling high-dimensional tasks.

Table 4.8: Influence of high dimensionality on the performance of BCA: Results of benchmark functions ($f_1 - f_{13}$) with different dimensions.

	Dimensions = 30		**Dimensions = 100**		**Dimensions = 1000**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**
f_1	9.80E-28	1.70E-22	5.06E-15	1.95E-14	1.24E-13	3.93E-13
f_2	8.91E-17	1.36E-11	9.80E-08	5.24E-07	2.60E-07	8.97E-07
f_3	7.63E-16	2.46E-15	2.40E-11	9.48E-11	1.35E-08	5.71E-08
f_4	7.01E-10	1.77E-09	1.95E-08	8.32E-08	5.08E-08	1.52E-07
f_5	3.72E+00	8.72E+00	3.34E+01	3.69E+01	5.41E+01	3.85E+01
f_6	1.36E-02	9.83E-03	3.25E-01	2.59E-01	7.73E+01	4.58E+01
f_7	3.95E-04	4.78E-04	5.25E-04	4.79E-04	6.34E-04	7.71E-04
f_8	-1.25E+04	4.30E+01	-4.18E+04	2.01E+02	-4.18E+05	4.36E+03
f_9	1.70E-14	3.04E-14	4.92E-14	1.06E-13	1.81E-12	2.70E-12
f_{10}	8.88E-16	2.89E-11	7.27E-10	2.34E-09	9.14E-09	2.78E-08
f_{11}	3.49E-15	1.86E-14	2.16E-15	7.31E-15	5.55E-15	2.60E-14
f_{12}	4.19E-06	5.43E-04	2.31E-03	1.28E-03	2.74E-03	1.64E-03
f_{13}	1.99E-03	5.37E-03	7.14E-02	5.95E-02	1.01E+00	7.95E-01

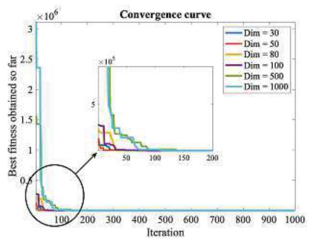
License: CC BY 4.0

4.5 BCA for Standard Engineering Problems

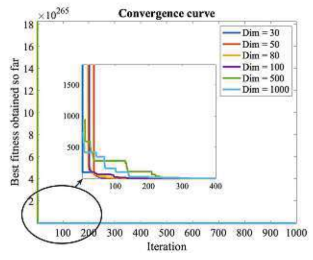
Next, we illustrate the effectiveness and versatility of the proposed algorithm (BCA) in tackling real-world engineering optimization challenges, both constrained and unconstrained. To assess its performance, we apply BCA to six standard benchmark engineering design problems: the welded beam design problem, pressure vessel design problem, tension/compression spring design problem, three-bar truss design problem, speed reducer design problem, and gear train design problem.

The engineering design optimization tasks undertaken and the variables involved are summarized in Table 4.9. In addressing these problems, BCA is employed across 30 independent runs, each comprising a population size (N_{Pop}) of 30 and 1000 iterations. To handle the various constraints inherent in engineering design problems, a constraint handling method is necessary. For simplicity, we adopt the death penalty approach (also known as scalar penalty function), where solutions violating constraints are penalized with a significant fitness value [Coe02]. This method ensures that solutions adhering to

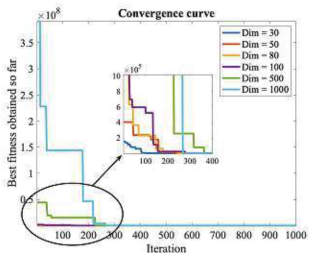
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub



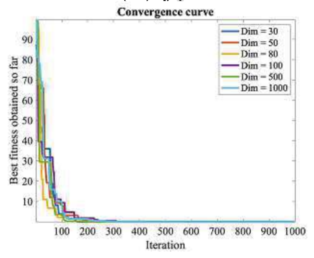
(a) f_1



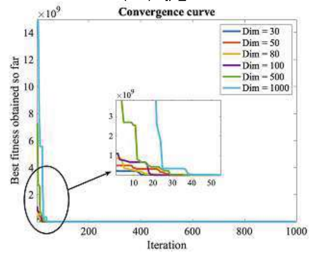
(b) f_2



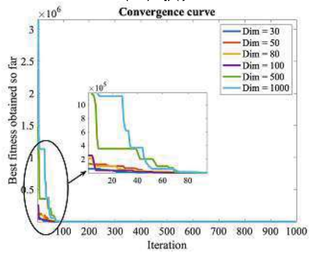
(c) f_3



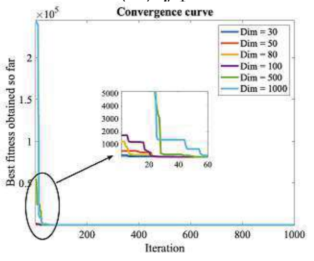
(d) f_4



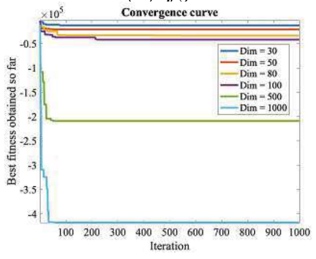
(e) f_5



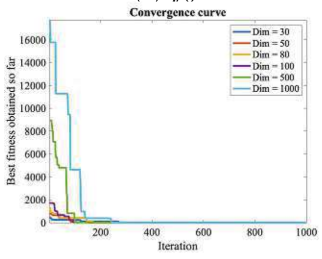
(f) f_6



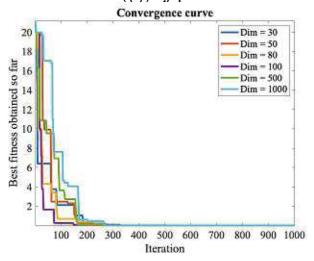
(g) f_7



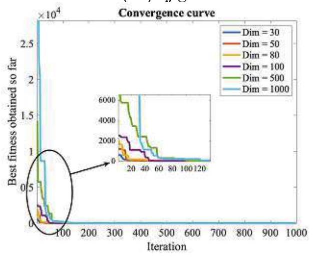
(h) f_8



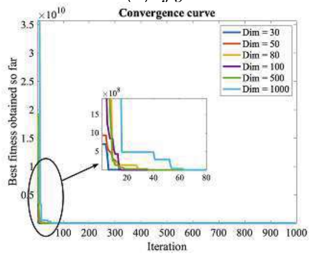
(i) f_9



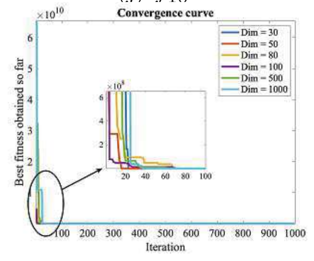
(j) f_{10}



(k) f_{11}



(l) f_{12}



(m) f_{13}

Figure 4.9: Scalability analysis of the proposed BCA for different dimensions of the benchmark functions ($f_1 - f_{13}$).
License: CC BY 4.0

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

constraints are favored during optimization. Also, we compare the results of BCA with other meta-heuristic approaches in the literature.

Table 4.9: Brief description of the tackled engineering design optimization problems.

Problem	**D, C**	**Objective**	**Description of variables**
Welded beam design	4, 7	Minimize cost	thickness of weld (h), length of the clamped bar (l), height of the bar (t), thickness of the bar (b)
Pressure vessel design	4, 4	Minimize cost	the thickness of the shell (T_s), the thickness of the head (T_h), the inner radius (R), the length of the cylindrical section without considering the head (L)
Tension-compression spring design	3, 4	Minimize weight	wire diameter (d), mean coil diameter (D), number of active coils (N)
Three-bar truss design	2, 3	Minimize weight	area of bars 1 and 3 (A_1), area of bar 2 (A_2)
Speed reducer design	7, 11	Minimize weight	Face width (x_1), Teeth module (x_2), Number of teeth (x_3), First shaft length (x_4), Second shaft length (x_5), First shaft diameter (x_6), Second shaft diameter (x_7)
Gear train design	4, 0	Minimize gear ratio	Gear teeth: T_a, T_b, T_d, T_f

Note: D = number of variables, C= total number of constraints
License: CC BY 4.0

Welded beam design problem. The welded beam design problem stands as a classic benchmark, aiming to minimize the fabrication expenses related to the welded beam. It involves constraints on shear stress (τ), bending stress (σ), buckling load (P_c), and end deflection (δ) of the beam, with design variables outlined in Table 4.9. Mathematically, the optimization objective can be expressed as follows:

Consider $x = [x_1\, x_2\, x_3\, x_4] = [h\, l\, t\, b]$
Minimize $f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$
Subject to $g_1(x) = \tau(x) - \tau_{max} \leq 0$
$g_2(x) = \sigma(x) - \sigma_{max} \leq 0,$
$g_3(x) = \delta(x) - \delta_{max} \leq 0,$
$g_4(x) = x_1 - x_4 \leq 0,$
$g_5(x) = P - P_c(x) \leq 0,$
$g_6(x) = 0.125 - x_1 \leq 0,$
$g_7(x) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0,$
Variable range $0.1 \leq x_1 \leq 2,$
$0.1 \leq x_2 \leq 10,$
$0.1 \leq x_3 \leq 10,$
$0.1 \leq x_4 \leq 2$
where $\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},$
$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{x_2}{2}\right), R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2}, J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\},$

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

$\sigma(x) = \frac{6PL}{x_4 x_3^2}, \delta(x) = \frac{6PL^3}{Ex_4 x_3^2}, P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),$
$P = 6000\,lb, L = 14\,in., \delta_{max} = 0.25\,in., E = 30 \times 10^6\,psi, G = 12 \times 10^6\,psi,$
$\tau_{max} = 13600\,psi, \sigma_{max} = 30000\,psi$

Many researchers have tackled this engineering design optimization problem using various algorithms. Table 4.10 presents a comparison of the best solution attained by BCA and other algorithms. The results clearly demonstrate that BCA achieves an optimal design with minimized cost, surpassing all other algorithms. Additionally, statistical metrics derived from 30 independent runs of BCA, including the best, worst, mean, and standard deviation (SD) values, are compared with existing algorithms in the literature, as shown in Table 4.11. The statistical analysis for the welded beam design problem presented in Table 4.11 underscores BCA's ability to produce superior results, reaffirming its effectiveness in tackling this optimization challenge.

Pressure vessel design problem. Another renowned structural design benchmark problem is the pressure vessel design, which aims to minimize the overall fabrication cost (including material, forming, and welding) of a cylindrical pressure vessel with hemispherical heads at both ends. The design variables are outlined in Table 4.9. The optimization model for this problem can be expressed mathematically as follows:

Consider $x = [x_1\, x_2\, x_3\, x_4] = [T_s\, T_h\, R\, L]$
Minimize $f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$
Subject to $g_1(x) = -x_1 + 0.0193x_3 \leq 0,$
$g_2(x) = -x_2 + 0.00954x_3 \leq 0,$
$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0,$
$g_4(x) = x_4 - 240 \leq 0,$
Variable range $0 \leq x_1 \leq 99,$
$0 \leq x_2 \leq 99,$
$10 \leq x_3 \leq 200,$
$10 \leq x_4 \leq 200$

Numerous researchers have tackled this design problem using various algorithms, encompassing both meta-heuristic approaches and mathematical techniques. Table 4.12 offers a comparison of the optimal solutions obtained by BCA and other prominent algorithms documented in the literature. Upon reviewing the results in Table 4.12, it is apparent that BCA yields results comparable to PO and even surpasses other optimizers. Thus, we can assert that BCA is proficient in discovering feasible optimal designs for pressure vessels at minimized costs (specifically, 5885.3991). Furthermore, Table 4.13 presents statistical insights (including best, worst, mean, and standard deviation values) of BCA and other optimizers concerning the pressure vessel design problem. These results highlight the superior performance of BCA compared to its counterparts. Notably, BCA exhibits a considerably lower standard deviation of 8.4237, signifying its reliability and efficiency in solving this optimization problem.

Tension/compression spring design problem. The aim of the tension/compression

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

Table 4.10: Comparison of the best solution obtained by BCA with other algorithms in the literature for the welded beam design problem.

Algorithm	**Optimal values of design variables**				**Optimal cost**
	h	l	t	b	
BCA	**0.205729**	**3.470471**	**9.036622**	**0.205729**	**1.72484**
HHO [HMF+19]	0.204039	3.531061	9.027463	0.206147	1.73199057
RANDOM [RP76]	0.4575	4.7313	5.0853	0.66	4.1185
DAVID [RP76]	0.2434	6.2552	8.2915	0.2444	2.3841
SIMPLEX [RP76]	0.2792	5.6256	7.7512	0.2796	2.5307
APPROX [RP76]	0.2444	6.2189	8.2915	0.2444	2.3815
GA [Deb91]	0.248900	6.173000	8.178900	0.253300	2.433116
GA [Coe00]	0.208800	3.420500	8.997500	0.210000	1.748310
HS [LG04]	0.2442	6.2231	8.2915	0.2443	2.3807
ESs [MMC05]	0.199742	3.61206	9.0375	0.206082	1.7373
CDE [HWH07]	0.203137	3.542998	9.033498	0.206179	1.733462
MFO [Mir15]	0.203567	3.443025	9.230278	0.212359	1.732541
MVO [MMH16]	0.205611	3.472103	9.040931	0.205709	1.725472
SCA [Mir16b]	0.204695	3.536291	9.004290	0.210025	1.759173
GA [BDT99]	0.164171	4.032541	10.00000	0.223647	1.873971
ES [MMC08]	0.199742	3.612060	9.037500	0.20682	1.73730
SA [HF06]	0.20564426	3.472578742	9.03662391	0.2057296	1.7250022
Co-evolutionary PSO [HW07a]	0.20573	3.47049	9.03662	0.20573	1.72485084
GSA [RNPS09]	0.18219	3.856979	10.0000	0.202376	1.879952
Improved PSO [CEC08]	0.205729	3.470488	9.036624	0.205729	1.724852
DE [MMCCVRMD07]	0.20573	3.470489	9.0336624	0.205730	1.724852
CS [GYA13]	0.2015	3.562	9.0414	0.2057	1.73121
ABC [KB08]	0.205730	3.470489	9.036624	0.205730	1.724852
ACO [KT10a]	0.205700	3.471131	9.036683	0.205731	1.724918
PO [AYS20]	0.205730	3.470472	9.036624	0.205730	1.724851
CAEP [CCB04]	0.205700	3.470500	9.036600	0.205700	1.724852
HGA [YQ10]	0.205700	3.470500	9.036600	0.205700	1.7249
WCA [ESBH12]	0.205728	3.470522	9.036620	0.205729	1.724856
CGWO [KA18]	0.343891	1.883570	9.031330	0.212121	1.725450
GWO [MML14]	0.205676	3.478377	9.036810	0.205778	1.726240
CPSO [KdSC06]	0.202369	3.544214	9.048210	0.205723	1.728024
GA [CM02]	0.205986	3.471328	9.020224	0.206480	1.728226
WOA [ML16]	0.205396	3.484293	9.037426	0.206276	1.730499
CS [YD09]	0.182200	3.795100	9.998100	0.211100	1.946000
BA [Yan10]	0.154300	5.736100	8.862700	0.229700	2.084000
CapSA [BSAH21]	0.205723	3.470789	9.036622	0.205737	1.7249

License: CC BY 4.0

spring design problem [Aro04] is to minimize the weight of the spring while ensuring it meets specific constraints related to shear stress, surge frequency, and deflection. The design variables for this optimization problem are outlined in Table 4.9. Formally, the optimization problem can be expressed as follows:

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 4.11: Comparison of statistical results of BCA with other optimization algorithms in the literature for solving the welded beam design problem.

Algorithm	**Best**	**Worst**	**Mean**	**SD**
BCA	**1.72484**	**1.7272306**	**1.7254101**	**4.237E-7**
MFO [Mir15]	1.732541	1.802364	1.775231	0.012397
MVO [MMH16]	1.725472	1.741651	1.729680	0.004866
SCA [Mir16b]	1.759173	1.873408	1.817657	0.027543
GA [BDT99]	1.873971	2.320125	2.119240	0.034820
ES [MMC08]	1.728226	1.993408	1.792654	0.07471
SA [HF06]	1.7250022	1.8843960	1.7564428	NA
Co-evolutionary PSO [HW07a]	1.728024	1.782143	1.748831	0.012926
Improved PSO [CEC08]	1.724852	NA	2.0574	0.2154
DE [MMCCVRMD07]	1.724852	1.725000	1.725	1.0E-15
CS [GYA13]	1.7312065	2.3455793	1.8786560	0.2677989
ABC [KB08]	1.724852	NA	1.741913	0.031
ACO [KT10a]	1.72918	1.775961	1.729752	0.009200
PO [AYS20]	1.724851	1.724852	1.724851	2.53E-07
CAEP [CCB04]	1.724852	3.179709	1.971809	0.443000
WCA [ESBH12]	1.724856	1.744697	1.726427	0.004290
CGWO [KA18]	1.725450	2.435700	2.428900	1.357800
CPSO [KdSC06]	1.728024	1.782143	1.748831	0.012900
CapSA [BSAH21]	1.72481904	1.72723071	1.72541110	4.2376E-7
PSO-DE [LCW10]	1.724852	1.724852	1.724852	6.7E-16
COMDE [MS12]	1.724852	1.724852	1.724852	1.6E-12
DELC [WL10]	1.724852	1.724852	1.724852	4.1E-13
MADE [HAL$^+$18]	1.724852	1.724852	1.724852	9.6E-16
AMDE [AFI17]	1.724852	1.724852	1.724852	1.1E-15

NA = Not available, SD = Standard Deviation
License: CC BY 4.0

Consider $x = [x_1\, x_2\, x_3] = [d\, D\, N]$
Minimize $f(x) = (x_3 + 2)x_2 x_1^2$
Subject to $g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0,$
$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} \leq 0,$
$g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0,$
$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0,$
Variable range $0.05 \leq x_1 \leq 2.00,$
$0.25 \leq x_2 \leq 1.30,$
$2.00 \leq x_3 \leq 15.0$

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Table 4.12: Comparison of the best solution obtained by BCA with other algorithms in the literature for the pressure vessel design problem.

Algorithm	Optimal values of design variables				Optimal cost
	T_s	T_h	R	L	
BCA	**0.7782**	**0.3847**	**40.3215**	**199.973**	**5885.3991**
MFO [Mir15]	0.835241	0.409854	43.578621	152.21520	6055.6378
MVO [MMH16]	0.845719	0.418564	43.816270	156.38164	6011.5148
SCA [Mir16b]	0.817577	0.417932	41.74939	183.57270	6137.3724
GA [BDT99]	0.752362	0.399540	40.452514	198.00268	5890.3279
HS [LG05]	1.099523	0.906579	44.456397	179.65887	6550.0230
DA [Mir16a]	0.782825	0.384649	40.3196	200	5923.11
Co-evolutionary PSO [HW07a]	0.812500	0.437500	42.091266	176.746500	6061.077
ES [MMC08]	0.812500	0.437500	42.098087	176.640518	6059.7456
CS [GYA13]	0.812500	0.437500	42.0984456	176.6363595	6059.7143348
ABC [AK12]	0.812500	0.437500	42.098446	176.636596	6059.714339
Improved PSO [HPW04]	0.812500	0.437500	42.098445	176.6365950	6059.7143
Penalty guided ABC [Gar14]	0.7781686	0.3846491	40.3210545	199.9802367	5885.40322828
DE [MMCCVRMD07]	0.812500	0.437500	42.098446	176.6360470	6059.701660
WOA [ML16]	0.812500	0.437500	42.0982699	176.638998	6059.7410
PO [AYS20]	0.7782	0.3847	40.3215	199.9733	5885.3997
NMPSO [ZK09]	0.8036	0.3972	41.6392	182.4120	5930.3137
GWO [MML14]	0.8125	0.4345	42.0892	176.7587	6051.5639
HPSO [HW07b]	0.8125	0.4375	42.0984	176.6366	6059.7143
G-QPSO [dSC10]	0.8125	0.4375	42.0984	176.6372	6059.7208
CDE [HWH07]	0.8125	0.4375	42.0984	176.6376	6059.7340
DE [LS04]	0.8125	0.4375	42.0984	176.6377	6059.7340
GA [CM02]	0.8125	0.4375	42.0974	176.6540	6059.9463
CPSO [KdSC06]	0.8125	0.4375	42.0913	176.7465	6061.0777
GSA [RNPS09]	1.1250	0.6250	55.9887	84.4542	8538.8359
HHO [HMF+19]	0.81758383	0.4072927	42.09174576	176.7196352	6000.46259
GA [Deb91]	0.812500	0.437500	42.097398	176.654050	6059.9463
Lagrangian multiplier (Kannan) [ML16]	1.125000	0.625000	58.291000	43.6900000	7198.0428
Branch-bound (Sandgren) [ML16]	1.125000	0.625000	47.700000	117.701000	8129.1036
DELC [WL10]	0.812500	0.437500	42.0984456	176.6365958	6059.7143
CSS [KT10b]	0.812500	0.437500	42.103624	176.572656	6059.0888
ESs [MMC05]	0.812500	0.437500	42.098087	176.640518	6059.7456

License: CC BY 4.0

Various optimization algorithms have been employed previously to tackle this design challenge. Table 4.14 provides a comparative overview of the optimal solutions obtained by BCA and alternative algorithms for the tension/compression spring design problem. The results depicted in Table 4.14 indicate that BCA successfully identifies an optimal spring design with minimal weight. Furthermore, statistical analyses comparing BCA with other methods from existing literature are summarized in Table 4.15. These results suggest that BCA either surpasses or performs comparably to other algorithms listed in Table 4.15.

Three-bar truss design problem. This engineering design problem holds significance in optimization studies due to its inherently constrained search space. The objective revolves

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Table 4.13: Comparison of statistical results of BCA with other optimization algorithms in the literature for solving the pressure vessel design problem.

Algorithm	**Best**	**Worst**	**Mean**	**SD**
BCA	**5885.3991**	**5910.5321**	**5891.3265**	**8.4237**
MFO [Mir15]	6055.6378	7023.8521	6360.6854	365.597
MVO [MMH16]	6011.5148	7250.9170	6477.3050	327.007
SCA [Mir16b]	6137.3724	6512.3541	6326.7606	126.609
GA [BDT99]	5890.3279	7005.7500	6264.0053	496.128
HS [LG05]	6550.0230	8005.4397	6643.9870	657.523
DA [Mir16a]	5923.11	222536	21342.2	47044.2
Co-evolutionary PSO [HW07a]	6061.077	6363.8041	6147.1332	86.4545
ES [MMC08]	6059.7456	7332.8798	6850.004	9426.000
CS [GYA13]	6059.714	6495.3470	6447.7360	502.693
ABC [AK12]	6059.714339	NA	6245.308144	205
Improved PSO [HPW04]	6059.7143	NA	6289.92881	305.78
WOA [ML16]	NA	NA	6068.0500	65.6519
GSA [RNPS09]	NA	NA	8932.9500	683.5475
PO [AYS20]	5885.3997	5908.0250	5891.8068	8.4746
NMPSO [ZK09]	5930.3137	5960.0557	5946.7901	9.1610
PSO-DE [LCW10]	6059.7140	NA	6059.7140	NA
HPSO [HW07b]	6059.7143	6288.6770	6099.9323	86.2000
CPSO [KdSC06]	6061.0777	6363.8041	6147.1332	86.4500
GWO [MML14]	6051.5630	6395.3600	6159.3200	379.6740
G-QPSO [dSC10]	6059.7208	7544.4925	6440.3786	448.4711

NA = Not available, SD = Standard Deviation
License: CC BY 4.0

around discovering the optimal configuration for a truss with three bars, minimizing its weight. The specifics of this problem are outlined in Table 4.9, encompassing various constraints related to stress, deflection, and buckling. Mathematically, this problem can be expressed as follows:

Consider $x = [x_1\, x_2] = [A_1\, A_2]$
Minimize $f(x) = (2\sqrt{2}x_1 + x_2)l$
Subject to $g_1(x) = \frac{\sqrt{2}x_1+x_2}{\sqrt{2}x_1^2+2x_1x_2}P - \sigma \leq 0,$
$g_2(x) = \frac{x_2}{\sqrt{2}x_1^2+2x_1x_2}P - \sigma \leq 0,$
$g_3(x) = \frac{1}{\sqrt{2}x_2+x_1}P - \sigma \leq 0,$
Variable range $0 \leq x_1,\, x_2 \leq 1,$
where $l = 100\, cm,\ P = 2\, kN/cm^2, \sigma = 2kN/cm^2$

The results of applying BCA to solve the 3-bar truss design problem are detailed in

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 4.14: Comparison of the best solution obtained by BCA with other algorithms in the literature for the tension/compression spring design problem.

Algorithm	**Optimal values of design variables**			**Optimal cost**
	d	D	N	
BCA	0.05248	0.37594	10.24509	0.01267
PO [AYS20]	0.05248	0.37594	10.24509	0.01267
DEDS [ZLW08]	0.05169	0.35672	11.28897	0.01267
HEAA [WCZF09]	0.05169	0.35673	11.28829	0.01267
DELC [WL10]	0.05169	0.35672	11.28897	0.01267
WCA [ESBH12]	0.05168	0.35652	11.30041	0.01267
MADE [HAL+18]	0.05169	0.35672	11.28897	0.01267
GWO [MML14]	0.05169	0.35674	11.28885	0.01267
DE [LS04]	0.05161	0.35471	11.41083	0.01267
HS [LG05]	0.05115	0.34987	12.07643	0.01267
CPSO [KdSC06]	0.05173	0.35764	11.24454	0.01267
WOA [ML16]	0.05121	0.34522	12.00403	0.01268
GA [CM02]	0.05199	0.36397	10.89052	0.01268
GSA [RNPS09]	0.05028	0.32368	13.52541	0.01270
MFO [Mir15]	0.05000	0.313501	14.03279	0.012753902
MVO [MMH16]	0.05000	0.315956	14.22623	0.012816930
SCA [Mir16b]	0.050780	0.334779	12.72269	0.012709667
SSA [MGM+17]	0.051207	0.345215	12.004032	0.0126763

License: CC BY 4.0

Table 4.16. Moreover, the performance of BCA is benchmarked against various algorithms documented in existing literature for tackling this challenge. The comparison reveals that BCA delivers competitive results when compared with algorithms such as HHO [HMF+19], DEDS [ZLW08], PSO–DE [LCW10], and SSA [MGM+17]. Notably, BCA surpasses several other algorithms, including MVO [MMH16], GOA [SML17], MFO [Mir15], CS [GYA13], and AOA [ADM+21]. These findings underscore BCA's efficacy in navigating constrained optimization spaces.

Speed reducer design problem. The objective of the Speed reducer design problem is to minimize the weight of the speed reducer by determining the optimal values of design variables. Constraints are imposed on stresses in the shafts, transverse deflection of the shafts, surface stress, and bending stress of the gear teeth. The design variables are outlined in Table 4.9. Mathematically, the problem can be expressed as follows:

Consider $x = [x_1\, x_2\, x_3\, x_4\, x_5\, x_6\, x_7]$
Minimize $f(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$
Subject to $g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0,$

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 4.15: Comparison of statistical results of BCA with other optimization algorithms in the literature for solving the tension/compression spring design problem.

Algorithm	**Worst**	**Mean**	**Best**	**SD**
BCA	**0.0127**	**0.0127**	**0.0127**	**0.0000**
PO [AYS20]	0.0128	0.0127	0.0127	0.0000
DELC [WL10]	0.0127	0.0127	0.0127	0.0000
HEAA [WCZF09]	0.0127	0.0127	0.0127	0.0000
PSO–DE [LCW10]	0.0127	0.0127	0.0127	0.0000
MADE [HAL$^+$18]	0.0127	0.0127	0.0127	0.0000
AMDE [AFI17]	0.0127	0.0127	0.0127	0.0000
DEDS [ZLW08]	0.0127	0.0127	0.0127	0.0000
DE [LS04]	0.0128	0.0127	0.0127	0.0000
HPSO [HW07b]	0.0127	0.0127	0.0127	0.0000
ABC [AK12]	NA	0.0127	0.0127	0.0128
CPSO [KdSC06]	0.0129	0.0127	0.0127	0.0005
GA [CM02]	0.0130	0.0127	0.0127	0.0001
WCA [ESBH12]	0.0130	0.0127	0.0127	0.0001
G-QPSO [dSC10]	0.0178	0.0135	0.0127	0.0013
CAEP [CCB04]	0.0151	0.0136	0.0127	0.0008
GSA [RNPS09]	NA	0.0136	NA	0.0026

NA = Not available, SD = Standard Deviation
License: CC BY 4.0

$$g_2(x) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \leq 0,$$
$$g_3(x) = \frac{1.93 x_4^3}{x_2 x_6^4 x_3} - 1 \leq 0,$$
$$g_4(x) = \frac{1.93 x_5^3}{x_2 x_7^4 x_3} - 1 \leq 0,$$
$$g_5(x) = \frac{\left[\left(\frac{745 x_4}{x_2 x_3}\right)^2 + (16.9 \times 10^6)\right]^{\frac{1}{2}}}{110 x_6^3} - 1 \leq 0,$$
$$g_6(x) = \frac{\left[\left(\frac{745 x_5}{x_2 x_3}\right)^2 + (157.5 \times 10^6)\right]^{\frac{1}{2}}}{85 x_7^3} - 1 \leq 0,$$
$$g_7(x) = \frac{x_2 x_3}{40} - 1 \leq 0,$$
$$g_8(x) = \frac{5 x_2}{x_1} - 1 \leq 0,$$
$$g_9(x) = \frac{x_1}{12 x_2} - 1 \leq 0,$$
$$g_{10}(x) = \frac{1.5 x_6 + 1.9}{x_4} - 1 \leq 0,$$
$$g_{11}(x) = \frac{1.1 x_7 + 1.9}{x_5} - 1 \leq 0,$$

Variable range $2.6 \leq x_1 \leq 3.6$,
$0.7 \leq x_2 \leq 0.8$,

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Table 4.16: Comparison of the best solution obtained by BCA with other algorithms in the literature for the three-bar truss design problem.

Algorithm	**Optimal values of design variables**		**Optimal cost**
	x_1	x_2	
BCA	**0.788662816**	**0.408283133832**	**263.8958434**
HHO [HMF+19]	0.788662816	0.408283133832900	263.8958434
DEDS [WCZF09]	0.78867513	0.40824828	263.8958434
MVO [MMH16]	0.78860276	0.408453070000000	263.8958499
GOA [SML17]	0.788897555578973	0.407619570115153	263.895881496069
MFO [Mir15]	0.788244771	0.409466905784741	263.8959797
PSO–DE [LCW10]	0.7886751	0.4082482	263.8958433
SSA [MGM+17]	0.788665414	0.408275784444547	263.8958434
MBA [SBEH13]	0.7885650	0.4085597	263.8958522
Tsai [Tsa05]	0.788	0.408	263.68
Ray and Saini [RS01]	0.795	0.395	264.3
CS [GYA13]	0.78867	0.40902	263.9716
AOA [ADM+21]	0.79369	0.39426	263.9154

License: CC BY 4.0

$17 \leq x_3 \leq 28,$
$7.3 \leq x_4 \leq 8.3,$
$7.3 \leq x_5 \leq 8.3,$
$2.9 \leq x_6 \leq 3.9,$
$5.0 \leq x_7 \leq 5.5$

The results achieved by BCA for the speed reducer design problem are compared against various other algorithms documented in the literature, as presented in Table 4.17. The comparison indicates that the performance of BCA is comparable to that of other algorithms, demonstrating satisfactory performance. Additionally, statistical findings for the speed reducer design problem utilizing BCA and other meta-heuristic algorithms are summarized in Table 4.18. These statistical results affirm that BCA provides highly competitive outcomes for addressing this problem.

Gear train design problem. The gear train design problem, a prominent discrete optimization challenge in mechanical engineering, was first introduced in 1990 [San90]. Its objective is to minimize the gear ratio, as defined by Equation (4.11), for a compound gear train comprising four gears. The problem revolves around optimizing the number of teeth of these gears, resulting in four integer variables within the range of 12 to 60. Mathematically, the cost (objective) function, is formulated as shown in Equation (4.12):

$$\text{Gear ratio} = \frac{\text{Angular velocity of the output shaft}}{\text{Angular velocity of the input shaft}} \tag{4.11}$$

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Table 4.17: Comparison of the best solution obtained by BCA with other algorithms in the literature for the Speed reducer design problem.

Algorithm	**Optimal values of design variables**							**Optimal Weight**
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
BCA	**3.5**	**0.7**	**17**	**7.3**	**7.7153**	**3.3502**	**5.2867**	**2994.471**
PO [AYS20]	3.5	0.7	17	7.3	7.7153	3.3502	5.2867	**2994.471**
DEDS [ZLW08]	3.5	0.7	17	7.3	7.7153	3.3502	5.2867	**2994.471**
DELC [WL10]	3.5	0.7	17	7.3	7.7153	3.3502	5.2867	**2994.471**
WCA [ESBH12]	3.5	0.7	17	7.3	7.7153	3.3502	5.2867	**2994.471**
CapSA [BSAH21]	3.500	0.7	17	7.30	7.715320	3.350215	5.286654	**2994.4710**
HEAA [WCZF09]	3.5	0.7	17	7.3004	7.7155	3.3502	5.2867	2994.499
PSO-DE [LCW10]	3.5	0.7	17	7.3	7.8000	3.3502	5.2867	2996.348
MDE [MMVRC06]	3.5	0.7	17	7.3002	7.8000	3.3502	5.2867	2996.357
MFO [Mir15]	3.497455	0.700	17	7.82775	7.712457	3.351787	5.286352	2998.94083
WSA [BA15]	3.500	0.7	17	7.3	7.8	3.350215	5.286683	2996.348225
AAO [CZE17]	3.499	0.6999	17	7.3	7.8	3.3502	5.2872	2996.783
GWO [MML14]	3.501	0.7	17	7.3	7.811013	3.350704	5.287411	2997.81965
APSO [Gue16]	3.501313	0.7	18	8.127814	8.042121	3.352446	5.287076	3187.630486
CS [GYA13]	3.5015	0.7000	17	7.6050	7.8181	3.3520	5.2875	3000.9810
SCA [Mac64]	3.521	0.7	17	8.3	7.923351	3.355911	5.300734	3026.83772
FA [BO15]	3.507495	0.7001	17	7.719674	8.080854	3.351512	5.287051	3010.137492
AOA [ADM+21]	3.50384	0.7	17	7.3	7.72933	3.35649	5.2867	2997.9157

License: CC BY 4.0

$$f(T_a, T_b, T_d, T_f) = \left(\frac{1}{6.931} - \frac{T_b T_d}{T_a T_f}\right)^2 \tag{4.12}$$

where T_i denotes the number of teeth on the i-th gear wheel, where i corresponds to a, b, d, and f. The objective is to determine the number of teeth on each wheel to achieve a gear ratio of $\frac{1}{6.931}$. Although this problem does not have explicit constraints, we impose constraints by considering the range of variables.

The gear design problem has garnered significant attention from researchers and has been addressed using various heuristic methods. In this study, we employ BCA to tackle this problem and compare its performance with other algorithms found in the literature. Table 4.19 summarizes the outcomes of the gear train design problem, including the optimal parameters and the best objective function value obtained by BCA and alternative algorithms. The results demonstrate that BCA yields competitive outcomes, matching the optimal function value computed by other algorithms such as MFO, ABC, MBA, CS, and ISA. This underscores BCA's efficacy in effectively addressing discrete problems.

4.6 BCA for Falsification of CPSs

As previously discussed in Chapter 1, the falsification of CPSs plays a crucial role in uncovering system parameters (i.e., counterexamples) that may lead to defects, thus facilitating effective fault detection. In this context, we examine the Simulink model of

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 4.18: Comparison of statistical results of BCA with other optimization algorithms in the literature for solving the Speed reducer design problem.

Algorithm	Worst	Mean	Best	SD
BCA	**2994.471026**	**2994.471026**	**2994.471026**	**0.000003**
PO [AYS20]	2994.471057	2994.471051	2994.471047	0.000003
WCA [ESBH12]	2994.505578	2994.474392	2994.471066	0.007400
MDE [MMVRC06]	NA	2996.367220	2996.356689	0.008200
DELC [WL10]	2994.471066	2994.471066	2994.471066	0.000000
DEDS [ZLW08]	2994.471066	2994.471066	2994.471066	0.000000
ABC [AK12]	NA	2997.058000	2997.058000	0.000000
HEAA [WCZF09]	2994.752311	2994.613368	2994.499107	0.070000
PSO-DE [LCW10]	2996.348204	2996.348174	2996.348167	0.000006
SC [RL03]	3009.964736	3001.758264	2994.744241	4.000000
CapSA [BSAH21]	2998.09236	2995.12109	2994.47106	0.00002901

NA = Not available, SD = Standard Deviation
License: CC BY 4.0

the *Automatic Transmission Controller System* (ATCS), an illustration sourced from the avionics domain [ZKH03, Mat22d]. This model encompasses two input signals—throttle and brake—and features three outputs: vehicle speed v (mph), engine speed ω (RPM), and the gear position.

4.6.1 The Problem

Our focus lies in determining whether the system consistently adheres to the temporal property ${\phi_1}^{AT}$, which requires that both the engine and the vehicle speed never attain $\bar{\omega}$ and $\bar{v}$, respectively. This requirement is formulated in STL as follows: ${\phi_1}^{AT}(\bar{v}\,,\,\bar{\omega}) = \Box((v < \bar{v}) \wedge (\omega < \bar{\omega}))$. To violate this property, we search input signals capable of causing either the vehicle speed or the engine RPM to reach their designated thresholds. For this study, we set $\bar{v}$ = 120 mph and $\bar{\omega}$ = 4500 RPM. The total duration of simulation is T = 30 seconds.

The falsification methodology represents an optimization problem focused on minimizing the robustness, acting as the objective function, across decision variables z. These variables determine the input signal(s). The objective is to find values for z that lead to a resulting trace $\mathbf{x}$ with negative robustness, expressed as $\rho(\mathbf{x}, \phi) < 0$.

4.6.2 Simulation Results

In the following, we present the results of falsification testing with BCA and compare it with (i) other optimizers and, (ii) a state-of-the-art falsification testing tool.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 4.19: Comparison of results for Gear train design problem.

Approach	**Optimal values**				f_{min}
	T_a	T_b	T_d	T_f	
BCA	**43**	**16**	**19**	**49**	**2.7009E-12**
CAPSO [GYYT13]	49	19	16	43	2.701E-12
MFO [Mir15]	43	19	16	49	2.7009E-12
ABC [SBEH13]	49	16	19	43	2.7009E-12
MBA [SBEH13]	43	16	19	49	2.7009E-12
GeneAS [DG+96]	33	14	17	50	1.362E-9
CS [GYA13]	43	16	19	49	2.7009 E-12
PSO [PV05]	43	16	19	49	2.701E-12
Sequential linearization approach [LP91]	42	16	19	50	0.23E-6
Mixed-discrete nonlinear optimization with SA [ZW93]	52	15	30	60	2.36E-9
Nonlinear integer and discrete programming [San90]	45	22	18	60	5.712E-6
Mixed integer discrete continuous optimization [KK94]	33	15	13	41	2.146E-8
Mixed-variable evolutionary programming [CW97]	52	15	30	60	2.36E-9
GA [WC95]	NA	NA	NA	NA	2.33E-7
Mixed integer discrete continuous programming [FFC91]	47	29	14	59	4.5E-6
BOA [AS19]	43	16	19	49	2.701E-12
ISA [Gan14]	NA	NA	NA	NA	2.7009E-12

NA = Not available
License: CC BY 4.0

4.6.2.1 Comparison with other optimizers

We utilize `MoonLight`, a tool designed for monitoring temporal and spatial-temporal properties of CPSs [BBL+20], to identify counterexamples for ${\phi_1}^{AT}$. Considering the need for robustness values in solving the optimization problem, we adopt the *quantitative* semantics in this study. Our approach to falsify the STL formula ${\phi_1}^{AT}$ involves setting the Brake input as constant (kept at 0) while defining the throttle input as a step signal characterized by the *step time*, *initial value*, and *final value*. Additionally, we employ BCA to tackle the optimization problem and evaluate its performance against three established meta-heuristic techniques: GA, DE, and PSO. The parameter configurations for these algorithms are detailed in Table 4.4.

Utilizing the `MoonLight` tool, we acquire counterexamples using each optimization algorithm. Figure 4.10 illustrates the throttle input alongside the corresponding output speed and RPM for the counterexample pertaining to ${\phi_1}^{AT}$ obtained through BCA. The robustness values obtained by BCA and other algorithms, along with the computation time (i.e., the time required to falsify the property, in seconds), are summarized in Table 4.20. Two metrics are provided for each tool/algorithm: (1) *Robustness value*: This metric quantifies the robustness of the simulation trace with respect to the STL property. A lower (more negative) robustness value signifies better counterexamples, indicating that the approach has found more effective counterexamples, and (2) *Computation Time* (in

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

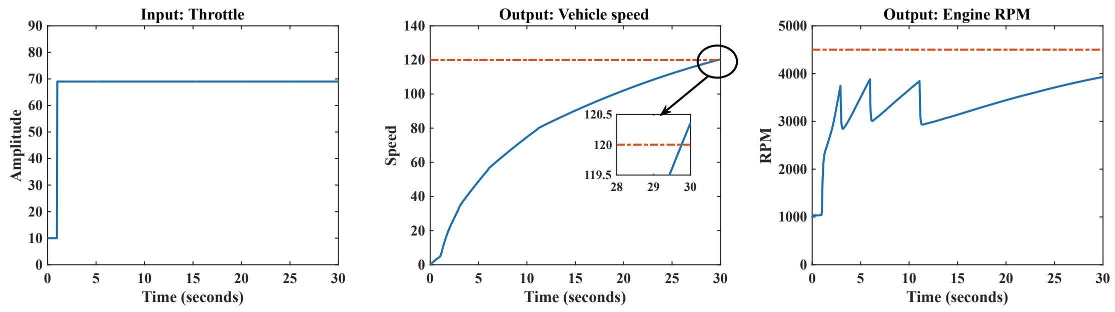


Figure 4.10: Input and output plots for the counterexample for ${\phi_1}^{AT}(120, 4500)$ found by BCA. The dash-dot lines show the thresholds of the outputs.

seconds): This indicates the time taken by each approach to find the counterexamples, with lower values representing faster performance. Notably, BCA exhibits faster convergence and surpasses other optimizers in performance. These findings underscore the competitive edge of the proposed BCA in tackling the falsification challenges of CPSs, underscoring its efficacy in addressing real-world optimization tasks, including search-based testing of CPSs.

Table 4.20: Results for the falsification problem of ATCS.

Approach	Robustness value	Computation Time (s)
BCA	-0.3564	28.3567
PSO	-0.7236	56.4472
GA	-1.6134	62.6923
DE	-1.7380	65.5578

4.6.2.2 Comparison with S-TaLiRo

We also employ the S-TaLiRo tool, an open-source falsification testing toolbox for CPSs, featured in the ARCH competition, as outlined in [DZS+15]. This tool enhances the performance of falsification methods by applying coverage metrics to the state space of hybrid systems.

We compare the results from S-TaLiRo with those obtained using BCA. For the BCA-based falsification testing (FT) experiments, we utilize the RTAMT tool [NY20] for monitoring and offline evaluation of STL properties. To ensure a fair comparison between FT-BCA and FT-S-TaLiRo, we terminate the process as soon as a counterexample is identified. Note that S-TaLiRo provides several stochastic optimization algorithms to choose from, including Simulated Annealing (SA), Ant Colony Optimization (ACO), Genetic Algorithms (GA), and Cross Entropy (CE). We present the results of the BCA-based falsification testing experiments and compare them with all four optimization

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

algorithms available in S-TaLiRo. Our results are summarized in Table 4.21. Note that the results presented are averaged over ten independent runs.

Table 4.21: Comparison of BCA-based falsification testing with S-TaLiRo.

Tool	**Algorithm**	**Performance measure**	
		Robustness value	**Computation Time (s)**
S-TaLiRo	SA	-0.0252	40.1423
	ACO	-0.0567	38.5672
	GA	-0.1435	46.3876
	CE	-0.0736	44.9159
	BCA	-0.1615	31.2671

We note that the BCA requires 31.2671 seconds to complete the testing, which is faster compared to the computation times of the algorithms in S-TaLiRo. Specifically, SA, ACO, GA, and CE take 40.1423, 38.5672, 46.3876, and 44.9159 seconds, respectively. The lower computation time of BCA implies that it is not only more effective in identifying counterexamples but also more efficient, potentially providing faster results with less computational effort. The results suggest that the BCA-based approach is both more effective and efficient compared to the stochastic optimization algorithms implemented in S-TaLiRo. It achieves better robustness values and has a shorter computation time, making it a preferable choice for falsification testing in terms of both performance and efficiency.

Collectively, the results outlined in this chapter substantiate the efficacy and reliability of the proposed BCA as a robust alternative to established meta-heuristic algorithms.

4.7 Conclusion

We introduced the Blood Coagulation Algorithm (BCA), a novel bio-inspired, population-based optimization algorithm that mimics the natural process of blood coagulation in the human body. To evaluate its performance, we conducted an extensive study using 23 mathematical benchmark functions, comparing BCA against 12 state-of-the-art algorithms. Our investigation covered intensification, diversification, local optima avoidance, and convergence behavior using both unimodal and multimodal functions. The simulation and statistical results demonstrate that BCA is competitive with other leading algorithms. Its robust performance on high-dimensional functions suggests that BCA is suitable for a wide range of optimization tasks. Furthermore, BCA's effectiveness was validated through six constrained and unconstrained engineering design problems, where it consistently outperformed other meta-heuristic algorithms. Additionally, we tested BCA for the falsification testing of CPS. The results indicated that BCA converges faster

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

and delivers better performance than other optimizers. This makes BCA particularly well-suited for the optimization challenges frequently encountered in CPS testing, such as ensuring system reliability, safety, and performance under various conditions. In summary, BCA not only proves its proficiency over other powerful meta-heuristic algorithms but also shows great potential for solving a variety of benchmark and real-world problems, especially in the domain of CPS testing.

Data Availability Statement. The experimental data and scripts are publicly accessible at https://gitlab.com/DrishtiYadav/bca.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

CHAPTER 5

Fault Localization

Fault localization stands as a pivotal component in the development, validation, and troubleshooting phases of CPS. However, finding the exact location of a fault that triggered a failure within a CPS model poses a formidable challenge, largely owing to the intricate structure and data-flow dynamics inherent in CPS models. In this chapter, we introduce a methodology that leverages formal specifications alongside search-based testing to achieve accurate fault localization. Given a CPS Simulink model, a formalized requirement serving as a test oracle, and a test case exhibiting failure against the formalized property, we develop a procedure that utilizes search-based testing to generate an additional test case that satisfies the same formalized property. Subsequently, we compare our two similar test cases with opposite verdicts to find the precise fault location. We implement our approach and assess its effectiveness through three case studies from automotive and avionic domains. Through empirical comparison against a state-of-the-art fault localization technique, we demonstrate that our method not only significantly reduces the number of suspected model variables and blocks compared to the previous work, but also remains resilient even with an increased number of active faults in the underlying models.

5.1 Introduction

While MBD streamlines various design tasks, debugging faulty models remains a labor-intensive and challenging process. It entails the meticulous identification of fault locations,

©2022 IEEE. Parts Reproduced, Reused, Reprinted, with permission, from Ezio Bartocci, Leonardo Mariani, Dejan Ničković and Drishti Yadav, "Search-based Testing for Accurate Fault Localization in CPS," 2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE), Charlotte, NC, USA, 2022, pp. 145-156, https://doi.org/10.1109/ISSRE55969.2022.00024 often requiring substantial expertise from engineers to uncover the root cause of failures. Rapid and accurate detection and diagnosis of faults are crucial for ensuring the continuous and reliable operation of the system. This is particularly essential during the early stages of CPS design, as undetected failures in safety-critical CPS can lead to significant costs and potentially catastrophic consequences [LNB+17, LNB+16, BFMN18]. Thus, precise fault localization is paramount as it can greatly expedite model correction processes and facilitate the development of safe CPS designs.

Simulation-based testing and its derivatives, like falsification testing, are effective methods utilized to detect defects in CPS design efficiently. Falsification testing, as documented in several studies [Don10, ALFS11, ZSD+15, SF12], employs a formal language, such as STL [MN04], along with its quantitative robustness semantics, to direct the search for tests that violate the given specification. While simulation-based testing is effective in identifying erroneous behavior in the model, it typically offers only the observed behavior as an explanation for the specification violation. More recently, methods for fault localization and explanation have emerged to aid in debugging MATLAB/Simulink models [BMM+21, SS20, BFMN18, BMM+19, LNB+17]. These techniques, known as gray-box procedures, employ various heuristics to pinpoint the fault, thus narrowing down the debugging scope. A common characteristic among existing localization methods is their detachment from testing activities. Consequently, the accuracy of localization heavily relies on the quality of the test suite.

In procedural and object-oriented programming environments, fault localization has been studied as the task of contrasting two comparable test executions with opposite outcomes [ZJP+14, RR03], one passing and the other failing. However, these methods rely on test generation strategies and heuristics that are not directly applicable to data-flow-oriented computational models like CPS Simulink models.

Contributions. We propose a fault localization method closely integrated with falsification testing, specifically designed for CPS Simulink models with explicit or implicit specifications. We utilize STL as the formal language for expressing CPS requirements. When explicit specifications are unavailable, we assume the existence of a *correct reference model* and establish its *equivalence* to the model under test as our implicit specification. In both scenarios, we employ the formal specification as a test oracle. For a test case which fails based on the formalized property, we develop a process to generate an alternative passing test. This passing test helps pinpoint the fault location when compared to the failing test. Essentially, we employ a global optimizer for search-based testing to generate a new passing test case that closely resembles the original failing test, using a specified distance measure. By examining the outcomes of these two similar tests with opposing results, we extract precise fault location information, reducing the scope and effort required for debugging. Our method categorizes and prioritizes suspicious variables based on their timing and severity of violation, presenting them alongside their corresponding model blocks. This list of suspicious blocks assists engineers in localizing multiple faults, potentially of various types, within the model.

In summary, our primary contributions include: **(1)** Introducing a novel *search-based*

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

approach for automatically generating a passing test closely resembling a failing test for (data-flow) Simulink models, and **(2)** Proposing a *fault localization method* for Simulink models that leverages a failing test and its corresponding closely resembling passing test.

We apply our approach to three case studies spanning various application domains. We implement our approach using both explicit STL specifications and implicit equivalence checking, assessing its performance across 240 faulty variants of the case study models harboring multiple faults. To gauge its effectiveness, we compare it against `CPSDebug` [BMM+21, BMM+19], a leading fault localization technique for CPSs. The experimental outcomes reveal that our approach outperforms the baseline in two key aspects: (1) efficiently reducing the number of suspicious variables and thereby minimizing the blocks requiring inspection for fault localization, and (2) accurately pinpointing fault locations in models with multiple faults. Particularly noteworthy is its superior performance in accurately localizing faults in models with multiple underlying faults, where the baseline exhibits diminished accuracy.

Chapter Organization. We introduce our fault localization approach in Sections 5.2 and 5.3. Following this, an empirical evaluation and a summary of the results are presented in Section 5.4. Section 5.5 offers an enhanced review of related works. Our conclusions are drawn in Section 5.6.

5.2 STL-guided Fault Localization

In this section, we elucidate the operational mechanism of our specification-guided fault localization approach, referred to as `STL-FL`. Figure 5.1 provides a visual depiction of the two core phases constituting the fault localization workflow:

1. *Testing*: This phase begins by assessing the SUT[1] using an initial test suite to identify failing test cases. Subsequently, a global optimizer is employed to create a new passing test case for each failed test case, guided by the STL specification ϕ. The objective is to generate a new passing test case that closely resembles its corresponding failing test.
2. *Localizing*: In this phase, simulation outputs from the failed and passing test case pair are utilized to pinpoint the exact location of the faulty component(s).

5.2.1 Testing

The objective of the testing phase is to select a collection of pairs, each consisting of a failing and a passing test case, which serve as representatives for fault localization. These pairs should exhibit similar executions, with any disparity likely attributed solely to the

[1] In contrast to stochastic systems where randomness is present in one or more system parts (such as system parameters drawn from probability distributions), the systems considered in this thesis are deterministic, meaning they produce the same output for a given input.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

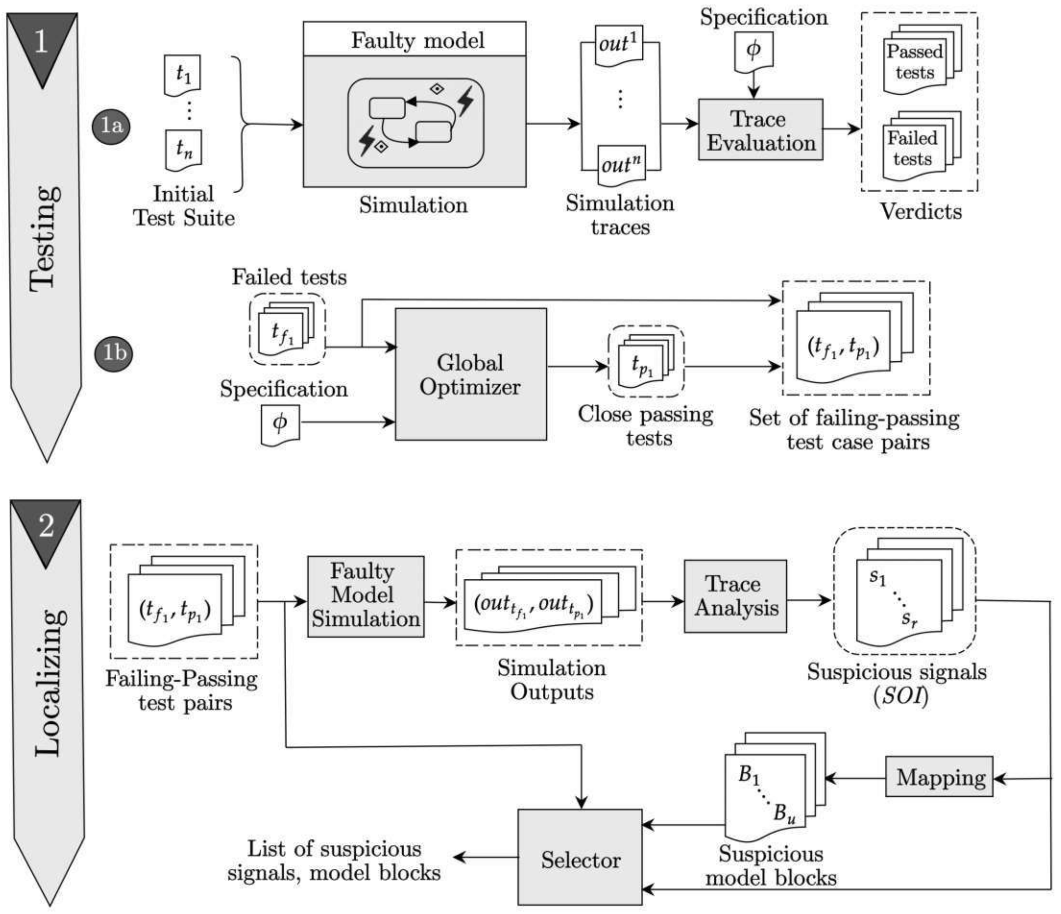


Figure 5.1: An overview of STL-guided fault localization procedure (STL-FL). ©2022 IEEE.

activation of the fault under investigation. This enables a detailed (and point-by-point) comparison of these executions, aiding in the precise localization of the fault.

During the testing phase, we find these pairs by first spotting the failing tests —those that do not meet the given STL specification —from the available tests. Then, we create new passing tests that get gradually closer to the failing one until we find the best candidate for locating the fault.

We start by assuming that there is an initial test suite, denoted as $\mathcal{TS}$, which has at least one test case that violates a formalized requirement. This test suite can be provided manually by a tester or generated using automated tools. These automated methods might use coverage-based techniques [LNLB19], considering input, output, or structural coverage. Alternatively, they might focus on methods for efficiently generating failing test cases, like falsification testing [NSF+10].

Algorithm 5.1 delineates the process for choosing the test pairs. Beginning with an initial

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Test Suite $\mathcal{TS}$, the available test cases undergo evaluation against a faulty model. The faulty model may have either a single fault or multiple faults. Each test case t from $\mathcal{TS}$ is then used to simulate the faulty model to produce the simulation results (Line 4). Subsequently, the output traces are assessed against the STL specification ϕ by a $Monitor$ to assign either a *pass* or *fail* verdict to the test case (Lines 5-7). The $Monitor$ performs 'Trace evaluation' (refer to Figure 5.1) and generates a robustness measure of the trace concerning ϕ (Line 5). The objective is to identify all the test cases leading to the violation of the property ϕ: the algorithm groups all failing test cases resulting in observable failures in the model (Line 7).

Algorithm 5.1: Test Suite Selection for localizing faults.

```
   Input  : $\mathcal{TS}$ : An initial test suite.
            $\mathcal{M}_F$ : A faulty model.
            $\phi$ : An STL specification.
   Output : $\mathcal{TS}_{FL}$ : Set of failing-passing test case pairs.
1  $\mathcal{TS}_{New} = [\,]$
2  $\mathcal{TS}_{FL} = [\,]$
3  for each item $t \in \mathcal{TS}$ do
4      $out(\mathcal{M}_F, t) = Simulate(\mathcal{M}_F, t)$
5      $R(out(\mathcal{M}_F, t), \phi) = Monitor(out(\mathcal{M}_F, t), \phi)$
6      if $R(out(\mathcal{M}_F, t), \phi) < 0$ then
7          $\mathcal{TS}_{New} = \mathcal{TS}_{New} \cup \{t\}$
8      end if
9  end for
10 for each item $t_f \in \mathcal{TS}_{New}$ do
11     $t_p \leftarrow \textsc{SearchPT}(t_f, \mathcal{M}_F, \phi)$
12     $\mathcal{TS}_{FL} = \mathcal{TS}_{FL} \cup \{(t_f, t_p)\}$
13 end for
14 return $\mathcal{TS}_{FL}$
```

For each failing test case, the algorithm seeks a nearby passing test case that could aid in localization. Specifically, for every failing test $t_f \in \mathcal{TS}_{New}$, the algorithm hunts for a passing test case t_p that closely resembles or is more similar to the failing test case t_f (Line 11). Our SEARCHPT() subroutine is depicted in Algorithm 5.2. The search is framed as an optimization challenge to find a passing test t_p that closely aligns with the failing test t_f, ensuring that the distance between t_p and t_f is minimized. This problem is formulated as follows:

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

Close passing test case search problem

Input: a faulty Simulink model $\mathcal{M}_F$ with a failed test case t_f and a formula ϕ such that $out(\mathcal{M}_F, t_f) \not\models \phi$.
Objective: Determine t_p such that $out(\mathcal{M}_F, t_p) \models \phi$, and $D(t_f, t_p)$ is minimized.

In our study, we utilize our BCA optimizer [Yad21], as elaborated in Chapter 4, for the search operation. The pivotal aspect of determining the most analogous passing test lies in defining the distance $D(t_f, t_p)$ between the failing and passing test cases for the faulty model $\mathcal{M}_F$. We compute this distance $D(t_f, t_p)$ using the Euclidean distance (also known as the *Euclidean norm* or L^2 *norm*) between the signals associated with the test cases t_f and t_p. The Euclidean distance stands out as one of the most commonly used distance measures in real vector spaces, offering the advantage of gauging similarities by quantifying the conventional distance between data points. Given that CPS models frequently involve continuous real-valued variables, the Euclidean distance emerges as a suitable choice for our analysis. Formally, the Euclidean distance D between two signals $\mathbf{y}$ and $\mathbf{z}$ is expressed as:

$$D(\mathbf{y}, \mathbf{z}) = ||\mathbf{y} - \mathbf{z}||_2 = \sqrt{\sum_{i=1}^{g} (y_i - z_i)^2}$$

In the above equation, $\mathbf{y}$ and $\mathbf{z}$ represent signals with a finite length of g samples, where $\mathbf{y} = (y_1, \cdots, y_g)$ and $\mathbf{z} = (z_1, \cdots, z_g)$. Generally, the number of samples in a signal is determined by the simulation and relies on the *step-size*, also known as the sample time, of the model. As the simulations for test cases t_f and t_p are conducted on the same faulty model, the timestamps (and consequently, the signal lengths) are ensured to match across both executions. The outcome of the testing phase is denoted as $\mathcal{TS}_{FL}$ (Line 14), which comprises pairs of failing test cases and their corresponding close passing test cases for the faulty model $\mathcal{M}_F$.

Let us delve into the SearchPT() subroutine illustrated in Algorithm 5.2, which introduces the adaptation of BCA to address the formulated 'Close passing test case search' task. Among various optimization algorithms available, we opted for BCA due to several reasons: (1) its parameters are intuitively interpretable and require no tuning effort, (2) it boasts a straightforward implementation, and (3) it exhibits high computational efficiency. Additionally, we conducted performance evaluations comparing BCA with other state-of-the-art optimizers, such as Particle Swarm Optimizer, Harmony Search, Water Cycle Algorithm, and Imperialist Competitive Algorithm, for our optimization problem. Our observations revealed that BCA surpasses others in terms of convergence and speed. Below is an overview of BCA, which is a population-based optimizer, as discussed previously in Chapter 4.

Definition 5.1. BCA follows these steps to explore the global optimum solution of an objective function $f(x)$ with n variables:

1. It initializes a population of candidates uniformly distributed across the search space.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

2. It evaluates the fitness of each candidate and identifies the fittest candidate based on $f(x)$.
3. It updates the position of each candidate using specific criteria and optimizer parameters. The fitness values are recalculated, and the global best solution is updated if a superior solution is discovered. This process iterates until termination criteria are satisfied.

In Algorithm 5.2, we commence by creating an initial population of candidates (Line 2) and subsequently determining the fitness of each candidate, corresponding to the test case (Line 3). The individuals within the initial population are generated by producing test inputs that uniformly cover the numerical input range of the SUT. It is important to note that *genPop* (Line 2) mirrors the first step of BCA as outlined in *Definition 5.1*, ensuring that the initial population comprises a non-empty collection of test candidates, the size of which is specified by the user. Currently, we focus solely on numerical input domains since they are predominant in CPS models. However, other input domains could also be addressed, provided a distance metric is available for their domain. For example, a Boolean distance function could be employed for enumerated signals.

Algorithm 5.2: The SEARCHPT() subroutine.

```
Input  : $\mathcal{M}_F$ : A faulty model.
         $\phi$ : An STL specification.
         $t_f$ : A failing test case.
Output : $t_p$ : Close passing test case.
1  Initialize optimizer parameters
2  InitPop $\leftarrow$ $genPop()$ ;                          // Initial population
3  FitPop $\leftarrow$ $Cost(\text{InitPop}, \phi)$ ;                  // fitness
4  $t_p \leftarrow Best(\text{FitPop})$ ;                            // best solution
5  $D_{max} = D(t_f, t_p)$ ;                              // Initial Distance
6  while TimeOut() do
7      for each candidate $u \in$ InitPop do
8          $u_{new} \leftarrow Update(u)$
9      end for
10     FitPop $\leftarrow$ $Cost(\text{InitPop}, \phi)$
11     $D \leftarrow D(t_f, Best(\text{FitPop}))$
12     if $D < D_{max}$ then
13         $D_{max} \leftarrow D$ ;                          // update the distance
14         Update $t_p$ ;                                // new best solution
15     end if
16 end while
17 return $t_p$
```

In this context, the *fitness* for each candidate within the population denotes the quantitative measure of how robustly the execution trace of the faulty model $\mathcal{M}_F$ adheres

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

to the STL specification ϕ. Line 4 identifies the optimal test case from the population that (1) is close to the failing test case t_f and (2) satisfies ϕ. Line 5 calculates the initial Euclidean distance between t_f and the best solution t_p, which serves as the algorithm's selection criterion. It is noteworthy that the algorithm iteratively updates the test cases (Line 8) and acknowledges a new solution as the best one if it meets the selection criteria, i.e., a closer passing test is found (Lines 12-15). The subroutine is designed to terminate under certain conditions, as indicated by the guard for the while loop (Line 6), including reaching the maximum number of iterations, observing no improvement after a set number of iterations, or reaching a timeout.

5.2.2 Localizing faults

Algorithm 5.3 presents our method for localizing faults by examining anomalous events in the SUT under the guidance of an STL specification. The inputs consist of a defective (aka faulty) model and a Test Suite $\mathcal{TS}_{FL}$ containing pairs of failing and passing test cases. The fault localization algorithm operates on the premise that signals exhibiting early temporal discrepancies between passing and failing test runs likely stem from faults within the blocks generating these signals. The fault localization algorithm effectively identifies model variables displaying significant early-time deviations between the failing and closely passing test cases, associates these variables with the respective blocks, and generates a list of suspicious blocks to aid in model debugging and rectification.

Algorithm 5.3: Fault Localization by analyzing anomalous events with STL.

```
Input  : $\mathcal{M}_F$ - A faulty model.
         $\mathcal{TS}_{FL}$ - Set of failing-passing test case pairs.
Output : $SOI_{best}$ - Signals of interest.
         $blockList_{best}$ - List of suspicious blocks.
1  for each pair $(t_f, t_p) \in \mathcal{TS}_{FL}$ do
2      $out(\mathcal{M}_F, t_f) = Simulate(\mathcal{M}_F, t_f)$
3      $out(\mathcal{M}_F, t_p) = Simulate(\mathcal{M}_F, t_p)$
4      $d = |out(\mathcal{M}_F, t_f) - out(\mathcal{M}_F, t_p)|$
5      $d_m, q_{viol} \leftarrow \textsc{Eval}(d)$
6      $d_{new} = Normalize(d_m)$
7      $SOI \leftarrow GetSignalInfo(d_{new})$
8      $blockList \leftarrow Map(SOI)$
9  end for
10 $SOI_{best}, blockList_{best} = \Pi(\mathcal{TS}_{FL}, \mathbf{SOI}, \mathbf{blockList})$
11 return $SOI_{best}, blockList_{best}$
```

Engineers initiate the process with a test suite $\mathcal{TS}_{FL}$ comprising pairs of failing-passing test cases. Lines 2-3 execute the task illustrated in the 'Faulty Model Simulation' block depicted in Figure 5.1. Specifically, Line 2 conducts a simulation of the faulty model using the failing test case t_f, yielding the simulation output for each model variable. Line

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

3 conducts a simulation of the faulty model using the passing test case t_p to generate the simulation output.

Lines 4-7 execute the process depicted as 'Trace Analysis' in Figure 5.1. Specifically, Line 4 calculates the deviations between the simulation traces obtained from the failing and passing tests. It is important to note that we assume each trace is sampled periodically, enabling direct point-by-point comparison between the two traces. Typically, $out(\mathcal{M}_F, t)$ represents an array of size $Y \times Z$, where Y denotes the total number of samples of the recorded signals, and Z is the total number of recorded signals. The deviation d (also an array of size $Y \times Z$) indicates the absolute differences between the Z recorded signals for t_f and t_p. It is worth mentioning that the 'minus' sign in Line 4 signifies element-wise subtraction of the values of each recorded signal at every timestamp for test case t_p from those of test case t_f.

In Line 5 of Algorithm 5.3, the function Eval utilizes the calculated deviation d and identifies the timestamp q_{viol} when the initial irregularity occurs. Additionally, it organizes all deviation values linked to q_{viol} in descending order internally and outputs d_m, which is an array sized $1 \times Z$. More precisely, d_m represents a sorted series of deviation values for all recorded signals observed at the first instance of violation, q_{viol}.

The function $Normalize$ computes the vector-wise z-score of d_m with a mean of 0 and a standard deviation of 1 (Line 6). As a result, d_{new} becomes an array sized $1 \times Z$ containing normalized deviation values (observed first in time) for each recorded signal. Normalization of data is applied to standardize the deviation values to a common scale, ensuring that all measured deviations fall within the same numerical range.

In Line 7, the function $GetSignalInfo$ retrieves the logging details of signals exhibiting initial misbehavior, identified by *normalized deviations exceeding zero*. This condition, based on normalized deviation values, enables focusing solely on a limited set of signals demonstrating notably high deviations. (It is worth noting that q_{viol} denotes the moment when the first deviation occurs.) This process helps localize all internal signals within the model accountable for the faulty behavior. Following this, Line 8 executes the function Map (analogous to 'Mapping' in Figure 5.1), which correlates the *Signals of Interest* (SOI) with their respective model blocks to pinpoint the suspicious blocks. The SOI represents model variables (internal signals) that exhibit early and significant deviations between t_f and t_p in the faulty model, essentially serving as indicators of the faults.

Finally, the failing-passing test case pair that results in the *minimum* number of SOI is selected as the optimal pair. This selection process is handled by the function Π (Line 10), referred to as 'Selector' in Figure 5.1. It is important to note that **SOI** and **blockList** represent the sets of SOI and $blockList$ linked with each pair $(t_f, t_p) \in \mathcal{TS}_{FL}$. For the chosen optimal pair, the variables SOI_{best} and $blockList_{best}$ are designated as the final output, aiding in pinpointing the "culprit" signals and components responsible for the failure, thereby localizing the faulty units (Line 11). Additionally, the violation time (q_{viol}) may also be provided as output to aid in understanding the context of the failure and its timing.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 5.1: Details of Failing-Passing test case pair for Example 1.

Test case	Throttle parameters			Robustness R	$D(t_f, t_p)$
	ST	IV	FV		
t_f	1	10	68	−1.8022	80.8332
t_p	1	10	65	0.8242	

©2022 IEEE.

Table 5.2: *SOI* for fault localization and the corresponding blocks.

Faulty variant	Signal Index	Signal Name	Parent Block	$q_{viol}(s)$
Offset fault in *Engine*	s_2	EngineTorque:1 → Offset1:1	Engine	1.00

Note: q_{viol} denotes the first time that the signal exhibits anomalous behavior w.r.t. the test case pair (t_f, t_p).

©2022 IEEE.

Example 5.1. Here, we illustrate the application of `STL-FL` with an example. We will utilize the Simulink model of an *Automatic Transmission Controller System*, which is elaborated in Section 5.4. For our demonstration, we will consider a faulty variant with a Bias/Offset fault, where the fault value is set to 10 units. This fault is injected into the signal path from the *EngineTorque* block to the *Sum* block within the *Engine* subsystem of the model.

Assuming a constant brake signal (set to zero), we identify a failing test case t_f for the model concerning the corresponding STL property ϕ outlined in Table 5.3. In this scenario, the test case reflects the throttle input signal. We suppose that the throttle signal is a step signal defined by three parameters: *Step Time* (ST), *Initial Value* (IV), and *Final Value* (FV). Employing our proposed 'Close passing test case search problem' detailed in Algorithm 5.2, we locate a passing test case t_p (refer to Table 5.1).

Utilizing Algorithm 5.3, we identified the *SOI* to isolate the faulty component. We note that only a single internal signal exhibits early misbehavior with normalized deviations exceeding zero, as elaborated in Table 5.2. This identified *SOI* stem from the *Engine* subsystem, precisely where we introduced the fault, thus confirming accurate localization of the faulty component.

5.3 Fault Localization with Equivalence Checking

In numerous CPS scenarios, explicit specifications are absent. In such cases, throughout different development stages, designers typically assess the design against a *reference model* assumed or verified to be accurate. Equivalence testing, also known as *back-to-back testing*, *differential testing*, or *differential fuzzing* in software engineering, is commonly employed to ascertain equivalence between the reference model $\mathcal{M}$ and its updated or refined version $\mathcal{M}'$. Equivalence is established by comparing the output signals produced

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

by simulating both models using a single (equivalence) test case. Specifically, the recorded output signals serve as the equivalence criteria between $\mathcal{M}$ and $\mathcal{M}'$, representing an implicit specification formally linking the two models. The non-equivalence between two models can be articulated as:

$$\underbrace{\bigwedge_{i=1}^{m} In_1.i = In_2.i}_{\text{all inputs equal}} \wedge \underbrace{\bigvee_{o=1}^{e} out_1.o \neq out_2.o}_{\text{an output not equal}} \tag{5.1}$$

In this context, In denotes the collection of model inputs with a length of m, while out signifies the set of traces for all input-internal-output (IIO) signals of the models with e outputs. Referring to Figure 2.1, $m = 2$ and $e = 3$. Any configuration that fulfills Formula (5.1) denotes two separate runs of $\mathcal{M}$ and $\mathcal{M}'$ that yield a distinct output signal for the identical input sequence.

Our method also includes support for equivalence checking. Similar to the procedure outlined in STL-FL, the workflow of our equivalence-driven approach, referred to as E-FL, initiates with testing, as elaborated in the preceding section. In this scenario, a failing test signifies evidence of non-equivalence between the two models. For each failing test, our objective is to locate the closest test case that produces an output most akin to the failing test case. We employ a distance metric $dist$ to assess the resemblance between two outputs. We treat the search task as an optimization problem formulated as follows: "Given a faulty model $\mathcal{M}_F$ with a failing test case t_f, determine t_c (similar to t_f) such that $dist(c_1, c_2) < \theta$ where $c_1 = out.o(\mathcal{M}_F, t_f)$ and $c_2 = out.o(\mathcal{M}_F, t_c)$".

To expedite the convergence of the search task utilizing the BCA optimizer, we enforce a constraint on the distance between the two outputs, stipulating $dist(c_1, c_2) < \theta$ where θ is set to 0.05. The localization process mirrors that employed in STL-FL (refer to Lines 1-9 in Algorithm 5.3), albeit with the analysis of deviations conducted on the simulation outputs of the failing test case t_f and the newly identified test case t_c.

5.4 Empirical Evaluation

In this section, we outline our research questions, detail our experimental configuration, specify the metrics used for evaluation, and present the results of our experiments. The following are the key research questions we aim to address:

RQ-5.1. [Fault Localization Ability] *To what extent does our method reduce the number of signals and blocks requiring inspection for fault localization?* We assess the effectiveness of our approach in minimizing the set of suspicious model variables and blocks to be examined for fault localization. Specifically, we analyze how well our method performs in this regard and compare our findings with those of the baseline technique, CPSDebug [BMM+21], a state-of-the-art automated fault localization method for CPS models.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

RQ-5.2. [Robustness Evaluation] *How robust is our approach in scenarios where the SUT exhibits numerous faults, potentially of diverse types?* We investigate the resilience of our approach by assessing its fault localization capabilities in the presence of *multi-fault models*, comparing its performance against the baseline technique.

RQ-5.3. [Computational Efficiency] *How does the computational efficiency of our approach compare to that of the baseline technique?* We gauge the computational efficiency of our method by examining its computation time and comparing it with the performance of the baseline technique.

To empirically assess our approach and address the research questions, we conducted systematic experiments utilizing Simulink models spanning various safety-critical domains. We devised a prototype implementation for our fault localization technique, as elucidated in Section 5.2. The processes encompassing test generation, model simulation, and model-based fault localization were realized within MATLAB. Additionally, for offline assessment of STL properties, we employed the RTAMT tool [NY20], augmenting it with our fault localization procedure outlined in Section 5.2.

5.4.1 CPSDebug

We now present a concise overview of our baseline method, `CPSDebug` [BMM+21], recognized as a state-of-the-art diagnostic solution tailored for Simulink models. `CPSDebug` serves the purpose of localizing bugs within CPS designs and elucidating the root causes of failures through a structured workflow comprising three primary phases.

During the *testing* phase, `CPSDebug` conducts simulations of the CPS Simulink model under analysis using an initial test suite. Subsequently, it categorizes the test cases into passing and failing ones based on their compliance with formal STL requirements.

Moving to the *specification mining* phase, `CPSDebug` leverages the passing test traces to extract relevant properties that delineate the expected behavior of the model. This process involves the utilization of two key tools: Daikon [ECGN01], which employs template-based property inference to deduce likely properties for the input variables, and Timed k-Tail (TkT) [PMGM22], an engine for automaton learning capable of generating timed automata from timed traces.

Lastly, in the *explaining* phase, `CPSDebug` utilizes the inferred properties to analyze failed traces and generate explanations for the detected failures. This phase encompasses two main steps: (1) *Monitoring*, wherein `CPSDebug` analyzes failed traces to identify signals violating the properties along with the corresponding violation time intervals, and (2) *Clustering and Mapping*, where `CPSDebug` clusters the fail-annotated signals based on their violation times and associates them with their respective model blocks. Ultimately, `CPSDebug` employs the violated signals and their originating blocks to produce a sequence of system snapshots for each cluster of property violations, thereby capturing the origins of faults and the propagation of failures in both spatial and temporal dimensions.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 5.3: Key information about the Simulink models of our subjects.

Model Name	*#Blocks*	*#Lines*	ϕ	q_T
Automatic Transmission Controller System (ATCS)	65	92	$\square((v \leq \bar{v}) \wedge (\omega \leq \bar{\omega}))$	30
Aircraft Elevator Control System (AECS)	825	577	$\square(\uparrow (cmd \geq m) \rightarrow \Diamond_{[0,T]}\square_{[0,a]}(\lvert cmd - pos\rvert \leq n))$	10
Abstract Fuel Control System (AFCS)	253	283	$\square_{[T_{start},T_{stop}]}\neg((AF > Ref - tol) \vee (AF < -Ref + tol))$	40

©2022 IEEE.

5.4.2 Experimental Setup

We conducted the experiments on a MacBook Pro equipped with an Apple M1 chip, 16 GB RAM, and running macOS Monterey with MATLAB® R2018b. This section provides an overview of the subjects utilized for our experiments and evaluations. Additionally, we detail the process of fault seeding, as well as the test suites and test oracles employed in our experimental setup.

5.4.2.1 Case studies

For our experiments, we employ three Simulink models sourced from the automotive and avionic sectors. Table 5.3 presents the dimensions of each model, along with the corresponding STL specifications utilized in the experiments and the simulation time denoted by q_T. The STL specifications referenced in Table 5.3 are sourced from [Don10, HAF14, BMM+21].

The **Automatic Transmission Controller System (ATCS)**, as previously discussed in Chapter 4, is a benchmark derived from the automotive field [HAF14]. This model incorporates two inputs, namely the throttle u_t and the brake u_b, which regulate two system outputs: vehicle speed v (mph) and engine speed ω (RPM). Throughout all time instants, the inputs span the range [0, 100]. A critical safety requirement for this system is to ensure that the vehicle speed v and engine speed ω do not surpass predefined thresholds $\bar{v}$ and $\bar{\omega}$. These thresholds are set as follows: $\bar{v} = 120$ mph and $\bar{\omega} = 4500$ RPM.

The **Aircraft Elevator Control System (AECS)**, as previously discussed in Chapter 3, is a model originating from the avionics sector and features a redundant actuator control module [GM05]. This model involves two output variables corresponding to the positions of left and right actuators, driven by the input variable (Pilot Command). One significant requirement entails achieving the desired aircraft position within a predetermined timeframe. Formally, whenever the Pilot Command *cmd* exceeds a threshold m, the measured actuator position *pos* must stabilize (i.e., become at most n units away from *cmd*) within $T + a$ time units. This requirement is expressed via the STL specification in Table 5.3, with parameters set as $m = 0.09$, $T = 2$, $a = 1$, and $n = 0.02$.

The **Abstract Fuel Control System (AFCS)** is a renowned Powertrain Control Verification Benchmark [JDK+14] sourced from the automotive domain, simulating the air-fuel controller for an engine. This model accepts user inputs of throttle command and engine speed to regulate both the fuel rate and the air-to-fuel (AF) ratio. The

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

primary control objective is to uphold the AF ratio at the set-point Ref (i.e., the *ideal* stoichiometric ratio) through closed-loop adjustments of pertinent model variables. The STL requirement for this model encapsulates constraints on the AF ratio, with parameter values set as $Ref = 14.7$, $tol = 0.01$, $T_{start} = 10$, and $T_{stop} = 40$.

In our experiments, we established the base sample time for the models, namely ATCS, AECS, and AFCS, as 0.04, 0.01, and 0.005 respectively. These models are openly accessible in the Simulink/Stateflow online documentation provided by MathWorks [Mat22d, Mat22a, Mat22c]. It is noteworthy that these models are indicative of typical industrial Simulink models concerning their size, behavioral dynamics, and intricacy.

5.4.2.2 Fault Seeding

We utilize our software tool FIM [BMNY22a] (as elaborated in Chapter 3) to introduce faults into a model. Our selection of faults for injection was based on those commonly employed in the literature [FMMJ21, PRWN16, SVET10, SMSJ17]. Specifically, we considered the following categories:

- *Sensor faults*: Including Stuck-at, Noise, and Bias/Offset.
- *Hardware faults*: Encompassing Bit-flip (single and multiple).
- *Network faults*: Involving Package drop and Time delay.
- *Block mutations*: Involving incorrect relational operator, incorrect logical operator, and incorrect arithmetic operator.

By incorporating fault types from these diverse categories, we aimed to simulate a realistic spectrum of fault scenarios commonly encountered in practical systems, facilitating the evaluation of fault localization techniques. For each of the previously mentioned faults, FIM generates tailored fault blocks equipped with flags to regulate their activation. Specifically, FIM introduces faults into a model by (1) integrating fault blocks and (2) substituting blocks with fault blocks, leveraging a specialized library of faults and alterations. To achieve diversity, we introduced faults of various types in different segments (target locations) of the SUT, particularly at varying hierarchical depths. Table 5.4 provides insight into the number of seeded faults (and their corresponding groups) for each subject. In total, we seeded 15 faults in ATCS, 45 faults in AECS, and 20 faults in AFCS models.

Before subjecting the fault model to testing, we implemented instrumentation to facilitate the logging of all internal signals. This process involved assigning unique names to each signal and activating data logging within the simulation engine. By leveraging the activation flag embedded within the seeded fault blocks introduced by FIM, we could selectively enable or disable specific faults, thereby generating distinct faulty variants of the SUT. Each faulty variant represented an instrumented version of the faulty model with the activation of specific *faults of interest* (FOI). Depending on user specifications and the objectives of the testing task, FOI could range from a single fault to multiple (i.e., equal to or greater than two) faults.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 5.4: Information of seeded faults in each case study.

Type	ATCS	AECS	AFCS
Sensor faults	3	20	9
Hardware faults	-	17	6
Network faults	9	-	4
Block mutations	3	8	1
Total	15	45	20

©2022 IEEE.

To address **RQ-5.2**, we generated various versions of our subjects with different combinations of activated fault blocks. Specifically, we divided each subject into three sets of faulty variants, where the number of active fault blocks ranged from 1 to 3 in each set. For instance, for the ATCS model, each set comprised 15 faulty variants, while for the AECS model, each set contained 45 faulty variants, and for the AFCS model, each set included 20 faulty variants. Thus, we produced $\{45, 135, 60\}$ faulty variants of {ATCS, AECS, AFCS}, respectively, i.e., a total of 240 faulty variants. Additionally, to maintain fairness in our evaluation, we ensured that the activated fault blocks encompassed diverse types and were situated in different sections of the models.

5.4.2.3 Setup for Equivalence Testing

To assess the effectiveness of our approach in scenarios where STL specifications are unavailable, we designate the initial/original versions of ATCS, AECS, and AFCS as reference models. Their respective faulty variants serve as the models-under-test. Subsequently, we utilize the non-equivalence witness trace and employ the E-FL methodology to pinpoint the source(s) of the fault injected into the models.

5.4.2.4 Test Suite and Test Oracle

We employ Adaptive Random Testing (ART) as described in [LNLB19] to produce the initial test suite ($\mathcal{TS}$). ART is a foundational method that uniformly samples test cases within the valid input domains. Across each case study, we generate an initial $\mathcal{TS}$ comprising 100 test cases.

For STL-FL, the monitor derived from the STL specification via the RTAMT library serves as the test oracle. Conversely, for E-FL, we implemented a procedure that computes the distance between the behaviors of the reference and the tested models, which serves as our test oracle.

5.4.2.5 Evaluation metrics

We assessed the results of STL-FL and E-FL in the following aspects: Scope Reduction, Fault Localization Cost, Fault Localization Accuracy, and Computation Time. *Scope*

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Reduction evaluates our approach's effectiveness by measuring the degree to which it reduces the overall model variables to a smaller set of suspicious variables crucial for fault localization. *Fault Localization Cost* quantifies the absolute number of blocks necessitating engineers' attention to localize and rectify the fault. *Fault Localization Accuracy* evaluates the approach's efficacy in fault detection, determining whether the faulty component is correctly pinpointed. Lastly, we analyze the *computation time* required by our proposed approach.

5.4.3 Results

We present the experimental findings based on our evaluation criteria, using the ATCS model as an illustrative example. To provide clarity, we assess `STL-FL` and `E-FL` under the following test configurations:

- **Test Configuration 1:** [One-fault] A 'Stuck-at 0' fault introduced within the *Engine* subsystem.
- **Test Configuration 2:** [Two-fault] Same as Test 1, with an additional 'Time Delay' fault within the *Transmission/TransmissionRatio* subsystem.
- **Test Configuration 3:** [Three-fault] Building upon Test 2, including a 'Sum to Product mutation' within the *Engine* subsystem.

For each test configuration mentioned above, we employ both fault localization techniques, `STL-FL` and `E-FL`, to pinpoint the *SOI* and their associated blocks within the SUT. Subsequently, we assess the extent of reduction achieved in fault identification, as illustrated in Table 5.5. The "*Test*" column indicates the type of test configuration, while the "*Approach*" column specifies the fault localization method employed. In the "$\#SOI$ (Reduction in $\#Vars$)" column, we list the number of Signals of Interest, along with the reduction in the number of variables achieved, indicated within parentheses. Here, *SOI* refers to the suspicious variables exhibiting significant anomalous behavior at an early stage. The "FL_Cost (Reduction in $\#Blocks$)" column signifies the Fault Localization Cost (FL_Cost), with the corresponding reduction degree reported within parentheses. FL_Cost is determined by the total number of blocks necessitating inspection to localize the fault(s) within the faulty model. Lastly, the "*Fault(s) detected*" column specifies whether the faulty component(s), where the fault(s) were introduced, are correctly identified.

Looking at Table 5.5, we note that our approach effectively identifies the root cause of the fault by accurately pinpointing the faulty component across all three test configurations. In Test Configuration 1, `STL-FL` identifies only one *SOI*, indicating a significant reduction of 98.5% in the number of model variables. It is important to highlight that the reduction in variables and blocks is calculated based on the values specific to the instrumented fault model. Following fault injection and model instrumentation, the total number of blocks in ATCS is 77, and the total number of model variables is 67. As shown in Table 5.5, `E-FL` identifies two *SOI* in Test Configuration 1, achieving a reduction of 97%. It is worth emphasizing that focusing on a small subset of suspicious signals enables a

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 5.5: Scope reduction and fault detection in ATCS.

Test	Approach	*#SOI* (Reduction in *#Vars*)	*FL_Cost* (Reduction in *#Blocks*)	Fault(s) detected
1	`STL-FL`	1 (98.5%)	9 (88.3%)	✓
	`E-FL`	2 (97.0%)	9 (88.3%)	✓
	`CPSDebug`	8 (88.1%)	25 (67.5%)	✓
2	`STL-FL`	5 (92.5%)	24 (68.8%)	✓
	`E-FL`	7 (89.5%)	24 (68.8%)	✓
	`CPSDebug`	8 (88.1%)	25 (67.5%)	× (Only one faulty component identified)
3	`STL-FL`	12 (82.1%)	24 (68.8%)	✓
	`E-FL`	12 (82.1%)	24 (68.8%)	✓
	`CPSDebug`	9 (86.5%)	34 (55.8%)	× (Only one faulty component identified)

©2022 IEEE.

substantial decrease in the number of variables. This allows engineers to concentrate on a smaller set of signals, simplifying the debugging process. Additionally, we observe that the fault localization cost is low across all three Test Configurations. Using `STL-FL`, we achieve reductions of 88.3%, 68.8%, and 68.8% in the number of blocks to be inspected for Test Configurations 1, 2, and 3, respectively.

In contrast to `CPSDebug`, both `STL-FL` and `E-FL` not only narrow down the scope by reducing the number of model variables and blocks but also accurately identify the faulty components. It is evident that even though `CPSDebug` achieves higher reductions in the number of suspicious signals, particularly for scenarios with more faults (as seen in Test Configuration 3), it fails to correctly pinpoint the fault location.

We summarize the extent of reduction in model variables achieved by both `STL-FL` and `E-FL` using statistical evaluation metrics, namely *Mean* and *Standard Deviation (SD)*, in Table 5.6. These values are derived from all three sets of faulty variants for each subject. It is evident that both `STL-FL` and `E-FL` achieve substantial reductions in the number of suspicious model variables. On average, `STL-FL` and `E-FL` yield reductions of approximately 92% and 90%, respectively, for multi-fault scenarios (ranging from one- to three-fault models).

Fault Localization Cost. To address **RQ-5.1** and **RQ-5.2**, we calculate the *FL_Cost* for all 240 faulty models using `STL-FL`, `E-FL`, and the baseline fault localization method `CPSDebug`. Figure 5.2 illustrates the distributions of the *FL_Cost* values across all three benchmarks for one-fault, two-fault, and three-fault models. Each box plot in Figure 5.2 contains 80 data points (15 for ATCS, 45 for AECS, and 20 for AFCS), representing 80 faulty variants in each category of one-, two-, and three-fault models. The horizontal axis depicts the localization approach, while the vertical axis indicates the *FL_Cost*.

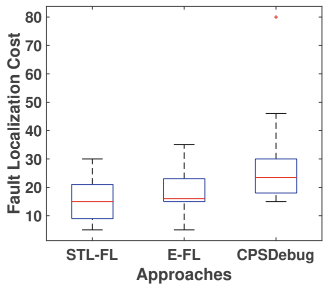
To statistically evaluate the fault localization performance of our approaches compared to `CPSDebug`, we conducted the widely recognized non-parametric Wilcoxon rank-sum test with a significance level of 5% [Wil92, GC14]. The results of the statistical test demonstrate that for multi-fault models (with one to three faults activated), both `STL-FL` and `E-FL` consistently outperform `CPSDebug`, as indicated by the obtained p-values being less than 0.05.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

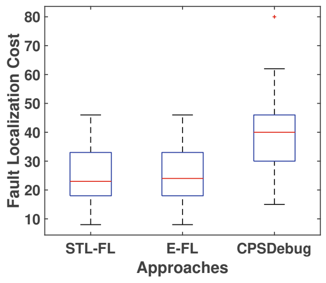
Table 5.6: Empirical evaluation of our approach.

Model		Reduction in model variables			
		STL-FL		E-FL	
		Mean	SD	Mean	SD
ATCS	One-fault	0.9463	0.0263	0.9328	0.0120
	Two-fault	0.8975	0.0158	0.8892	0.0118
	Three-fault	0.8199	0.0143	0.8003	0.0092
AECS	One-fault	0.9742	0.0019	0.9577	0.0132
	Two-fault	0.9526	0.0142	0.9217	0.0113
	Three-fault	0.9021	0.0155	0.8835	0.0129
AFCS	One-fault	0.9704	0.0037	0.9602	0.0167
	Two-fault	0.9364	0.0111	0.9153	0.0149
	Three-fault	0.8926	0.0237	0.8748	0.0265

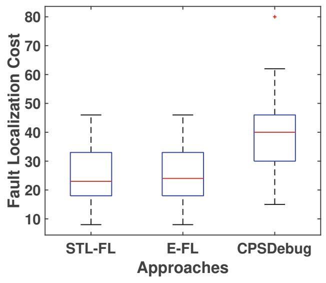
©2022 IEEE.



(a) One-fault models



(b) Two-fault models



(c) Three-fault models

Figure 5.2: Distributions of FL_Cost for one-fault to three-fault models.
©2022 IEEE.

To sum up, our proposed approach demonstrates significant enhancement in fault localization cost compared to CPSDebug. On average, STL-FL reduces the FL_Cost by approximately 43%, whereas E-FL achieves a reduction of nearly 35% in FL_Cost compared to CPSDebug.

Fault Localization accuracy. We assess the overall accuracy of fault localization in the proposed method and compare it with the baseline technique. Table 5.7 provides a summary of fault localization accuracy, expressed as 'total number of fault models for which the faults were correctly identified/the total number of fault models'. The value in parentheses represents the percentage of fault localization accuracy. Our observation indicates that both STL-FL and E-FL exhibit improved fault localization accuracy compared to CPSDebug, especially when confronted with an increasing number of active faults in the models. Notably, while CPSDebug performs adequately with one-

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 5.7: Fault Localization accuracy.

Approach	ATCS	AECS	AFCS
STL-FL	45/45 (100%)	135/135 (100%)	60/60 (100%)
E-FL	45/45 (100%)	135/135 (100%)	60/60 (100%)
CPSDebug	22/45 (48.88%)	53/135 (39.25%)	25/60 (41.66%)

©2022 IEEE.

Table 5.8: Computational cost (in seconds).

	Approach	ATCS	AECS	AFCS
STL-FL	*Testing*	21.4	24.5	23.9
	Localizing	2.3	3.1	2.8
E-FL	*Testing*	20.9	23.8	22.3
	Localizing	2.3	3.1	2.8
CPSDebug		294.1	3394.6	2174.8

©2022 IEEE.

fault models, its accuracy diminishes significantly in the case of multi-fault models. Furthermore, as shown in Table 5.5, the performance of CPSDebug deteriorates as the number of active faults rises.

Computational cost. Table 5.8 outlines the computational time required for our procedure applied to the three considered subjects. The costs for *Testing* and *Localizing* indicate the average time taken by the approach to identify a bug in a model. These values are derived from analyzing various fault variants (ranging from one to three faults) of our subjects for a failing-passing test pair. Notably, *Testing* exhibits greater dominance than *Localizing*. This is primarily attributed to the search task (i.e., optimization) involved in finding the most similar test case. Additionally, it is worth mentioning that the involvement of specification monitoring (through external tool invocation) in STL-FL results in slightly higher computational costs compared to E-FL.

Looking at Table 5.8, both STL-FL and E-FL demonstrate acceptable global overheads of {23.7, 27.6, 26.7}s and {23.2, 26.9, 25.1}s respectively, for bug localization in {ATCS, AECS, AFCS} models. The higher computational cost associated with CPSDebug primarily arises from specification mining utilizing Daikon and TkT. Our approach showcases efficiency, completing the diagnosis in approximately 25 seconds on average, significantly faster than CPSDebug, which requires an order of magnitude more time, as evident from Table 5.8. *To sum up*, the answer to **RQ-5.3** is that our approach is more computationally efficient than the baseline, offering a cost-effective solution for debugging CPS Simulink models.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

5.5 Related Work

In recent years, researchers and practitioners in CPSs have embraced two primary methodologies to assess safety and reliability: *set-based reachability analysis* [DMVP15, FGD+11, CÁS13] and *rigorous testing*, such as differential testing [GHJ07] and falsification analysis [ALFS11, Don10]. Our approaches leverage the characteristics of both these methodologies, with `E-FL` employing aspects of differential testing and `STL-FL` incorporating elements of falsification analysis. From a software engineering standpoint, fault localization has presented ongoing challenges [PCJ+17, WGL+16]. While the presence of faults can be detected by examining falsifying traces or related symptoms, pinpointing the exact fault location within the SUT remains laborious. Recent studies have employed conventional and iterative statistical debugging techniques [LNB+16, LLN+16, LNLB19] to generate ranked lists of blocks requiring further investigation by an engineer to identify fault locations. In contrast, our approach `E-FL` diverges from statistical methods by utilizing (1) differential testing to identify failing test cases and (2) optimization to identify new test candidates for fault localization.

Similar to our `STL-FL` approach, which relies on a STL specification to identify bugs, the study by [DST17] delves into the neighborhoods of falsifying traces within CPSs. Their aim is to pinpoint and analyze input segments responsible for violating the overall specification. This methodology holds promise for systematically refining test candidates to aid in debugging processes. Another approach, focusing on Spectrum-Based Fault Localization (SBFL), integrates trace diagnostics with model slicing techniques [BFMN18]. However, due to the absence of empirical validation, the efficacy of this SBFL method in reducing debugging efforts remains unclear. Moreover, this approach operates under the assumption that faults are injected into specific SUT components, limiting its applicability and making it challenging to gauge its effectiveness comprehensively.

`CPSDebug` [BMM+21] is a recently introduced tool that adopts a gray-box testing methodology to localize bugs and explain failures within CPS designs. The foundational principles of `CPSDebug`, which integrate testing, specification mining, and failure analysis to uncover faults, are extensively elaborated in [BMM+19]. Our empirical assessment indicates that our approach surpassed `CPSDebug` in performance. Additionally, Singh and Saha [SS20] propose a bug localization technique based on matrix decomposition, leveraging falsification of STL properties. Their method provides engineers with a set of suspected signals thought to be responsible for the violation of the STL specification.

In contrast to existing methodologies, we extend our analysis to include a broader spectrum of fault types and mutation operators applicable to Simulink models. Additionally, we refrain from assuming the location of seeded faults or mutations within the SUT, except for ignoring faults within Stateflow charts. Unlike conventional fault localization techniques, our approach integrates two distinct strategies: (1) combining equivalence testing with search-based testing, and (2) merging specification monitoring with search-based testing. This fusion enables us to pinpoint the *fundamental* cause of anomalous occurrences that ultimately culminate in observed failures. As a result, we generate a customized list of

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

suspicious model variables and blocks, ideally suited to aid engineers in their debugging endeavors.

5.6 Conclusion

Venturing into the design of fail-safe CPS entails thorough understanding of its complexity due to heterogeneous units. Besides, it is equally important to know how to deal with observable failures in such systems due to anomalous behavior of any of its components. Tracing back the signals through the model components and atomic units to reveal the faulty blocks is of significant importance in debugging. However, localizing faults in safety-critical CPS can be extremely challenging, due to the complexity in the behavioral dynamics of the many intertwined elements present in CPS models.

We introduced a new method for localizing faults within Simulink models of safety-critical CPS. This approach, guided by an STL property, leverages both failing executions and automatically generated passing executions (resembling the failing ones) to identify the anomalous signals, thereby identifying the probable faulty blocks. Besides, we instantiated our approach using the concept of equivalence testing, wherein fault localization is guided by implicit specifications. We demonstrated the effectiveness of our methodology through three distinct scenarios. Our experimental findings indicate that our approach adeptly localizes multiple faults within acceptable fault localization cost, surpassing the performance of `CPSDebug`, a state-of-the-art fault localization technique.

Data Availability Statement. The experimental data and scripts are publicly accessible at https://gitlab.com/DrishtiYadav/fl.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

CHAPTER 6

Property-Based Mutation Testing

Mutation testing is a well-established technique in software quality assurance, commonly used to evaluate test suites. However, its effectiveness is limited when validating software against specific requirements, especially in scenarios like embedded systems where stringent safety properties are involved. In such cases, a mutant is relevant only if it affects the satisfaction of the tested properties, and it is considered meaningfully-killed w.r.t. a property only if it leads to the violation of that property.

To overcome these limitations, we propose a novel approach called *Property-Based Mutation Testing*. This method assesses the ability of a test suite to exercise the software against specific properties. We conducted experiments with our property-based mutation testing framework on Simulink models of safety-critical CPSs from automotive and avionic domains. Our findings demonstrate how property-based mutation testing is more informative than traditional mutation testing. These results pave the way for advancements in mutation testing and test case generation for CPSs.

6.1 Introduction

Software plays a critical role in ensuring the safety of applications like autonomous vehicles and medical devices. Insufficient software quality assurance can lead to severe

©2023 IEEE. Parts Reproduced, Reused, Reprinted, with permission, from Ezio Bartocci, Leonardo Mariani, Dejan Ničković and Drishti Yadav, "Property-Based Mutation Testing," 2023 IEEE Conference on Software Testing, Verification and Validation (ICST), Dublin, Ireland, 2023, pp. 222-233, https://doi.org/10.1109/ICST57152.2023.00029 system failures, posing significant risks. Therefore, it is essential to conduct thorough testing of software to verify that it complies with critical properties. Software testing is a key aspect of quality assurance and raises important questions: *How can we determine if test cases are capable of detecting faults? How do we assess the effectiveness of our test suites?*

Mutation testing (MT) is a widely recognized technique used to evaluate the adequacy of a test suite with respect to a fault model [DLS78, ABD$^+$79, OPB21, FCA22]. MT involves introducing artificial defects into the software being tested and then assessing the test suite's effectiveness by measuring the percentage of injected faults that the test suite can identify. This injection process utilizes *mutation operators* that systematically modify the software according to predefined patterns, resulting in modified versions known as *mutants*. A test case is said to "*kill*" a mutant if its execution reveals observable differences between the original program and the mutated version. The mutation score, calculated as the ratio of killed mutants to non-equivalent mutants, serves as a measure of the test suite's effectiveness. Ideally, a test suite should achieve a mutation score of one, indicating that it can detect all injected faults.

While MT is effective for evaluating a test suite's performance against a broad range of faults scattered throughout the software, its effectiveness diminishes when the goal is to validate the software against specific requirements. This is particularly evident in the realm of embedded software, where software often undergoes validation against precisely defined safety properties. For instance, in our experimental evaluation, we utilized the *Automatic Transmission Controller System* (ATCS), which is annotated with numerous safety properties expressed using STL. Test cases are crafted to validate the software against these properties.

When applying mutation testing to assess a test suite's ability to thoroughly exercise software concerning a specific property, two key challenges arise: the relevance of the mutants and the relevance of the executions that detect and kill these mutants.

Relevance of Mutants with respect to a Tested Property: Not every mutant is pertinent for evaluating the effectiveness of a test suite against a given property. Specifically, only mutants whose effects propagate in a manner that leads to a violation of the property are deemed relevant. Mutants that do not affect a property should not factor into assessing the adequacy of a test suite concerning that property. Traditional mutation testing fails to differentiate between these types of mutants, thereby overlooking the distinction when calculating the mutation score.

Relevance of the Execution that Kills a Mutant. Simply generating different outputs for the original and mutated programs is insufficient to eliminate a mutant when evaluating a test suite against a property. Effectively exercising the software with respect to a property requires the differences in outputs to be significant and impactful enough to cause a violation of the property in question. Otherwise, the test is generating differences that are marginal w.r.t. the testing objective. For example, in our evaluation of test cases for the ATCS against a property dictating thresholds for engine and vehicle speeds,

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

several tests managed to trigger a mutant in the Transmission component, resulting in output discrepancies. However, these discrepancies did not lead to violations of the specified properties, highlighting a deficiency in the test suite. This scenario is visually depicted in Figure 6.1 (top), where the test induces differences in engine and vehicle speeds without surpassing the threshold. Despite regular MT counting the mutant as killed, the test does not prompt the software to violate the property. In reality, if the fault were present in the original model, the test would not reveal it. This underscores how mutations can be easily killed under regular MT in data-flow models, where most of the components are activated in every computation and values propagate through the blocks in the model. However, these propagated values often result in minor and insignificant output differences. Killing mutants while considering the tested properties presents a substantially more challenging task. For instance, Figure 6.1 (bottom) illustrates a test that successfully reveals the mutant by violating the tested properties, as demonstrated in our experiments.

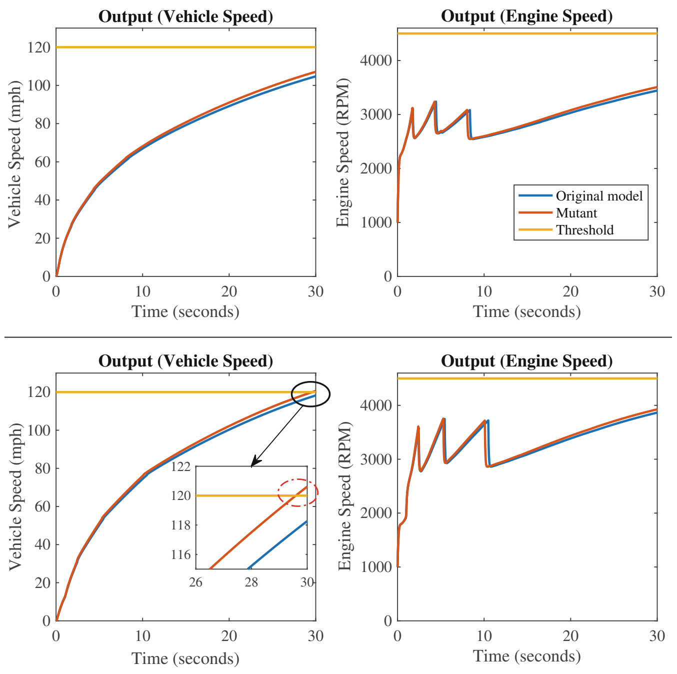


Figure 6.1: Output plots for the original and mutated models of ATCS: (**top**) for a test case satisfying the property on the mutant, (**bottom**) for a test case violating the property on the mutant. The portion of the output trace (vehicle speed) responsible for property violation is highlighted.
©2023 IEEE.

Contributions. In this chapter, we address these challenges by introducing the concept of *Property-Based Mutation Testing (PBMT)* aimed at evaluating test suites against properties or specifications. In doing so, we redefine key aspects of mutation testing to gauge the efficacy of test suites in effectively exercising the software with respect to

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

a property. Additionally, we devise a search-based test generation approach tailored for Simulink models to systematically identify relevant mutants that could be killed through meaningful executions, among a pool of injected mutants. Through empirical analysis, we demonstrate that PBMT offers greater insights compared to traditional MT in assessing the comprehensiveness of test suites. This evaluation is conducted using two benchmarks within the realm of safety-critical CPSs, whose requirements are expressed in STL formalism.

In summary, this chapter presents the following contributions:

1. Introduction of the innovative concept of *Property-Based Mutation Testing* designed for testing software against properties.
2. Definition of a *search-based strategy* aimed at automatically identifying the mutants relevant to PBMT experiments.
3. Presentation of *empirical findings* derived from experiments conducted on Simulink models, illustrating the superior informativeness of PBMT compared to traditional MT when assessing software against properties.

Chapter Organization. Section 6.2 introduces Property-Based Mutation Testing, our proposed methodology. Section 6.3 details the process of testing CPS Simulink models against STL specifications. Section 6.4 outlines our evaluation of two safety-critical industrial benchmarks. Threats to validity are discussed in Section 6.5, while lessons learned are described in Section 6.6. Related work is presented in Section 6.7, and Section 6.8 concludes the chapter.

6.2 Property-Based Mutation Testing

In this section, we introduce *PBMT*, a mutation testing methodology aimed at validating test suites against programs and their properties. This approach assumes a program $\mathcal{P}$ written in a language $\mathcal{L}$ as the SUT, a property ϕ of the SUT, a test suite $\mathcal{T}$, and a set of mutation operators $\mathcal{O}$. PBMT assesses the extent to which the test suite $\mathcal{T}$ validates $\mathcal{P}$ against the property ϕ, focusing on the ability of $\mathcal{T}$ to uncover faults —as specified by $\mathcal{O}$—that could affect ϕ.

Definition 6.2.1 (ϕ-*killed mutant*)**.** "A mutant p is said to be ϕ-*killed* by a test suite $\mathcal{T} \subset \mathcal{T}_U^{\mathcal{P}}$ iff $\exists$ a test case $t \in \mathcal{T}$ such that the following conditions hold:

1. $O(t, \mathcal{P}) \models \phi$, i.e., t satisfies ϕ when executed on the original program $\mathcal{P}$, and
2. $O(t, p) \not\models \phi$, i.e., t violates ϕ when executed on the mutant p. It follows that t exercises the mutation/fault in p in such a way that its effect is propagated to the output up to the violation of the property ϕ." [BMNY23]

©2023 IEEE. Parts Reproduced, with permission, from Ezio Bartocci, Leonardo Mariani, Dejan Ničković and Drishti Yadav, "Property-Based Mutation Testing," 2023 IEEE Conference on Software Testing, Verification and Validation (ICST), Dublin, Ireland, 2023, pp. 222-233, https://doi.org/10.1109/ICST57152.2023.00029

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

The above two conditions together ensure that executing the mutant p with a test case t results in an output that violates the property ϕ (i.e., $O(t, p) \not\models \phi$), while the original program still satisfies it (i.e., $O(t, \mathcal{P}) \models \phi$). This indicates that the test effectively exercises the software, so that if a fault is present, it is propagated to the output, creating significant behavioral differences that ultimately violate the property ϕ.

Analogous to the concept of equivalent mutants in traditional mutation testing, we introduce a refined category called *ϕ-trivially different mutants*. The idea here is that a mutant is not only considered irrelevant if it behaves identically to the original program, but also if the behavioral differences it introduces are insignificant with respect to the property ϕ. In other words, no test case $t \in \mathcal{T}_U^{\mathcal{P}}$ can distinguish between the mutant p and the original program $\mathcal{P}$ in terms of the property ϕ.

Definition 6.2.2 (*ϕ-trivially different mutant*)**.** "A mutant p is *ϕ-trivially different* from $\mathcal{P}$ iff $\nexists t \in \mathcal{T}_U^{\mathcal{P}} : O(t, \mathcal{P}) \models \phi \wedge O(t, p) \not\models \phi$." [BMNY23]

The group of *ϕ-trivially different* mutants encompasses equivalent mutants. However, determining which mutants are *ϕ-trivially different* is an undecidable problem.

Definition 6.2.3 (*ϕ-adequate test suite*)**.** "A test suite $\mathcal{T}$ is *ϕ-adequate* w.r.t. a set of mutation operators $\mathcal{O}$ if it kills all the non ϕ-trivially different mutants that can be generated by $\mathcal{O}$." [BMNY23]

Definition 6.2.4 (*Mutation score* [BMNY23])**.** Given a program $\mathcal{P}$, a set of mutation operators $\mathcal{O}$ and a test suite $\mathcal{T}$, if KD_ϕ represents the ϕ-killed mutants and NTD_ϕ represents the non ϕ-trivially different mutants, then the mutation score assigned to $\mathcal{T}$ is given by:

$$\mathcal{MS}_\phi = \frac{|KD_\phi|}{|NTD_\phi|} \tag{6.1}$$

The goal of developing test suites that meet the requirements of PBMT leads to what is known as the **Mutant Killing Problem**. In essence, this problem entails finding a test case t for a given program $\mathcal{P}$ and a mutant p of $\mathcal{P}$, along with a property ϕ, where t satisfies $O(t, \mathcal{P}) \models \phi$, while $O(t, p) \not\models \phi$.

PBMT poses unique challenges compared to regular MT:

- *Higher risks of introducing ϕ-trivially different mutants*: PBMT may generate more irrelevant mutations compared to MT. In PBMT, in addition to equivalent mutants, there could be mutants that are not equivalent but introduce irrelevant differences concerning a property ϕ.
- *Harder to kill mutants*: In PBMT, mutants must be exercised in a way that not only propagates to the output but also leads to the violation of ϕ. This requirement makes killing mutants in PBMT more challenging compared to MT, as it necessitates more stringent test cases that specifically target the violation of the property ϕ.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

6.3 Mutation Testing of CPS Simulink Models

We apply PBMT to safety-critical CPS Simulink models, which utilize STL to express system safety properties. The fundamentals of Simulink models and STL are covered in Chapter 2. In this section, we introduce a novel method to automatically identify mutants that could be ϕ-killed (i.e., killed with respect to a property ϕ) by test suites.

6.3.1 Search-Based Generation of Mutation Adequate Test Cases

One of the primary challenges in mutation testing, including PBMT, lies in accurately calculating the mutation score, which is complicated by the undecidable nature of identifying equivalent mutants. In PBMT, this challenge is even harder due to the necessity of identifying ϕ-trivially different mutants, which extend beyond equivalent mutants. To tackle this problem, we devised a search-based test generation approach leveraging insights into mutant locations to produce targeted executions, determining whether a mutant can be ϕ-killed. While this method does not provide insights into mutants not killed by the procedure, our experimental findings suggest that assuming it can identify all ϕ-killable mutants offers a reasonable approximation of the mutation score.

It is worth noting that the proposed test generation strategy is not feasible for generating tests in real-world scenarios since it relies on knowledge of fault locations typically unavailable during software testing. Nonetheless, within the PBMT framework, this test generation strategy proves useful for gathering precise empirical data about the number of ϕ-killable mutants.

Specifically, we define the 'Property-based test search problem' as an optimization challenge aimed at discovering a ϕ-adequate test case, expressed as:

Property-based test search problem

Input: a Simulink model $\mathcal{M}$, a FOM $\mathcal{M}'$ (where signal s is altered to signal s' or block b with output s is modified to block b' with output s'), and a property ϕ.
Problem: Find a test case t such that $\rho(O(t,\mathcal{M}),\phi) > 0$, $\rho(O(t,\mathcal{M}'),\phi) < 0$ and $D(s,s')$ is maximum.

Our proposed 'Property-based test search problem' integrates three essential aspects, two stemming from the definition of ϕ-killed mutant and one directing the search towards the mutant and generating an execution that effectively alters the system state:

- $\rho(O(t,\mathcal{M}),\phi) > 0$ necessitates finding a test that succeeds on the original program.
- $\rho(O(t,\mathcal{M}'),\phi) < 0$ necessitates finding a test that violates ϕ in the modified program.
- Maximizing $D(s,s')$ ensures that the mutation significantly affects the internal signal.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

We select the Euclidean distance (also known as the L^2 norm) as the metric to calculate the distance between s and s'. Given that CPS models consist of continuous real-valued variables, the Euclidean distance, a widely used metric for real vector spaces, is an appropriate choice for this computation. More precisely, for two signals of finite length (each with k samples), denoted as $s = (s_1, \cdots, s_k)$ and $s' = (s'_1, \cdots, s'_k)$, the Euclidean distance between s and s' is defined mathematically as:

$$D(s, s') = ||s - s'||_2 = \sqrt{\sum_{i=1}^{k} (s_i - s'_i)^2}$$

The optimization objective is to maximize $D(s, s')$ while ensuring that $\rho(O(t, \mathcal{M}), \phi) > 0$ and $\rho(O(t, \mathcal{M}'), \phi) < 0$ are met. To address this, we leverage BCA [Yad21], a global optimization technique discussed in Chapter 4 and detailed in Algorithm 6.1. We selected BCA due to its superior convergence and speed compared to other available optimizers. Algorithm 6.1 adapts the standard BCA approach with two modifications. Firstly, the initial population (Line 2) comprises randomly generated test cases within their valid numerical input range. Secondly, fitness (Line 3) corresponds to the test objective function value for the test cases, obtained by transforming the constrained optimization problem into an unconstrained one using the scalar penalty constraint handling method [Coe02]. The algorithm iteratively updates the test cases (Lines 6-8) and selects the best solution for the new population based on their fitness values (Lines 9-10). The fittest candidate among all others in the population becomes the new global best solution (Lines 11-14). Termination of Algorithm 6.1 (loop at Line 5) occurs upon finding a test case satisfying the optimization constraints or exhausting the budget (time or maximum iterations).

For every mutant, we address the ‘Property-based test search problem’ to identify a test case that kills it with respect to ϕ. The test suite generated through this process becomes a *fault-directed test suite*, designed to uncover all the *non ϕ-trivially different* mutants.

6.3.2 Test Suite Reduction

To ensure the practicality of the fault-directed test suite, we implement an automatic size reduction mechanism. We define a test case t_r as *ϕ-redundant* with respect to a fault-directed test suite $\mathcal{T}$ if the set of *ϕ-killed* mutants remains the same after adding t_r to $\mathcal{T}$. In simpler terms, if including t_r does not result in the discovery of any new ϕ-killed mutants, then it is considered redundant in the context of $\mathcal{T}$, i.e., $|KD_\phi|_\mathcal{T} = |KD_\phi|_{\mathcal{T} \cup t_r}$.

A *ϕ-non-redundant* test suite is one that excludes ϕ-redundant test cases. Typically, a test suite may include redundant test cases without compromising its *testing power*, meaning these redundant cases can still detect the same mutants with respect to ϕ. Essentially, a single test case may be effective in detecting multiple mutations.

In our experiments, we employ a greedy algorithm similar to the one outlined in [PURMPL12] for reducing the test suite size. Theoretically, covering all p non ϕ-trivially different mutations would demand p test cases in the worst-case scenario. However, in practical scenarios, a smaller number of tests typically prove sufficient.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Algorithm 6.1: Search-based test generation.

```
Input  : $\mathcal{M}$ : A Simulink model.
         $\mathcal{M}'$ : A mutant of $\mathcal{M}$.
         $\phi$ : An STL specification.
Output : $t_{best}$ : A test case that $\phi$-kills $\mathcal{M}'$.
1  Initialize optimizer parameters
2  IP ← GenerateInitialPopulation()
3  FP ← $Fitness(\mathsf{IP}, \mathcal{M}, \mathcal{M}', \phi)$
4  $t_{best}, F_{best} \leftarrow BestFound(\mathsf{FP})$
5  while TimeOut() do
6      for each candidate $k \in \mathsf{IP}$ do
7          $k_{new} \leftarrow Update(k)$
8      end for
9      FP ← $Fitness(\mathsf{IP}, \mathcal{M}, \mathcal{M}', \phi)$
10     $t_{new}, F \leftarrow BestFound(\mathsf{FP})$
11     if $F > F_{best}$ then
12         $F_{best} \leftarrow F$ ;                          // update best fitness
13         $t_{best} \leftarrow t_{new}$ ;                        // update best test
14     end if
15 end while
16 return $t_{best}$
```

6.4 Evaluation

Our evaluation seeks to investigate Property-Based Mutation Testing (PBMT) concerning the testing of CPS Simulink models against STL properties, in comparison to regular Mutation Testing (MT).

6.4.1 Research Questions

Our experiments seek to answer the following research questions:

RQ-6.1: *Does PBMT provide a more accurate assessment of test suite adequacy compared to MT when focusing on a safety property?* To address this question, we evaluate the effectiveness of various test suites using both PBMT and MT. We analyze the resulting scores to determine how well the test cases exercise the software with respect to the target property.

RQ-6.2: *Do all mutation operators have an equal impact on PBMT?* This question explores the influence of various mutation operators on the mutation score, with the goal of identifying operators that produce mutants that are either easily detectable or challenging to detect.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 6.1: Details of Simulink models of our case studies.

Model	Ref.	*#Blocks*	*#Lines*	ϕ (STL specification)	q_T	Sample time	*#Samples*
ATCS	[Don10]	65	92	$\Box((v \leq \bar{v}) \wedge (\omega \leq \bar{\omega}))$	30	0.04	751
AECS	[BMM+21]	825	577	$\Box(\uparrow (cmd \geq m) \rightarrow \Diamond_{[0,T]}\Box_{[0,a]}(\lvert cmd - pos\rvert \leq n))$	10	0.01	1001

©2023 IEEE.

6.4.2 Experimental Setup

We performed our experiments on a MacBook Pro with Apple M1 chip, 16 GB RAM, macOS Monterey with MATLAB™ R2018b. For our evaluation, we developed a prototype implementation of both PBMT and MT with CPS Simulink models in MATLAB. We used the RTAMT library [NY20] for offline evaluation of STL properties.

1. *Our analysis is limited to FOMs.*
2. *Given that our simulations typically have a predetermined duration, we adopt a finite time horizon (referred to as simulation time q_T) for the Simulink models.*
3. *We presume a uniform (i.e., fixed-length) sampling interval in the Simulink models.*
4. *In each case of mutation, we assume that the fault is activated at the beginning of the simulation and persists until its end.*

Below, we delineate our experimental subjects, mutants, and test suites.

6.4.2.1 Experimental subjects

We assess PBMT using Simulink models from two industrial benchmarks within the safety-critical domain. These benchmarks, namely the Automatic Transmission Controller System (*ATCS*) and the Aircraft Elevator Control System (*AECS*), are publicly accessible in the Simulink/Stateflow online documentation of MathWorks [Mat22d, Mat22a]. Both systems are indicative of safety-critical CPS in terms of their size, complexity, and system dynamics. These two models were previously discussed in Chapter 5. A summary of their details is provided in Table 6.1.

6.4.2.2 Fault seeding and mutant generation

For each experimental subject, mutants were created using our FIM tool [BMNY22a] (discussed in Chapter 3), which offers a range of mutation operators tailored for Simulink models. We expanded FIM's capabilities by introducing two supplementary operators for injecting faults into Look-Up Tables (LUTs): (1) Injecting a stuck-at 0 fault into any single entry, and (2) Exchanging entries between two randomly selected neighboring positions.

Table 6.2 presents the count of mutants produced for each subject concerning the particular mutation operator. In Table 6.3, the total number of mutants generated for each subject and their respective generation time is provided. Notably, the process of

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

mutant generation is fast, with an average duration of 1.74 seconds per mutant across both ATCS and AECS.

Table 6.2: Number of mutants of our experimental subjects.

Type	**# Mutants**	
	ATCS	AECS
`Noise`	13	17
`Bias/Offset`	13	17
`Negate`	13	17
`Absolute`	13	17
`ROR`	0	10
`S2P`	1	3
`P2S`	2	6
`ASR`	3	8
`LUT`	2	5

©2023 IEEE.

Table 6.3: Information of generated mutants.

Subject	**Mutants generated**	**Mutant generation time (seconds)**
ATCS	60	68.76
AECS	100	261.64

©2023 IEEE.

6.4.2.3 Test Suite

To evaluate PBMT against MT, we compare test suites generated using two distinct strategies: Adaptive Random Testing (ART) [LNLB19] and Falsification Testing (FT) [ADD+17, AH15]. ART serves as a baseline method that produces test cases uniformly distributed across valid input ranges, ensuring diversity in inputs. Conversely, FT focuses on generating counterexamples, i.e., test cases that violate a given property for a specific model [ZHA19, ZLA+21]. It is important to note that ART and FT operate in fundamentally different ways. ART prioritizes generating numerous test inputs quickly, emphasizing diversity but overlooking the specific property being tested. Conversely, FT concentrates on creating tests specifically designed to violate the property under examination. In FT, for each mutant $\mathcal{M}'$, the objective is to generate a test case t such that $O(t, \mathcal{M}') \not\models \phi$, indicating a violation of the property ϕ. The underlying hypothesis suggests that while ART may yield a higher overall mutation score ($\mathcal{MS}$), it might result in a smaller mutation score specifically related to the property ϕ ($\mathcal{MS}_\phi$) since the

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Wien Bibliothek Your knowledge hub

generated tests are not tailored to ϕ. Conversely, FT may kill fewer mutants overall but is expected to target more mutants relevant to ϕ, potentially resulting in a higher $\mathcal{MS}_{\phi}$.

In our assessment, we created 30 test cases for ATCS and 50 for AECS using ART. Meanwhile, FT generates a test that violates the property for each mutant, if successful. In our falsification testing experiments, we utilize the BCA method and the S-TaLiRo tool, as detailed in Chapter 4 (see Section 4.6.2.2), to generate test suites $\mathcal{T}_{FT-BCA}$ and $\mathcal{T}_{FT-S-TaLiRo}$. For S-TaLiRo, we use the built-in Simulated Annealing (SA) optimizer.

To gather data for addressing our research questions, we executed all test cases in the test suite for each subject and mutant generated. For our experiments, we employed parallel execution of multiple simulations using the Parallel Computing Toolbox™ within the MATLAB/Simulink®environment. Table 6.4 presents the total number of executed test cases (including both suites) and the overall execution time for each subject.

Table 6.4: Scale of Experiments.

Subject	**Total test cases executed**	**Total execution time (seconds)**
ATCS	150	4,768
AECS	250	43,412

©2023 IEEE.

6.4.3 Results

RQ-6.1 explores how effectively PBMT can assess the comprehensiveness of a test suite concerning a safety property that the software-under-test must meet. We compare MT and PBMT on our experimental subjects and calculate the mutation scores $\mathcal{MS}$ and $\mathcal{MS}_{\phi}$. It is worth noting that we utilize identical mutants to calculate both scores. The outcomes are summarized in Table 6.5. The results for regular mutation testing (MT) and Property-Based Mutation Testing (PBMT) are presented in separate rows, with columns representing the ATCS and AECS subject systems. Each subject system displays the scores attained by test suites generated through Adaptive Random Testing ($\mathcal{T}_{ART}$) and Falsification Testing ($\mathcal{T}_{FT-BCA}$ and $\mathcal{T}_{FT-S-TaLiRo}$). Specifically, we provide details on the number of generated mutants, the count of killable and ϕ-killable mutants, the mutants killed by each test suite according to MT and PBMT, and finally the mutation scores $\mathcal{MS}$ and $\mathcal{MS}_{\phi}$.

To identify the killable mutants, we embarked on identifying the equivalent mutants. Our examination of the non-killed mutants aimed to identify mutations that produced variants indistinguishable from the original program. With a high degree of confidence, we successfully identified all equivalent mutants. Indeed, all 13 equivalent mutants within the ATCS model fall under the `Absolute` fault type, introduced in the 'Transmission' component. Each of these mutants attempts to alter certain signals into positive values, a modification that is invalid due to the nature of those signals, which cannot be negative.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Similarly, the AECS model encountered a comparable scenario, with 17 equivalent mutants exhibiting the same behavior. For the identification of ϕ-killable mutants, we leveraged the Search-Based Test Generation (SBTG) technique detailed in Algorithm 6.1. Notably, the computational cost of the SBTG strategy exceeds that of ART and FT due to its optimization constraints. Our procedure successfully pinpointed every ϕ-killable mutant across thirty independent runs of the search algorithm, using a maximum of 1000 iterations as the stopping criterion. All the remaining ϕ-trivially different mutants consist of equivalent mutants that are not susceptible to being killed. This outcome reinforces our confidence in the effectiveness of our method to facilitate fully automated experiments with Simulink models. We make the assumption that the mutants which are not killed by our strategy are ϕ-trivially different mutants, indicating that they do not need to be killed. Consequently, these mutants are omitted from the calculation of $\mathcal{MS}_\phi$.

Table 6.5: Results of Mutation Testing.

Approach		**ATCS**			**AECS**		
		$\mathcal{T}_{ART}$	$\mathcal{T}_{FT-BCA}$	$\mathcal{T}_{FT-S-TaLiRo}$	$\mathcal{T}_{ART}$	$\mathcal{T}_{FT-BCA}$	$\mathcal{T}_{FT-S-TaLiRo}$
MT	# Mutants	60	60	60	100	100	100
	# *Killable* mutants	47	47	47	83	83	83
	# *Killed* mutants	47	46	46	74	70	69
	Mutation Score $\mathcal{MS}$ (in %)	100%	97.87%	97.87%	89.15%	84.33%	83.13%
PBMT	# Mutants	60	60	60	100	100	100
	# ϕ-*killable* mutants	47	47	47	83	83	83
	# ϕ-*killed* mutants	25	27	26	39	35	33
	$\mathcal{MS}_\phi$ (in %)	53.19%	57.44%	55.32%	46.98%	42.16%	39.76%

©2023 IEEE.

When comparing the results between MT and PBMT, it is evident that the mutation score obtained with MT is notably higher compared to PBMT. Specifically, the mutation score ($\mathcal{MS}$) ranges between 83.13% and 100% across all four test suites and both subject systems. In contrast, the $\mathcal{MS}_\phi$ values range between 39.76% and 57.44%. This disparity can be attributed to the inherent characteristics of Simulink models and data-flow computations. In such models, it is generally easy to activate every component, essentially covering all elements in the program. However, ensuring these components effectively contribute to the computation, propagating faults to the output and causing observable issues, is considerably more challenging. In essence, while *reaching* faults may be relatively straightforward, meaningfully *propagating* and *detecting* faults presents a greater difficulty. This trend is consistent across the test suites generated using two different strategies.

These findings underscore the potential pitfalls of relying solely on MT, particularly when critical properties need validation. For instance, consider the scenario depicted in Figure 6.1 (top), where a test case can successfully “kill” a mutant but fails to “ϕ-kill” it. Indeed, the test suites produced through ART and FT attain high mutation scores ($\mathcal{MS}$), potentially leading testers to conclude that the software is thoroughly being tested. However, upon closer inspection, it becomes evident that these test cases may not

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

adequately detect even simple faults, such as those we introduced, which could impact the property under consideration.

It is noteworthy that FT, which aims to falsify the property, does not necessarily kill more mutants compared to ART, which prioritizes diversity over property testing. Moreover, considering that nearly half of the killable mutants were not effectively targeted by the ϕ-killing process, this highlights the necessity for further investigation into thoroughly testing software concerning specific properties, particularly in the context of Simulink programs.

Another notable observation is that all mutants killed by $\mathcal{T}_{FT-S-TaLiRo}$ were also killed by $\mathcal{T}_{BCA}$. However, $\mathcal{T}_{BCA}$, for both AECS and ATCS, achieved higher $\mathcal{MS}_\phi$ values compared to $\mathcal{T}_{FT-S-TaLiRo}$. This indicates that BCA-based falsification testing generates test suites capable of ϕ-killing mutants that were not addressed by the tests produced by S-TaLiRo.

Upon evaluating the tests, we observed a notable level of redundancy, wherein each test had the ability to kill multiple mutants. Surprisingly, all mutants that were killed by ART could be targeted by a single test. This underscores the notion that while some surface faults are readily detectable, others, despite their simplicity, demand more intricate testing methodologies for detection.

Conversely, our investigation revealed that it took four test cases, generated through our SBTG technique, to uncover all 47 ϕ-killable mutations in ATCS. Similarly, a set of 12 test cases was sufficient to detect all 83 ϕ-killable mutations in AECS. This indicates that while compact yet effective test suites can be devised to identify faults using PBMT, a larger number of tests is necessary compared to regular MT to ϕ-kill the mutants.

RQ-6.2 examines the impact of individual mutation operators in PBMT. The objective is to distinguish between operators that produce mutants easily captured by tests (referred to as *simple mutants*), which may not significantly enhance test suite adequacy assessment, and those generating mutants that are difficult to detect (known as *stubborn mutants*), which could contribute more to assessing test suite thoroughness.

The results detailed in Table 6.6 encompass several metrics for each mutation operator: (1) the count of generated mutants, (2) the count (and %) of ϕ-trivially different mutants, (3) the count (and %) of NTD_ϕ (non ϕ-trivially different mutants), (4) the mutation score attained through ART, (5) the mutation score attained through FT, and (6) the number (and %) of NTD_ϕ mutants *not killed* by any test generation method (neither ART nor FT). Note that Table 6.6 presents the combined results across both experimental subjects (ATCS and AECS). Since BCA-based falsification testing produced test suites that successfully ϕ-killed mutants which were not addressed by S-TaLiRo, we present the results for $\mathcal{T}_{FT-BCA}$ in Table 6.6.

Around fifty percent or more of the mutations generated by the `Negate`, `ROR`, `S2P`, and `ASR` operators remained unkilled by either ART or FT. This observation hints that these operators could potentially be more beneficial for PBMT, as they seem to produce faults that are less easy to propagate to the output.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 6.6: Summary of results of PBMT for individual operators.

	`Noise`	`Negate`	`Bias`	`Absolute`	`ROR`	`S2P`	`P2S`	`ASR`	`LUT`
# Mutants generated	30	30	30	30	10	4	8	11	7
# (%) of ϕ-trivially different mutants	0 (0%)	0 (0%)	0 (0%)	30 (100%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
# (%) NTD_ϕ	30 (100%)	30 (100%)	30 (100%)	0 (0%)	10 (100%)	4 (100%)	8 (100%)	11 (100%)	7 (100%)
$\mathcal{MS}_{\phi ART}$ (in %)	66.67%	43.33%	46.66%	0%	0%	25%	62.5%	45.45%	85.71%
$\mathcal{MS}_{\phi FT}$ (in %)	70%	43.33%	50%	0%	0%	25%	62.5%	45.45%	28.57%
# (%) NTD_ϕ *not killed* by ART+FT	9 (30%)	17 (56.66%)	15 (50%)	0 (0%)	10 (100%)	3 (75%)	3 (37.5%)	6 (54.54%)	1 (14.28%)

©2023 IEEE.

For example, all mutations created by the `Negate` operator in the AECS system affected the *Right Outer Hydraulic Actuator* component. Although existing test cases could readily *infect* the execution, such as altering the output of the ‘Line resistance’ block, they were unable to fully *propagate* the impact due to the presence of an intermediate signal (for instance, ‘Piston Force’), which masks alterations if they are not sufficiently large.

None of the mutations introduced by the `ROR` operator were identified by $\mathcal{T}_{ART}$ and $\mathcal{T}_{FT}$. Specifically, we noted that for all available test cases $t \in \mathcal{T}_{ART} \cup \mathcal{T}_{FT}$, the robustness value assessed for the STL property remained unchanged for every `ROR` mutant, mirroring the behavior of the original model. However, certain test cases were able to induce noticeable differences in the outputs and effectively ϕ-kill the mutants, as demonstrated by tests generated using our SBTG technique.

Mutations produced by the `S2P` operator have also proven challenging to ϕ-kill. Furthermore, certain mutations generated by the `ASR` operator could not be identified by test cases within $\mathcal{T}_{ART}$ and $\mathcal{T}_{FT}$. Despite altering internal signals, these mutations do not impact the data-flow computations or signal propagation necessary to affect the property. For instance, consider the `ASR` mutation in the ‘Hydraulic Actuator’ component of the *Right Inner Hydraulic Actuator* unit of AECS, where the substitution of $-+$ with $+-$ induces notable changes in the local signal but lacks the strength to ϕ-kill the mutant.

Conversely, two operators did not yield significant benefits. The `Absolute` operator solely produced equivalent mutants, implying caution in its application, especially in systems designed to handle negative values. It is crucial to exercise control over where faults are injected when employing this operator, although such scenarios are rare in CPS. Overall, no valuable mutations were observed in our study subjects. Regarding the `LUT` operator, all mutations were easily ϕ-killable except for one, which generated values challenging to propagate to the output, yet feasible to propagate, as evidenced by the test suites generated using our SBTG method. Despite being the sole operator targeting look-up tables, testers may opt to omit it under stringent time constraints during testing.

6.5 Threats to Validity

We now delve into the validity threats, focusing on various perspectives and potential risks:

External validity. One significant threat to external validity revolves around the applicability of our findings to other software systems. Our experiments were conducted within the domain of data-flow oriented computations, specifically focusing on Simulink models. It is important to acknowledge that our observations might not be universally applicable across different contexts, such as object-oriented programs. However, within the realm of safety-critical CPS Simulink programs, where testing against safety properties holds paramount importance, our results are notably clear and relevant. Furthermore,

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

the manageable scale of our experiment facilitated manual mutation analysis, enabling us to identify equivalent mutants effectively.

Another validity concern pertains to the representativeness of the injected faults. Our study's outcomes are based on common mutation operators tailored for Simulink models, utilizing our FIM tool [BMNY22a] (discussed in Chapter 3) and its predefined mutation operators. Additionally, we augmented these operators with extra mutation operator to address LUTs. However, the absence of evidence regarding *real* faults limits the breadth of our conclusions. We posit that our approach could potentially be expanded to encompass models that possess documented (real) faults, thus alleviating this concern. This avenue represents a potential direction for future research.

Internal validity. In our experiments, we solely examined FOMs, denoting faulty Simulink models with only one fault/mutation. Models may harbor multiple faults/mutations that could interact with one another. Consequently, the outcomes might diverge when investigated with multi-fault Simulink models. However, given that the majority of existing research on MT centers on FOMs of software artifacts [ALN13, TCPB+20], we evaluated our technique using single-fault models, deferring the exploration of HOMs for future investigations.

Conclusion validity. The primary threat to conclusion validity is random variations. To mitigate this threat, we conducted thirty independent runs of the test generation algorithms.

6.6 Insights and Reflections

We now walk through the lessons learned and insights gained from our experiments.

Insight 1 - Generating PBMT-adequate test suites presents challenges. Our investigation reveals that neither of the two leading test generation strategies for Simulink programs (of our two experimental subjects) achieved a high mutation score with PBMT. Unlike regular MT, PBMT is more intricate: a test case capable of *killing* a mutant may not effectively ϕ-kill the same mutant. Given the significant reliance of the embedded software industry on properties for verification and validation, there is a pressing need to develop testing tools that comprehensively assess software functionality. The conceptualization of PBMT represents a notable advancement in the field, potentially shaping the development of more refined and effective test generation strategies.

Insight 2 - MT inadequately gauges the comprehensiveness of a test suite. While MT remains applicable to Simulink programs, its effectiveness is limited. Test generation techniques may successfully kill mutants if properties are not taken into account. This highlights the necessity of designing test cases that not only cover mutants but also propagate the errors generated by mutants, thereby enhancing their detectability in the output and the system properties. These crucial aspects of testing are not adequately addressed within the scope and framework of traditional MT.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Insight 3 - Test case generation guided by PBMT can yield effective test cases. We introduced a Search-Based Test Generation (SBTG) technique aimed at identifying test cases capable of ϕ-killing mutants. This approach has proven highly effective in ϕ-killing mutants, suggesting its potential as a foundation for developing a mutation-based test case generation strategy.

Insight 4 - Not all mutations equally contribute to testing CPS Simulink models. Our findings suggest that certain mutation operators tend to produce mutants that are less informative for testing purposes, i.e., they generate ϕ-trivially different mutants. For example, the `Absolute` operator consistently generated equivalent mutants, indicating its limited utility. Conversely, operators such as `Negate`, `ROR`, and `ASR` produced mutants that were challenging to ϕ-kill, emphasizing the need for test case generation strategies that thoroughly exercise the software in non-trivial manners.

6.7 Related Work

Mutation Testing. From a software engineering standpoint, MT stands out as a powerful software testing method for evaluating the quality of test suites [DLS78, ABD+79]. The literature on MT and analysis encompasses a broad array of theoretical inquiries and empirical explorations across various software artifacts [JH11, SdSdS17].

The work in [PM11] combines symbolic execution, concolic execution, and evolutionary testing to automate the test generation for weak MT of programs. Along a similar line of research, the work in [PM12] introduces a path selection strategy aimed at selecting test cases capable of killing the mutants. Additionally, efforts to minimize test suites have been explored through techniques like Integer Linear Programming (ILP) [PLEBMBN18], Greedy algorithms [JSKW16, PURMPL12], formal concept analysis [LG15], etc.

Several notable studies have investigated the applicability of MT to safety-critical industrial systems, as evidenced by empirical inquiries such as those reported in [BH12, RWK17, DPHG+18, OPB21]. While the work in [OPB21] proposes a comprehensive mutation analysis pipeline for assessing the quality of test suites for embedded software, it overlooks the significance of software properties and how they should be handled during MT. In contrast to conventional MT research, our approach incorporates properties (allowing to express software requirements and specifications), enabling the formalization of the notion of killing the mutants.

Mutations with Simulink models. Mutation in Simulink models primarily involves introducing faults through alterations using mutation operators [B+12]. Various tools and techniques have been proposed by researchers for generating mutants in Simulink models, including SIMULTATE [PRWN16], MODIFI [SVET10], ErrorSim [SMSJ17] and FIBlock [FMMJ21]. Additionally, SLforge [CMM+18] is noteworthy for its capability to automatically generate random valid Simulink models for conducting differential testing. Our work also introduces FIM (as discussed in Chapter 3), a toolkit designed for systematically injecting faults into Simulink models in an automated manner. For

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

our experiments, we chose FIM due to its higher level of automation compared to other available tools.

Mutation-based test case generation. Mutation-based test case generation approaches in regular MT utilize mutants to generate test cases capable of identifying errors and detecting mutants. Certain approaches focus on generating tests that uncover mutations introduced in the specification, such as in UML models [KST$^+$15, AAJ$^+$14, ABJ$^+$15, ABJK11, KS16, FKS$^+$19]. Property-Based Mutation Testing (PBMT), however, diverges in several aspects: it does not aim at mutations within the specification, and it introduces a unique concept in MT.

Existing methods for addressing Simulink models primarily concentrate on targeted test-data generation, utilizing approaches such as search-based testing [ZC05, ZC08] or behavioral analysis methods (like bounded reachability) [BHM$^+$09, HRK11]. Essentially, these techniques aim to create a test suite suitable for MT, achieving full mutation coverage based on the RIP model. Drawing inspiration from these methods, we developed our search strategy to automatically ϕ-kill mutants. Moreover, PBMT introduces a unique form of MT that evaluates the adequacy of test suites concerning properties, a concept not previously explored in mutation-based testing.

6.8 Conclusion

We introduced Property-based Mutation Testing (PBMT), an innovative approach aimed at assessing the effectiveness of test suites concerning software properties. Our formulation of mutant killability addresses the satisfaction (and violation) of a property in the original program (and its mutated version). We establish rigorous semantics for PBMT and its associated problem of mutant killing, enabling the generation of test cases through a global optimizer-based search. Employing various test generation strategies, we examined their impact on mutant killability.

Our investigation of PBMT focused on two Simulink models within the safety-critical CPS domain, demonstrating the greater challenge and significance of testing software against properties compared to conventional MT, where mutants are easily killed. Ultimately, our assessment highlights the limitations of state-of-the-practice Adaptive Random Testing and Falsification Testing techniques in generating test suites capable of effectively killing mutants when evaluated against properties.

Data Availability Statement. The tools and experimental data are publicly accessible at https://gitlab.com/DrishtiYadav/mt to facilitate reproduction and support future research.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

CHAPTER 7

Signal Feature Coverage and Testing

Designing CPSs typically involves dataflow modeling, which has a structure different from traditional software, rendering standard coverage metrics inadequate for thorough testing. To overcome this limitation, we introduce *signal feature coverage*, a new metric for systematically testing CPS dataflow models. Signal feature coverage is derived by leveraging various signal features. We developed a testing framework within Simulink® to generate test cases based on this coverage metric. Our approach was evaluated through experiments on five Simulink® models tested against various STL specifications. The results show that our coverage-based testing approach significantly outperforms state-of-the-art testing methods in terms of fault detection capability.

7.1 Introduction

As discussed in Chapter 1, detecting faults promptly is vital in the CPS development process because fixing issues after hardware integration is extremely expensive. With the rising complexity of safety-critical CPSs [Res23], the importance of thoroughly testing CPS dataflow models becomes more evident, highlighting the need for robust test generation strategies that can swiftly identify and resolve potential issues. In embedded software testing, the effectiveness of test suites is frequently assessed by examining their

The content of this chapter is from the following unpublished work which is currently *under submission*: Ezio Bartocci, Leonardo Mariani, Dejan Ničković and Drishti Yadav, "Signal Feature Coverage and Testing for CPS Dataflow Models".

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

code coverage and their fault-detection capabilities. Structural coverage metrics, such as statement and branch coverage, are widely employed in both research [FA11, YLW06] and industry [IPJF19]. These metrics assess how thoroughly a test suite exercises a system's components, identifying potential gaps in the testing process. Despite that, a large body of related research works have shown that depending exclusively on structural coverage criteria is frequently inadequate for identifying faults in software programs and models [GRS+16, IH14, NA09, SP10]. This is especially the case for standard metrics used in CPS dataflow models. Because of the interconnectedness of the elements in these models, running a test case to cover one element often results in the activation of multiple other elements. For example, our experiments showed that an average of 7.6 random test cases could cover over 90% of the signals in the CPS models we tested. In one instance, a single test was enough to cover all the signals in the model. Consequently, assessing structural coverage offers minimal understanding of a test suite's effectiveness, which might result in undetected flaws in the system. For comprehensive testing of CPS dataflow models, it is vital to employ more advanced coverage metrics that accurately reflect the complex behaviors and interactions within these models.

This chapter of the thesis introduces a novel approach to tackle this challenge, presenting the concept of *signal feature coverage*. This coverage criteria involves the comprehensive exploration of the *internal signals* within a CPS dataflow model, considering their diverse time and frequency *features*. Unlike traditional coverage metrics constrained within the 0-100% range, signal feature coverage intentionally operates in an unbounded domain from 0 to infinity, acknowledging that there is always room for more thorough software testing. Much like traditional structural metrics, enhancing coverage can incur significant costs, necessitating a careful balance between costs and benefits.

It is crucial to emphasize that our contribution goes beyond just introducing a new coverage criterion. Building upon this concept, we developed *Feature-Coverage Testing* (FCT), a search-based test generation strategy aimed at generating test suites that maximize signal feature coverage. Additionally, we apply our FCT method to various combinations of target signals and their specific features. The initial variant, termed *Comprehensive Feature-Coverage Testing*, focuses on exhaustively testing the system by maximizing the signal feature coverage of all features across all signals within a dataflow model. Moreover, we explore a testing scenario that concerns CPS dataflow models assessed against formal STL properties. More specifically, our second variant of FCT, denoted as *Specification-directed Feature-Coverage Testing*, strives to maximize the coverage exclusively for those features and signals that substantially influence both the system outputs and the fulfillment of STL properties.

Contributions. In summary, this chapter delivers the following contributions:

1. Introduction of *Signal Feature Coverage*, a coverage criterion tailored to enhance the assessment of test suites for CPS dataflow models.
2. Introduction of *Feature-Coverage Testing*, a novel approach for generating test cases that can be implemented in two variations for comprehensive testing of CPS

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

dataflow models.

3. Experimental evaluation of signal feature coverage and feature coverage testing using five Simulink® models and ten STL properties. Results indicate that test case generation based on signal feature coverage can yield an approximately 60% increase in mutation score compared to traditional testing approaches.

Chapter Organization. In Section 7.2, we introduce the signal feature coverage criterion and the corresponding coverage-based test generation algorithm. Section 7.3 details our evaluation using five industrial CPS dataflow Simulink models. Section 7.4 reviews related work, and Section 7.5 offers the concluding remarks.

7.2 Signal Feature Coverage

We begin by introducing the concept of a *signal feature* and outlining a collection of such pertinent features (refer to Section 7.2.1). Subsequently, we propose *signal feature coverage* (outlined in Section 7.2.2) as a metric for assessing test adequacy in CPS dataflow models. Following this, we present an algorithm designed to generate test cases that maximize the coverage of signal features (detailed in Section 7.2.3). Additionally, we implement the feature-coverage based test generation method using a criterion that prioritizes signals and features likely to have a significant influence on the system (detailed in Section 7.2.3.2).

7.2.1 Signal Features

In this section, we explore signal features, which encompass statistical, spectral, and other quantifiable properties of signals. We delineate a standardized set of signal features commonly utilized in both CPS applications and signal processing. Our proposal includes ten signal features that offer valuable insights into system behavior. The choice of these features is justified by their recognized importance in signal processing, system analysis, and fault detection, as evidenced by prior research [SG23, RMP20, KISH20, BG20, YET20, KAC21]. These features have found widespread application across diverse fields such as communications, biomedical applications, and control systems.

Integrating these features into our testing methodology for Simulink® models of CPSs, our objective is to comprehensively assess essential aspects of signal behavior, facilitating thorough system testing, fault diagnosis, and performance evaluation. We categorize the chosen signal features into four distinct feature sets: Basic statistics, Higher-order statistics, Impulsive metrics, and Frequency features. In what follows, we reserve the use of the bold letter $\mathbf{u}$ for a signal of finite length which contains k samples such that $\mathbf{u} = (u_1, u_2, ..., u_k)$.

Basic statistics. Within this feature set are four distinct features. Basic statistics are employed to measure the spread of a signal [SG23]. Table 7.1 outlines the features encompassed in this set, comprising the mean, standard deviation, root-mean-square, and shape factor.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

Table 7.1: The Basic Statistics feature set [SG23].

Feature	**Mathematical formula**
Mean	$\mu = \frac{1}{k}\sum_{i=1}^{k} u_i$
Standard Deviation (SD)	$\sigma = \sqrt{\frac{1}{k-1}\sum_{i=1}^{k} \lvert u_i - \mu\rvert^2}$
Root-Mean-Square (RMS)	$\mathbf{u}_{RMS} = \sqrt{\frac{1}{k}\sum_{i=1}^{k} \lvert u_i\rvert^2}$
Shape factor (SF)	$\mathbf{u}_{SF} = \frac{\mathbf{u}_{RMS}}{\frac{1}{k}\sum_{i=1}^{k} \lvert u_i\rvert}$

Table 7.2: The Higher-Order statistics and Impulsive metrics of a signal.

Feature	**Description**	**Mathematical formula**
Kurtosis	It quantifies the distribution shape (length of the tails) of a signal [RMP20].	$\mathbf{u}_{kurt} = \frac{\frac{1}{k}\sum_{i=1}^{k}(u_i-\mu)^4}{\left[\frac{1}{k}\sum_{i=1}^{k}(u_i-\mu)^2\right]^2}$
Skewness	It measures the asymmetrical spread of a signal about its mean value [RMP20, Mat23a].	$\mathbf{u}_{skew} = \frac{\frac{1}{k}\sum_{i=1}^{k}(u_i-\mu)^3}{\left[\frac{1}{k}\sum_{i=1}^{k}(u_i-\mu)^2\right]^{\frac{3}{2}}}$
Peak Value	It is the peak or maximum absolute value of the signal [SG23].	$\mathbf{u}_p = \max_i \lvert u_i\rvert$
Impulse factor	"It presents the impulsive behavior and the multitude of peaks" [KBG+21] in the signal.	$\mathbf{u}_{IF} = \frac{\mathbf{u}_p}{\frac{1}{k}\sum_{i=1}^{k}\lvert u_i\rvert}$
Crest factor	It indicates how extreme the peaks are in the signal [Mat23a].	$\mathbf{u}_{crest} = \frac{\mathbf{u}_p}{\mathbf{u}_{RMS}} = \frac{\mathbf{u}_p}{\sqrt{\frac{1}{k}\sum_{i=1}^{k}\lvert u_i\rvert^2}}$

Higher-order statistics. Within this feature set, there are two features. Higher-order statistics assess the symmetry and flatness of a signal [SG23], offering insights into system behavior via the third moment (skewness) and fourth moment (kurtosis) of the signal [Mat23a], as summarized in Table 7.2.

Impulsive metrics. Within this feature set, there are three features. Impulsive metrics are utilized to assess peak changes in the signal, encompassing peak value, impulse factor, and crest factor [Mat23a], as discussed in Table 7.2.

Frequency features. The notion of frequency is straightforward for sinusoidal signals, but real-world signals often exhibit greater complexity. Analyzing non-stationary signals presents challenges in frequency analysis, prompting the emergence of *instantaneous frequency* [Boa92a, Boa92b]. Instantaneous frequency captures the time-varying characteristics of signals and holds significance across various applications such as seismic analysis, radar systems, communications, and biomedical research. In our investigation,

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

we incorporate the '*Peak Instantaneous frequency*' feature into the frequency feature set to enhance the analysis of associated signals. As a result, this feature set encompasses a single feature.

In mathematical terms, the instantaneous frequency $f_{inst}(t)$ is computed as the derivative of the phase of the analytic signal derived from the input signal, represented as follows:

$$f_{inst}(t) = \frac{1}{2\pi}\frac{d\phi}{dt}$$

In the above equation, ϕ represents the phase of the analytic signal $\mathbf{u}_A$ derived from the input signal $\mathbf{u}$. The process involves computing the analytic signal $\mathbf{u}_A$ by performing the *hilbert transform* on $\mathbf{u}$. Subsequently, the *Peak Instantaneous frequency*, represented as $\mathbf{u}_{PIF}$, is determined as the maximum value of $f_{inst}(t)$.

7.2.2 Signal Feature Coverage

Let us consider an internal signal s from a CPS dataflow model[1] $\mathcal{M}$ and a feature $\mathbf{x}$ associated with that signal, denoted as $\mathbf{x}^s$. We assume that this feature $\mathbf{x}^s$ needs to be exercised within a certain range R, bounded by its minimum (min_R) and maximum (max_R) values. Let $X^s = \{x_1^s, x_2^s, \ldots, x_n^s\}$ represent a set of n observations of the feature $\mathbf{x}^s$.

To assess the extent to which X^s exercises the feature $\mathbf{x}$ of s, we evaluate how *uniformly* and *densely* the values in X^s sample the range of values for the feature. This method provides a reasonable confidence regarding the test suite's ability to encompass a wide range of behaviors for the selected feature and signal.

To introduce this metric, we utilize a simple automatic binning algorithm to divide the range R into N bins of equal width, denoted as $R_1, \ldots, R_N$. For example, if $R = [0, 10]$ and $N = 4$, the resulting bins would be $[0, 2.5), [2.5, 5), [5, 7.5), [7.5, 10]$[2]. These N bins are referred to as the *Standard bins*.

Given a set of observations X^s and a range R divided into N bins $R_1, \ldots, R_N$, we establish the concepts of bin occupancy and bin coverage as follows.

Definition 7.2.1 (**Bin Occupancy**)**.** A bin R_i is considered occupied if there exists an observation $x_i^s \in X^s$ such that x_i^s falls within R_i. We define the bin occupancy function $Occ(R_i)$ as:

$$Occ(R_i) = \begin{cases} 1, & \text{if } R_i \text{ is occupied} \\ 0, & \text{otherwise} \end{cases}$$

Definition 7.2.2 (**Bin Coverage**)**.** The *Bin Coverage*, denoted by β, is then defined as

$$\beta = \frac{\#\,Standard\,Bins\,Covered}{\#\,Standard\,Bins} = \frac{\sum_{i=1}^{N} Occ(R_i)}{N}$$

[1]We assume that the models are determinsitic.

[2]Typically, we consider the last bin to be inclusive on both ends.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

Figure 7.1 illustrates two scenarios of bin coverage for the same number of samples. In the bottom scenario, eight samples achieve 100% bin coverage with a division into 6 bins. Conversely, in the top scenario, the same eight samples only achieve $\frac{7}{12} = 58.33\%$ bin coverage with a division into 12 bins.

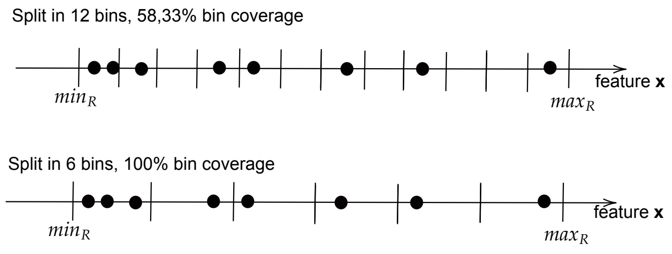

Split in 12 bins, 58,33% bin coverage
min_R
max_R
feature **x**
Split in 6 bins, 100% bin coverage
min_R
max_R
feature **x**

Figure 7.1: Sample bin coverage of a feature.

To assess how *densely* and *uniformly* a feature is sampled, we determine the *finest division* in bins of its domain that is covered by the available samples. Essentially, if a feature is thoroughly exercised across its entire range, it should allow for a finer-grained division that is well-covered by the available samples. We quantify this coverage by counting the number of bins in the finest division. For example, if a set of ten samples covers up to five standard bins (i.e., bins of equal length), the coverage achieved would be five.

Achieving higher coverage levels is increasingly harder with additional samples. For example, covering four bins requires at least four evenly distributed values, whereas covering a division in five or six bins, demands better distribution of samples.

This measure of coverage satisfies some unusual but useful properties. Notably, the coverage is measured as a natural number, which (i) discretizes the domain, capturing the incremental coverage levels that can be achieved, and (ii) is unbounded, reflecting the idea that testers can continuously add further evidence to better exercise a system. Indeed, an empirical research question pertains to identifying meaningful coverage levels that offer a beneficial balance between cost and results. This question is not exclusive to this coverage metric, but is also relevant to bounded coverage metrics, which are seldom addressed until reaching 100% coverage.

In a more formal manner, considering a set of samples X^s, the signal feature coverage of a feature $\mathbf{x}$ of a signal s can be defined as follows.

Definition 7.2.3 (Individual Signal Feature Coverage).

$$\zeta_{\mathbf{x}^s} = \arg\max_N (\beta \geq 0.95) \tag{7.1}$$

Note that here, to be practical and accommodate for intervals that might be particularly hard or impossible to cover, we require the samples to cover a significant portion of the standard bins, defined as 95% of the bins in the partition.

We see the signal feature coverage as primarily a multi-optimization problem, that is, test suites should target the individual features and signals, and coverage reports should

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

detail the results per signal. However, it is also possible to derive an aggregated coverage measure for an entire system, such as considering the average or minimum feature coverage value obtained for a system. In our analysis, we compute the coverage measure for a system by determining the *average* of *individual signal feature coverage* values across the target signals and target features within the system.

Definition 7.2.4 (**Signal Feature Coverage**)**.** In conventional dataflow system models, there is usually a wide range of signals, each possibly containing numerous features that testers may want to explore beyond those initially discussed in Section 7.2.1. Let S represent the set of all signals in the dataflow model of a system, and F denote the set of all features. If $S' \subseteq S$ and $F' \subseteq F$ are the subsets of signals and features that a tester wants to examine, the *Signal Feature Coverage* is calculated as follows:

$$\zeta_{S'F'} = \frac{1}{|S'||F'|} \sum_{s \in S'} \sum_{\mathbf{x} \in F'} \zeta_{\mathbf{x}^s} \tag{7.2}$$

To customize the coverage evaluation based on specific testing needs, we offer two versions of the meta-definition of $\zeta_{S'F'}$ with varying subsets of signals and features. For a thorough examination and to gain a comprehensive insight into the system, a tester might aim to explore the coverage of all features across all signals within a system model. This version of signal feature coverage is referred to as the "*Comprehensive Signal Feature Coverage*" (ζ_{comp}), calculated with $S' = S$ (all signals) and $F' = F$ (all features).

Alternatively, testers might choose to narrow their focus and calculate signal feature coverage for a specific subset of signals and their corresponding subset of features, which can be beneficial for system testing. These subsets are typically determined by testers and may require domain knowledge or thorough testing. One relevant scenario involves selecting only the features that could significantly impact a property ϕ under examination. This variant of our coverage criterion is termed "*ϕ-driven Signal Feature Coverage*" (ζ_ϕ), concentrating on the analysis of *important* features F_ϕ of *important* signals S_ϕ that notably affect the satisfaction of ϕ. ζ_ϕ is essentially $\zeta_{S'F'}$ with $S' = S_\phi$ and $F' = F_\phi$.

7.2.3 Testing for Signal Feature Coverage

In this section, we introduce *Feature-Coverage Testing (FCT)*, a test generation strategy designed to maximize the *signal feature coverage* for S' signals and their associated F' features. The test generation process is formulated as a single-objective optimization problem aimed at maximizing the *signal feature coverage* of the target features of the target signals of a system.

Signal feature coverage-based test generation problem

INPUT: a dataflow model $\mathcal{M}$ with S' signals and F' features of each signal.
PROBLEM: Find $\mathcal{TS}$ s.t. $\zeta_{S'F'}$ is maximum.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

The aim of this maximization problem is to generate a test suite $\mathcal{TS}$ comprising test cases $\mathbf{t} \in \mathcal{TS}$ that thoroughly cover the target features of the target signals within the system. We implement the test generation process using Simulink® as our CPS dataflow modeling and simulation platform, along with BCA (discussed in Chapter 4)[3], to maximize signal feature coverage by iteratively refining a test suite based on the coverage metric.

In our BCA-based test generation method, each test $\mathbf{t}$ is similar to a thrombocyte within the swarm, while the swarm itself corresponds to a test suite. In our approach, we formulated fitness (expressed as the objective function $\zeta_{S'F'}$) at the swarm level rather than at the individual particle level. This choice is consistent with our aim of maximizing $\zeta_{S'F'}$, which is derived from the entire test suite rather than from individual test cases.

Similar to classical BCA, we initiate with an initial population, consisting of thrombocyte positions. Here, thrombocyte *positions* signify individual test cases. It is important to note that we represent each test case as a numerical vector $[pos_0, \ldots, pos_N]$, where each value pos_i denotes a characteristic of the system's input signal. For example, in the AECS model, the system's input includes the *pilot command*, characterized by its amplitude Amp and frequency $Freq$. Thus, the test case for the AECS model is expressed as $t_{AECS} = [Amp, Freq]$, where both Amp and $Freq$ take numeric values within their valid ranges.

We then compute the fitness $\zeta_{S'F'}$ as outlined in Section 7.2.2. Upon performing the coverage computation, we obtain a set of bins that partition the feature values of the signal s associated with the test cases in the population. Our goal in maximizing feature coverage is to ensure there is at least one element in each bin while also fostering diversity in the feature values. To achieve this, we update the test cases using the BCA update equations. The BCA update equations guide the tests to explore and converge towards promising regions, optimizing the overall search process. When confronted with bins containing multiple feature values, we pick a test case at random from that bin, designate it as the global best, and update the other test cases within that bin using the regular BCA update equations. Following the update, the algorithm recomputes the fitness. If the objective function value $\zeta_{S'F'}$ shows improvement, the updated test suite is accepted as an optimal solution. The process continues until the budget is exhausted.

Several testing needs can impact the generation of tests for feature-coverage testing, which are outlined as follows.

7.2.3.1 Comprehensive Feature-Coverage Testing

When testing covers all signals S and features F, the method is referred to as *Comprehensive Feature-Coverage Testing* (FCT-C). The aim of FCT-C is to maximize the coverage of all signal features, denoted as Comprehensive Signal Feature Coverage ζ_{comp}.

[3]We also compared the performance of BCA with two other metaheuristic algorithms including classical Particle Swarm Optimization (PSO) and Water Cycle Algorithm (WCA), observing similar results up to two decimal places.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

This approach is achieved by applying the BCA framework discussed earlier to optimize ζ_{comp}.

7.2.3.2 Specification-directed Feature-Coverage Testing

In real-world scenarios, testing activities often prioritize ensuring the system's adherence to functional, non-functional, and safety requirements. Especially when conforming to safety specifications is critical, testing aligns with formal specifications expressed in STL. In these cases, we refer to the feature-coverage testing strategy as *Specification-directed Feature-Coverage Testing* (FCT-S). Within FCT-S, the focus is on a meaningful subset of features within the internal signals that have a significant impact on meeting the specified safety requirements. In our implementation of FCT-S, we limit signal feature coverage to the *features of the internal signals that significantly impact* the robustness (and consequently the satisfaction/violation) of the STL specifications[4]. Identifying features with the greatest impact on robustness helps streamline testing by focusing on the most influential aspects of the system, thereby improving the effectiveness of the fault detection process.

There is no unique way of measuring the impact of a feature on the robustness, and various approaches have been proposed in the literature. One common approach is *sensitivity analysis*, which introduces small perturbations to a feature's value to observe changes in system robustness, directly measuring the feature's influence on the satisfaction or violation of the formal specification. Another method is *feature importance ranking*, used in machine learning models like decision trees or random forests, where a feature's significance is based on its contribution to decision-making or error reduction. These techniques can be adapted to formal specification-based testing, prioritizing features that significantly impact robustness outcomes. Other methods include *causal inference*, assessing the causal effect of feature changes on robustness, and *mutual information-based techniques*, which capture non-linear dependencies by measuring the shared information between a feature and robustness score. However, these methods may introduce added computational complexity or require extensive data, which can be challenging in practical scenarios.

In our study, we propose estimating the impact of features on robustness by examining the correlation between a feature's values and the robustness of the simulation trace concerning the STL property. The correlation can indicate whether a feature tends to increase or decrease robustness as its values change, offering insights into how the feature influences system behavior in relation to the formal specifications. We chose this method because it provides a straightforward, data-driven way to identify significant features. More specifically, we employ a statistical procedure based on bivariate analysis to empirically uncover the correlations between each feature and the robustness metric, which we found to be effective based on our experimental findings. However, it is important to

[4]It is worth mentioning that in specification-less scenarios, one could still evaluate the impact of a feature directly on the model outputs. However, this approach holds limited relevance for CPS dataflow models, which are typically tested against predefined requirements.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

note that correlation measures the strength and direction of a linear relationship between features and robustness. If the relationship is non-linear, correlation may not adequately capture the true impact of the feature, potentially leading to an incomplete assessment.

Algorithm 7.1 outlines our significant feature identification procedure, which iteratively reduces (i) the signal population by considering its correlation with the output, and (ii) the number of features by considering its correlation with the robustness with respect to the STL property. It operates in three main stages. First (lines 6-9), the procedure generates and executes tests to gather evidence of correlation among signals and features. Second (lines 10-15), it progressively incorporates randomly generated test cases to the existing test suite and checks correlation to iteratively reduce the set of the significant signals and significant features. The gradual reduction in the number of signals is achieved by discarding 20% (as per the discard threshold) of the least correlated signals in each iteration. Third (line 16), it removes the redundant internal signals (i.e., those significant signal variables whose values are implied by other significant signal variables) and their associated features to derive the final reduced set of significant signals and features, denoted as $\hat{\mathcal{V}_H}$ and $\hat{\mathbf{X}}$, respectively.

In more detail, the algorithm starts with a model $\mathcal{M}$, a property ϕ, and a set of signal features `Feat` to be investigated. In our experiments, we examine the ten features outlined in Section 7.2.1. The algorithm initiates by randomly creating an initial test suite (line 6). Subsequently, it employs the function Monitor to simulate the model against each test in the test suite and calculate the robustness ρ (as explained in Chapter 2) of each resulting simulation trace with respect to the STL property ϕ.

To identify the significant signals, the algorithm utilizes the FindTopS subroutine (Algorithm 7.2) which calculates correlation coefficients between each internal signal in $\mathcal{V}_H$ and the output signals, assessing their level of association (lines 1-7). It then rejects a small portion of the internal signals with weaker correlations with the system outputs, based on the discard threshold (line 8). The resulting set of internal signals (i.e., $\hat{\mathcal{V}_H}$) is identified as those signals demonstrating strong association with the system output.

In the subsequent phase, the procedure employs the FindF subroutine (Algorithm 7.3) to identify the features of each internal signal $v \in \hat{\mathcal{V}_H}$ that significantly influence the robustness of ϕ. For every $v \in \hat{\mathcal{V}_H}$, the algorithm computes the feature values for all the features in the feature set `Feat` (line 2), and evaluates the correlation coefficient between the signal feature values and the robustness ρ (line 3). It marks a feature as relevant if it demonstrates a substantial impact on ρ, indicating that the absolute value of the correlation coefficient surpasses the correlation threshold θ_2. A threshold exceeding 50% is selected to identify a subset of significant features $\hat{\mathbf{X}}$ that exert either strong or very strong influence on the robustness values.

Next, the procedure incrementally adds a series of test cases to the existing test suite (line 11), and recalculates the robustness (line 12) and correlation values to further reduce the number of significant internal signals and significant features (lines 13-14).

As the final step, the procedure further reduces the number of significant signals and

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Algorithm 7.1: Feature Identification Procedure: Identifying significant features of significant signal variables that impact ϕ.

```
Input  : $\mathcal{M}$ - A CPS dataflow model.
         $\phi$ - An STL specification of $\mathcal{M}$.
         Feat - A set of signal features.
Output : $\hat{\mathcal{V}}_H$ - Significant signal variables.
         $\hat{\mathbf{X}}$ - Significant signal features.
1  $SimData = [\,]$; $Robustness = [\,]$
2  $\hat{\mathbf{X}} = [\,]$; $\mathbf{CORR} = [\,]$; $\mathbf{corr} = [\,]$
3  $\theta_1 = 0.99$ ;                // initialize redundancy threshold
4  $\theta_2 = 0.5$;                  // initialize correlation threshold
5  $\theta_3 = 0.2$;                  // initialize discard threshold
6  $\mathcal{TS} = \textsc{RandomTestSuite}(\mathcal{M})$
7  $SimData, Robustness \leftarrow \textsc{Monitor}(\mathcal{TS}, \mathcal{M}, \phi)$
8  $\hat{\mathcal{V}}_H \leftarrow \textsc{FindTopS}(SimData, \theta_3)$ ;   // identify internal signals
     correlated with output
9  $\hat{\mathbf{X}} \leftarrow \textsc{FindF}(\hat{\mathcal{V}}_H, SimData, \texttt{Feat}, Robustness, \theta_2)$ ; // identify features
     that correlate with robustness
10 for $q$ times do
11     $\mathcal{TS} = \mathcal{TS} \cup \mathcal{J}$ ;          // add tests to the test suite
12     $SimData, Robustness \leftarrow \textsc{Monitor}(\mathcal{TS}, \mathcal{M}, \phi)$
13     $\hat{\mathcal{V}}_H \leftarrow \textsc{FindTopS}(SimData, \theta_3)$
14     $\hat{\mathbf{X}} \leftarrow \textsc{FindF}(\hat{\mathcal{V}}_H, SimData, \texttt{Feat}, Robustness, \theta_2)$
15 end for
16 $\hat{\mathcal{V}}_H, \hat{\mathbf{X}} \leftarrow \textsc{Remove}(SimData, \theta_1, \hat{\mathcal{V}}_H, \hat{\mathbf{X}})$ ;     // discard redundant
     internal signals and associated features
17 return $\hat{\mathcal{V}}_H$, $\hat{\mathbf{X}}$
```

Algorithm 7.2: The subroutine FindTopS().

```
1 for each $v \in \mathcal{V}_H$ do
2     for each item $\mathbf{t} \in \mathcal{TS}$ do
3         $corr_v = \textsc{Correlation}(v, \mathcal{V}_{O_\mathbf{t}})$
4         $\mathbf{CORR} = \mathbf{CORR} \cup corr_v$
5     end for
6     $\mathbf{corr} = \mathbf{corr} \cup \mathbf{CORR}$
7 end for
8 $\hat{\mathcal{V}}_H \leftarrow \textsc{Reject}(\mathbf{corr}, \mathcal{V}_H, \theta_3)$ ;         // discard internal signals
    exhibiting low correlation with output
9 return $\hat{\mathcal{V}}_H$
```

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

Algorithm 7.3: The subroutine FindF().

```
1 for each $v \in \hat{\mathcal{V}_H}$ do
2     $A = \textsc{Feature}(Simdata_v, \texttt{Feat})$
3     $[\gamma] \leftarrow \textsc{Select}(A, Robustness, \theta_2)$ ;    // identify features that correlate with robustness
4     $\hat{\mathbf{X}} = \hat{\mathbf{X}} \cup [\gamma]$
5 end for
6 return $\hat{\mathbf{X}}$
```

features by eliminating the redundant internal signals and their associated features (line 16). To identify redundant internal signals, the procedure employs the function Remove, which calculates the correlation coefficients between the current set of significant internal signals using the Pearson method. It removes those that provide redundant information. Specifically, when the correlation coefficient between any two internal variables v_1 and v_2 exceeds the redundancy threshold θ_1, one of the variables is discarded from further analysis along with its associated features. The resulting set of significant internal signals and features represents the targets of FCT-S. Test case generation is ultimately defined by applying the BCA problem to maximize $\zeta_{\hat{\mathcal{V}_H}\hat{\mathbf{X}}}$, as described in the previous section.

7.3 Empirical Evaluation

We conduct empirical assessments of the proposed coverage criteria alongside its related coverage-based test generation strategies, comparing them against several state-of-the-art test generation methods. Specifically, we explore the following RQs:

RQ-7.1-Coverage. *To what extent do existing test generation methodologies explore the features of CPS Simulink® models based on signal feature coverage?* The first RQ aims to find out if covering the features of signals is a non-trivial problem. This RQ examines *the efficacy of current test generation methods in terms of signal feature coverage.* If current methods already produce test suites with extensive feature coverage, further investigation into maximizing coverage becomes redundant. To address this query, we compare the performance of three test generation techniques against our feature-coverage test generation method to check how thoroughly signals' features can be covered and identify any deficiencies in existing testing methods. Details of the considered test generation methods are outlined in Section 7.3.1.1. Additionally, we explore how variations in resource allocation affect the attained coverage, providing insights into the efficiency of resource allocation in enhancing signal feature coverage.

RQ-7.2-Fault Revealing Capability. *Do test suites that achieve high feature coverage also uncover more faults in CPS Simulink® models?* This question uses mutation testing (MT) to *evaluate the performance of feature coverage-based test generation methods,*

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

specifically FCT-C and FCT-S, in comparison to established testing methods. The MT procedure is detailed in Section 7.3.1.4, while the test generation techniques are discussed in Section 7.3.1.1.

7.3.1 Experimental Setup

Now, we will outline the test generation methods under comparison, the platform used for execution, the subjects of our experiments, and the mutations applied.

7.3.1.1 Test generation

In our analysis, we examine two groups of test generation methods: the two variations of feature-coverage testing (FCT-C and FCT-S) and six state-of-the-art test generation techniques for CPS Simulink models (namely ART, OD-VB, OD-FB, FT-BCA, FT-S-TaLiRO and SDV-EC), as discussed below.

- *Comprehensive Feature-Coverage Test Generation* (FCT-C) is designed to generate tests that cover all targets, including all signals and features. It aims to determine achievable levels of signal feature coverage.
- *Specification-directed Feature-Coverage Test Generation* (FCT-S) is the approach that focuses solely on important features of key signals ($\hat{\mathcal{V}}_H$ and $\hat{\mathbf{X}}$) identified using the Feature Identification Algorithm detailed in Section 7.2.3.2.
- *Adaptive Random Testing* (ART)[5] is a method for generating tests that prioritizes creating varied test cases "by maximizing distances between the test inputs" [CLM05, LNLB19], as previously mentioned in Chapter 6. It is often regarded as a baseline because of its simplicity and efficacy.
- *Output Diversity (OD)* is a method for generating tests that aims to create test cases with highly diverse outputs [LNLB19, MNBB19, MNBB15]. It employs a search-based approach that begins with an initial test suite and then iteratively enhances it by making small adjustments to the test cases to maximize output diversity. This technique evaluates neighboring test cases at each iteration and selects the one that notably increases output diversity. This process continues until further improvement is no longer feasible. In this study, we consider two variations of OD: *vector-based* (VB) and *feature-based* (FB), as proposed in [MNBB19]. The *vector-based* approach directly analyzes output signal vectors, while the *feature-based* method operates on feature vectors derived from output signals. This entails describing the *output* signals based on their *value* as well as their *first and second derivatives*, which help identify distinct signal shapes.
- *Simulink® Design Verifier™—Execution Coverage* (SDV-EC) is utilized to generate tests aimed at achieving model coverage. The test generation is configured to achieve *block execution coverage*, a structural coverage metric that indicates whether each

[5]Considering that ART generally outperforms random testing (RT) in terms of both failure detection and program-based coverage completeness [MZC+20, CKLW08], we have excluded RT from our comparison.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

block is executed during simulation. This coverage metric is most comparable to *statement coverage* in code coverage analysis. Note that Simulink® Design Verifier™ is a product from MathWorks[6].

- *Falsification Testing* (FT) is a search-based approach used to generate tests that violate system requirements or assumptions [ADD+17], as mentioned previously in Chapter 6. Essentially, given an STL property ϕ of a model $\mathcal{M}$, FT aims to find a test case $\mathbf{t}$ such that $\rho(\phi, O(\mathbf{t}, \mathcal{M})) < 0$, meaning $O(\mathbf{t}, \mathcal{M}) \not\models \phi$. In our experiments, we employ BCA to generate test cases guided by classical quantitative robustness. In essence, FT's goal is to create a test that violates the property being examined, with the aim of uncovering bugs. We also use the S-TaLiRo tool, as discussed in Chapter 4 (see Section 4.6.2.2), for coverage-guided falsification-based test generation.

To address **RQ-7.1**, we analyze the effectiveness of two feature-coverage testing methods (FCT-C and FCT-S), along with four state-of-the-art techniques: ART, OD-VB, OD-FB and SDV-EC. However, since FT is tailored for fault detection rather than coverage, we apply it exclusively to address RQ-7.2 in our MT experiments. Therefore, **RQ-7.2** considers all the aforementioned test generation strategies. It is worth noting that we conducted 30 independent runs of each test generation algorithm to mitigate the influence of random fluctuations and to provide statistical insights.

7.3.1.2 Simulation Platform

For our experiments, we employed MATLAB® Simulink® R2018b provided by MathWorks® and utilized the RTAMT library [NY20] for the offline assessment of STL properties. These experiments were conducted on a MacBook Pro running macOS Ventura, equipped with an Apple M1 chip and 16 GB of RAM.

7.3.1.3 Experimental Subjects

In addressing our RQs, we examined five diverse Simulink® models, sourced from [Mat22c, Mat22d, Mat22a, EAF+22]. Also, we considered ten requirements expressed as STL properties, representing essential conditions that these systems must adhere to. These models have been extensively utilized in prior research works [NKJ+17, SS20, BMNY22c, EAF+22].

Fault-Tolerant Fuel Control System (FCS). FCS serves as a benchmark within the automotive sector, simulating a fuel control system designed for a gasoline engine [Mat22c]. It is represented as a hybrid system, incorporating differential equations along with a switching condition. FCS operates with two inputs: throttle and engine speed, while producing two outputs: Fuel Flow Rate and the air-to-fuel ratio (λ). The primary control aim is to maintain λ in proximity to the reference mixture ratio (λ_{ref}).

Automatic Transmission Controller System (ATCS). This has been previously addressed in Chapter 5.

[6]https://in.mathworks.com/products/simulink-design-verifier.html

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

Table 7.3: Key features of selected Simulink models.

Model	#*Blocks*	#*Lines*	T	Sample time	#*Samples*
FCS	215	190	110	0.01	11001
ATCS	65	92	30	0.04	751
AECS	825	577	10	0.01	1001
NN	105	123	40	0.001	40001
SC	173	144	35	0.01	3501

Aircraft Elevator Control System (AECS). This has been already discussed in Chapter 5.

Neural-Network Controller (NN). The neural network controller is engineered to uphold a magnet at a specific reference position over an electromagnet, securing a stable hover [EAF$^+$22,Mat24]. Operating with a reference position value (Ref) ranging between 1 and 3 as input, the model outputs the actual position (Pos) of the hovering magnet. The aim is to guarantee that after alterations to the reference, the present position consistently aligns with the reference with minimal deviation.

Steam Condenser with Recurrent Neural Network Controller (SC). The model, initially presented in [YF19] and also employed in [EAF$^+$22], simulates a dynamic steam condenser model controlled by a Recurrent Neural Network in feedback. Operating within an input range of $[3.99, 4.01]$, the main objective is to maintain the output pressure consistently within a predefined tolerance range.

Table 7.3 provides an overview of the essential features of the five models, including the count of blocks, lines, simulation time denoted as T (in seconds), sampling time (in seconds), and the total number of samples. Additionally, Table 7.4 outlines their requirements described in natural language and in STL.

7.3.1.4 Property-Based Mutation Testing

To conduct the MT experiments necessary for addressing RQ-7.2, we utilize *Property-Based Mutation Testing* (PBMT), as previously outlined in Chapter 6. In our experimental setup, we employ FIM [BMNY22a], our tool designed for automatically injecting faults and mutations into Simulink models, as detailed in Chapter 3. We utilize seven fault types/mutation operators provided by FIM, which are commonly utilized in Simulink model testing scenarios. These include three line mutations (`Noise`, `Bias/Offset`, `Negate`) and four block mutations (`ROR`: Relational Operator Replacement, `S2P`: Sum to Product Mutation, `P2S`: Product to Sum Mutation, `ASR`: Arithmetic Sign Replacement). Notably, we opt out of using FIM's `Absolute` operator due to its tendency to generate mutants that are ϕ-trivially different from the original (ϕ-trivial mutants) [BMNY23]. Similarly, we exclude the `Invert` and `Bit-flip` operators due to their tendency to produce invalid mutants. Table 7.5 provides insights into the number of valid mutants (those that could be compiled) generated for each mutation operator across the various

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Table 7.4: Properties of experimental subjects in natural language and STL.

Property	Ref.	Natural Language	STL
Fault-Tolerant Fuel Control System (FCS)			
ϕ_1^{FCS}	[NKJ+17]	Always, within the interval $[\tau, T]$, the air-to-fuel ratio (λ) must not evolve beyond the tolerance bounds.	$\square_{[\tau,T]}(0.8\lambda_{ref} \leq \lambda \leq 1.2\lambda_{ref})$
ϕ_2^{FCS}	[HAF14]	The fuel flow rate should not be 0 for more than 1 second within the next 100 seconds period.	$\neg\lozenge_{[0,100]}\square_{[0,1]}(\text{Fuel Flow Rate} = 0)$
Automatic Transmission Controller System (ATCS)			
ϕ_1^{ATCS}	[HAF14]	"The engine speed never reaches $\bar{\omega}$".	$\square(\omega < \bar{\omega})$
ϕ_2^{ATCS}	[HAF14]	"The engine and the vehicle speed never reach $\bar{\omega}$ and $\bar{v}$, resp."	$\square\big((v < \bar{v}) \wedge (\omega < \bar{\omega})\big)$
ϕ_3^{ATCS}	[HAF14]	"If engine speed is always less than $\bar{\omega}$, then vehicle speed cannot exceed $\bar{v}$ in less than T seconds."	$\neg\big(\lozenge_{[0,T]}(v > \bar{v}) \wedge \square(\omega < \bar{\omega})\big)$
ϕ_4^{ATCS}	[HAF14]	"Within T seconds, the vehicle speed is above $\bar{v}$ and from that point on, the engine speed is always less than $\bar{\omega}$)."	$\lozenge_{[0,T]}\big((v \geq \bar{v}) \wedge \square(\omega < \bar{\omega})\big)$
Aircraft Elevator Control System (AECS)			
ϕ_1^{AECS}	[BMM+21]	Whenever the Pilot Command, denoted as *cmd*, exceeds the threshold m, the measured actuator position, *pos*, must not exceed a distance of n units from *cmd* within a time frame of Q+a units.	$\square\big(\uparrow (cmd \geq m) \rightarrow \lozenge_{[0,Q]}\square_{[0,a]}(\lvert cmd - pos\rvert \leq n)\big)$
Neural-Network Controller (NN)			
ϕ_1^{NN}	[EAF+22]	Within the time horizon of 40 time units, the output will be always close to the reference signal within 2 seconds.	$\square_{[1,37]}\big(\lvert Pos - Ref\rvert > \alpha + \beta\lvert Ref\rvert \rightarrow \lozenge_{[0,2]}\square_{[0,1]}\neg(\alpha + \beta\lvert Ref\rvert \leq \lvert Pos - Ref\rvert)\big)$
ϕ_2^{NN}	[EAF+22]	Given the reference signal Ref satisfying $1.95 \leq Ref \leq 2.05$, the output satisfies a conjunctive requirement defined over different time intervals.	$\lozenge_{[0,1]}(Pos > 3.2) \wedge \lozenge_{[1,1.5]}\big(\square_{[0,0.5]}(1.75 < Pos < 2.25)\big) \wedge \square_{[2,3]}(1.825 < Pos < 2.175)$
Steam Condenser with Recurrent Neural Network Controller (SC)			
ϕ_1^{SC}	[EAF+22]	Always, within the interval $[30, 35]$, the pressure never goes below 87 and beyond 87.5..	$\square_{[30,35]}(87 \leq Pressure \leq 87.5)$

Parameters: $\lambda_{ref} = 14.6$, $\tau = 10$ sec, $\bar{\omega} = 4500$ RPM, $\bar{v} = 120$ mph, $P = 4$ sec, $m = 0.09$, $Q = 2$ sec, $a = 1$ sec, $n = 0.02$, $\alpha = 0.005$, $\beta = 0.03$

Table 7.5: Mutation operators and number of mutants.

Type	**Operator**	**# Mutants**				
		FCS	ATCS	AECS	NN	SC
Line Mutation	`Noise`	10	13	17	10	10
	`Bias/Offset`	12	13	17	10	10
	`Negate`	10	13	17	0	0
Block Mutation	`ROR`	0	0	10	0	0
	`S2P`	1	1	3	2	3
	`P2S`	0	2	6	1	2
	`ASR`	3	3	8	3	9
Total		36	45	78	26	34

subjects. For every valid FOM[7] produced by FIM, we presume that the fault remains active for the entire duration of the simulation, commencing at time 0 and persisting until the end at time T.

For our PBMT evaluations, we utilize all the test generation techniques outlined in Section 7.3.1.1. It is noteworthy that this selection allows for a comparison between FCT-S (property-dependent) and FCT-C (property-independent) against other property-dependent (FT) and property-independent (ART, OD-VB, and OD-FB) methods. Additionally, in our FT-based test generation setup (for both BCA and S-TaLiRo) for PBMT, FT is applied individually to each mutant. To elaborate, for every mutant $\mathcal{M}'$, FT attempts to generate a test case $\mathbf{t}$ that violates the property, denoted as $O(\mathbf{t}, \mathcal{M}') \not\models \phi$.

7.3.2 Results and Discussion

RQ-7.1 - Coverage

RQ-7.1 assesses the effectiveness of state-of-the-art test generation strategies in thoroughly exercising features. As previously noted, the interconnected nature of elements in dataflow models often results in the execution of multiple elements with just a few test cases. This observation was confirmed in our experiments, where a single random test case was sufficient to cover all signals in the ATCS model, and on average, 9.25 random test cases were enough to cover more than 90% of the signals in the other four models. This highlights the necessity for more advanced coverage criteria. Consequently, we compare the signal feature coverage achieved by traditional test generation strategies with that of our proposed FCT-S and FCT-C techniques.

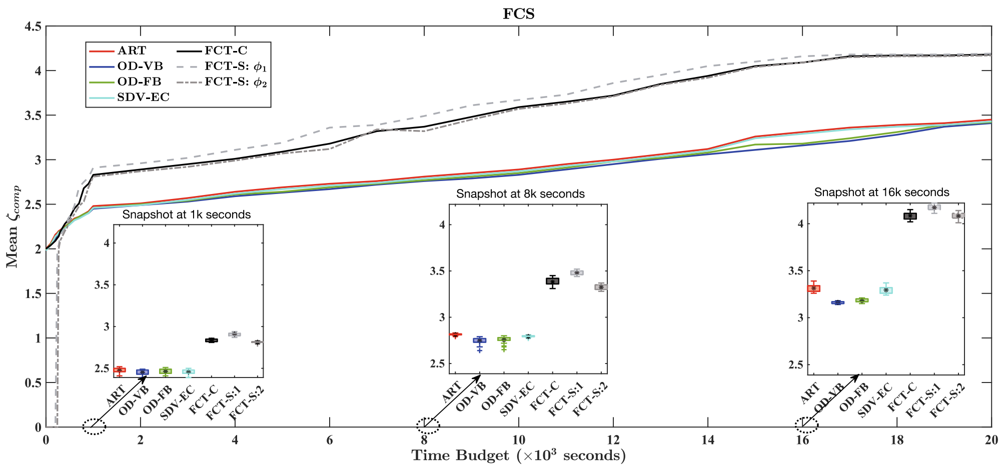
Figure 7.2 shows the trends of ζ_{comp} over time, averaged across 30 independent runs for each experimental subject and all properties. For a detailed view of the variability

[7] In our study, we exclusively consider *first-order mutants* of Simulink models.

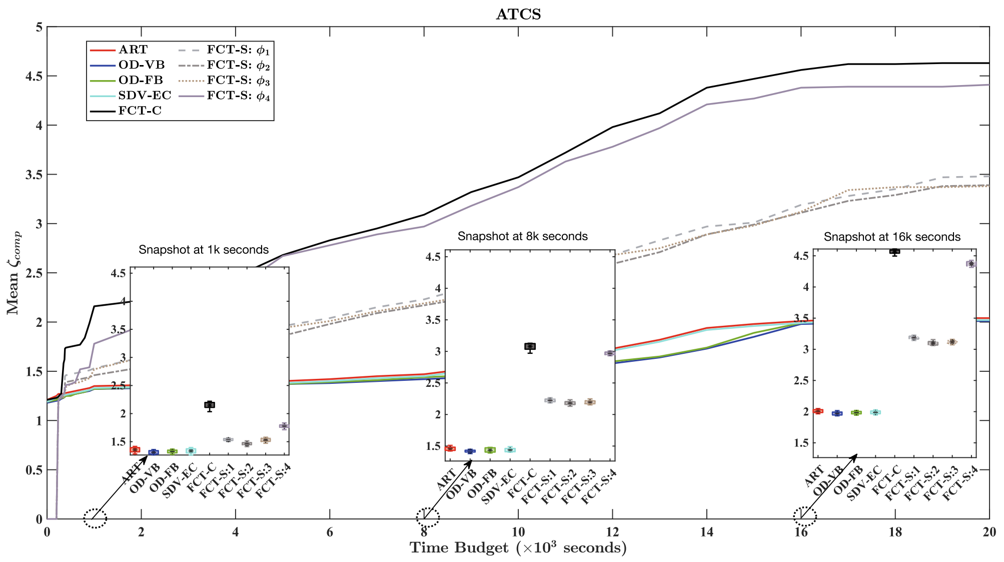
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

in the ζ_{comp} values across all techniques, refer to Figure 7.3. It is important to note that ART, OD-VB, OD-FB, SDV-EC and FCT-C are property-independent techniques, while FCT-S is property-directed. Thus, separate curves are shown for FCT-S for each property. Additionally, because FCT-S includes a feature identification step to determine significant signals and features, the initial coverage with FCT-S starts at zero.

Analyzing the figure reveals an initial brief period where our proposed FCT-C and FCT-S techniques show lower performance in terms of the coverage metric. However,



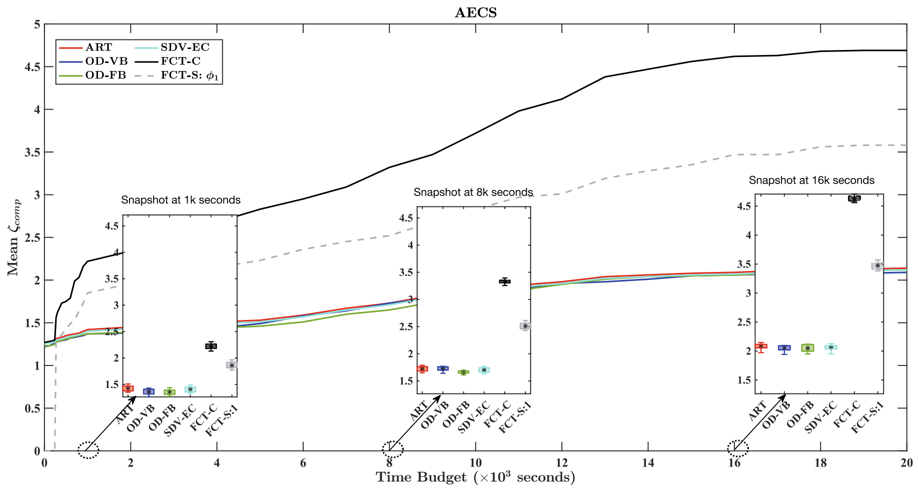
(a) FCS model



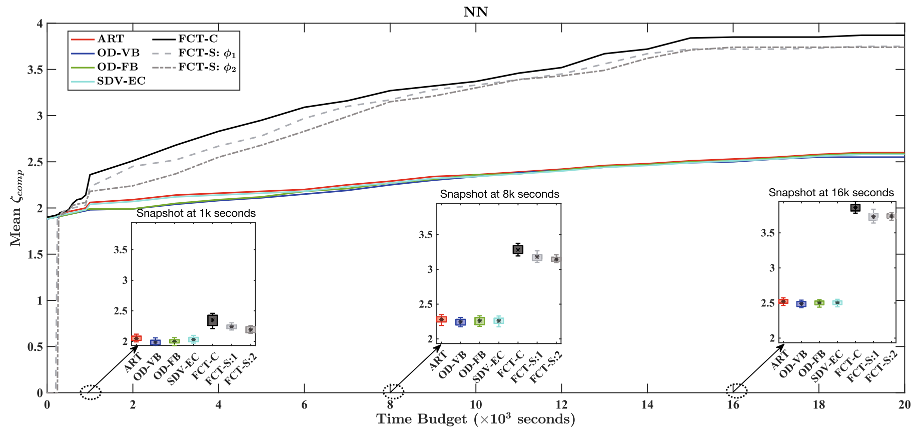
(b) ATCS model

Figure 7.2: Trends over time of mean value of ζ_{comp} for different test generation strategies for various models.

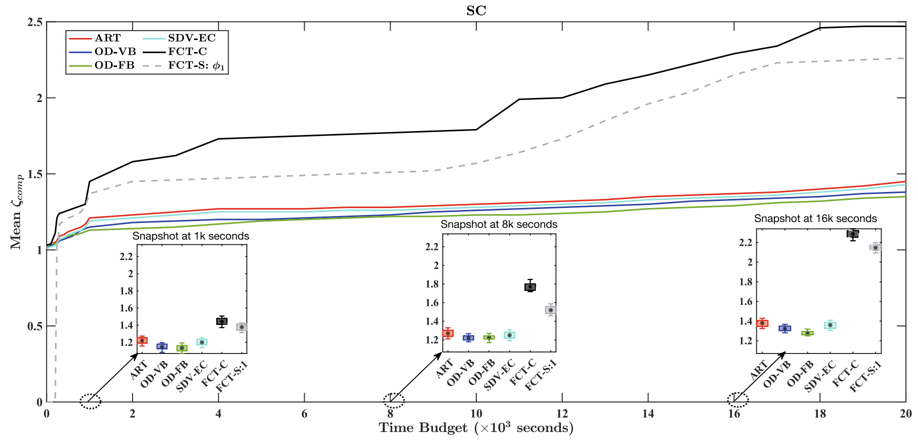
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



(c) AECS model



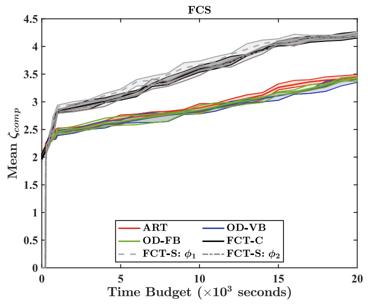
(d) NN model



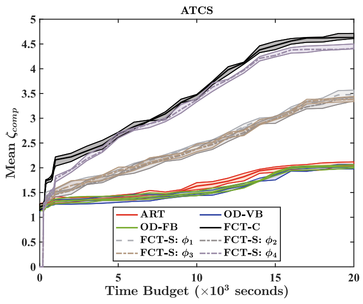
(e) SC model

Figure 7.2: Trends over time of mean value of ζ_{comp} for different test generation strategies for various models. (cont.)

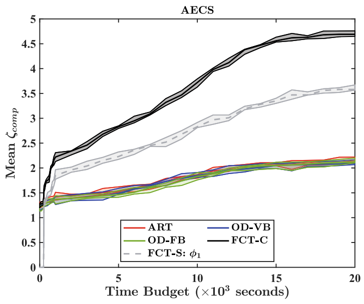
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub



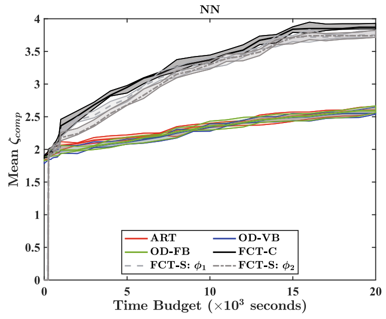
(a)



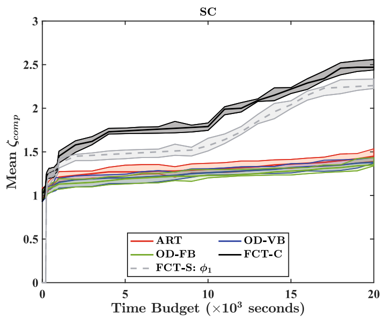
(b)



(c)



(d)



(e)

Figure 7.3: Trends over time of ζ_{comp} achieved by different test generation strategies: Tubular visualization indicating minimum, maximum and mean values.

as time progresses, both techniques exhibit a rapid improvement, eventually surpassing the state-of-the-art methods. This indicates that our techniques are capable of achieving high coverage given sufficient time and resources. Table 7.6 details (1) the time required for FCT-S to identify significant signals and features, referred to as *Feature Identification Time* (FIT), and (2) the time for FCT-C/FCT-S to exceed the performance of state-of-the-art methods in terms of ζ_{comp}, referred to as *Surpass Time* (ST). According to Table 7.6, the feature identification step takes an average of approximately 252.39 seconds across all experimental subjects. Additionally, FCT-C consistently outperforms FCT-S in surpassing state-of-the-art strategies, achieving higher ζ_{comp} values earlier than FCT-S.

Upon examining Figure 7.2, it is evident that the curves representing the mean ζ_{comp} for various state-of-the-art strategies, namely ART, OD-VB, OD-FB and SDV-EC, either overlap or closely align, indicating comparable performance among these techniques. In particular, there is a considerable overlap between the curves for ART and SDV-EC. As expected, the Comprehensive (FCT-C) approach, aimed at covering all features and signals comprehensively, consistently achieves the highest coverage levels compared to other methods, including the Specification-directed (FCT-S) technique. However, there is an exception to this trend. Specifically, for the FCS model assessed against property ϕ_1^{FCS}, the FCT-S method achieves slightly higher coverage values than FCT-C. We hypothesize that this superior performance of FCT-S may be attributed to its effectiveness in covering

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Table 7.6: Time required for identifying significant signals and features, and times at which FCT-S/FCT-C surpass state-of-the-art strategies w.r.t. ζ_{comp}.

Model	**Property**	**FIT (s)**		**ST for FCT-S (s)**		**ST for FCT-C (s)**	
		Average	**Worst**	**Average**	**Worst**	**Average**	**Worst**
FCS	ϕ_1	241.33	254.81	450.23	462.56	400.67	405.34
	ϕ_2	270.24	279.65	471.02	482.10		
ATCS	ϕ_1	243.67	278.67	380.23	397.24	352.67	365.38
	ϕ_2	246.73	269.66	384.56	395.26		
	ϕ_3	245.79	265.57	379.68	387.91		
	ϕ_4	244.83	271.58	372.59	380.26		
AECS	ϕ_1	272.80	293.72	406.27	411.84	274.56	281.67
NN	ϕ_1	246.52	270.43	403.86	415.27	304.53	311.25
	ϕ_2	271.48	291.27	502.36	516.28		
SC	ϕ_1	240.56	261.78	306.74	309.38	245.56	253.38

key features of key signals, leading to extensive coverage of multiple features across various signals. This observation suggests that the FCT-S strategy might capture certain feature values that could be challenging to cover with the Comprehensive (FCT-C) approach.

Looking at Figure 7.2, it is evident that both FCT-C and FCT-S tend to saturate towards the later stages of the experiments, usually around 16k-18k seconds. This suggests that achieving further coverage enhancement may become increasingly difficult. On the contrary, state-of-the-art techniques do not show full saturation in ζ_{comp} values, indicating that there is still room for improvement. This implies that these techniques may require additional time resources to generate tests that effectively cover the signal features.

In Figure 7.2, we also present the variability in ζ_{comp} across 30 runs for each test generation strategy using box plots captured at different time snapshots. Each box in the plot contains 30 data points, where each data point signifies the ζ_{comp} achieved by a test suite generated in a single run. The horizontal axis denotes the test generation approach, while the vertical axis represents the corresponding ζ_{comp} values. Within each box, an asterisk (*) denotes the mean value of ζ_{comp}. Examining Figure 7.2a, we notice minimal variation in the data for the FCS model, except for specific time snapshots. For instance, at the 8k second snapshot, both OD-VB and OD-FB show outliers (illustrated as lower whiskers in their respective boxes), indicating that certain test suites generated using these techniques achieve coverage levels lower than the mean value. Furthermore, upon analyzing all the box plots of the FCS model across various time snapshots, we observe that ART and SDV-EC exhibit slightly better performance in terms of ζ_{comp} among the traditional strategies. However, their performance falls behind when compared to our feature-coverage testing techniques.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

In comparison to the other models, the test suites generated for the NN model (Figure 7.2d) demonstrate slightly greater variations in the ζ_{comp} values. Furthermore, for certain models, notably FCS (Figure 7.2a) and AECS (Figure 7.2c), across all time snapshots, the box plots associated with FCT-C and FCT-S closely converge at similar levels. This convergence implies that FCT-C and FCT-S exhibit comparable effectiveness in achieving coverage levels for the specified properties in these models.

Across all experimental subjects, traditional ART and OD-based test generation methods demonstrate notably low coverage levels. On the other hand, FCT-C and FCT-S approaches employ focused test generation designed to maximize feature coverage. Consequently, the generated tests thoroughly exercise the target features of the internal signals, resulting in significantly higher coverage levels compared to conventional test generation methods, as depicted in Figure 7.2. These findings underscore the existence of a gap in the capability of baseline techniques in covering signal features in CPS Simulink models.

It is crucial to acknowledge that the signals and features identified through our feature identification procedure may not perfectly align with the *true* signals and features influencing the STL property. Discrepancies could exist between the identified signals and features and those genuinely affecting the satisfaction of the considered property, referred to as the *ground truth* (GT) signals and features. Moreover, there is not a definitive method for identifying the GT signals/features, as it is at the discretion of the tester or involves selecting a specific procedure for this purpose. To assess the performance of our feature identification step in capturing these features, we conduct correlation analysis over a large set of tests among internal signals, features, and the quantitative robustness of simulation traces with respect to the STL properties. This comprehensive analysis spans an extended duration, encompassing 10,000 random tests for each model and property, totaling approximately 74 hours across all our experimental subjects and properties.

After determining the GT signals and features, we assessed the extent to which the generated test suites cover the GT signals and features. Specifically, we assessed the *ϕ-driven Signal Feature Coverage* of the *ground truth* signals and their associated features. Figure 7.4 presents the trends of ζ_ϕ, which represent the *ϕ-driven signal feature coverage* computed for the ground truth signals and features, averaged over 30 runs. Additionally, Figure 7.5 illustrates the variability in ζ_ϕ across all models and properties. Notably, we have a separate plot for each property of a model since ground truth features and signals are specific to each property. Also, it is worth noting that the FCT-S-based test suites utilized to compute ζ_ϕ-values for GT features and signals are the ones that we generated focusing on maximizing coverage of signals and features identified through the feature identification procedure. This approach provides valuable insights into the effectiveness of our time-efficient feature identification procedure and offers indications of any correlation between the *identified signals+features* and the *ground truth signals+features*.

Upon analyzing Figure 7.4, distinct patterns emerge, delineating three main categories within the curves. Firstly, one set of curves suggests that the state-of-the-art testing approaches inadequately exercise the *ground truth* features and signals, failing to extensively

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

cover the signals most relevant to the property under test. Secondly, another set of curves indicates that FCT-C achieves better coverage of these features and signals compared to the state-of-the-art methods, highlighting the need for specific methods to thoroughly cover relevant features of relevant signals. Lastly, a third set of curves showcases that FCT-S surpasses both FCT-C and the state-of-the-art, achieving higher coverage of the GT features and signals. This trend holds true across all experimental models and properties. This observation underscores the effectiveness of our feature identification procedure in identifying features and signals that contribute to the coverage of GT signals and features, implying a correlation between them. Additionally, from Figure 7.4, we note significant overlaps among the curves corresponding to the state-of-the-art testing strategies across all experimental models and properties, indicating their comparable performance in exercising the GT signal features.

Figure 7.3 presents the trends over time of ζ_{comp} across various test generation techniques. Employing a tubular visualization, we encapsulate the ζ_{comp} values, integrating minimum, maximum, and mean statistics. The tubular boundaries delineate the range between minimum and maximum values, with the mean value represented by a bold line within this range. Upon examining Figure 7.3 for the variability in ζ_{comp} values, it is apparent that the tubes are closely grouped, indicating minimal dispersion between the extreme (i.e., minimum and maximum) values. Furthermore, significant overlaps are observed among the tubes corresponding to state-of-the-art techniques across most of our experimental models and properties. Similarly, Figure 7.5 illustrates the variability in ground truth signal feature coverage ζ_{ϕ} trends. This visualization employs the same tubular format, capturing minimum, maximum, and mean ζ_{ϕ} values. Notably, the tightly packed tubes suggest limited variance in the data. Due to the significant overlap among the data points for ART- and SDV-EC-based test suites, we have excluded the SDV-EC curves from Figure 7.3, Figure 7.4, and Figure 7.5 to enhance the readability of the plots. In both Figure 7.3 and Figure 7.5, a distinct contrast in performance is evident between our feature-coverage testing techniques and the state-of-the-art methods, with the former demonstrating superior efficacy.

RQ-7.2 - Fault Revealing Capability

RQ-7.2 explores whether test suites generated based on signal feature coverage are more effective at detecting faults compared to those created using state-of-the-art methods. Specifically, we assess the mutant-killing capability of test suites produced by ART, FT-BCA, FT-S-TaLiRo, SDV-EC, OD-VB, and OD-FB, and compare them to those generated by our FCT methods, namely FCT-C and FCT-S.

Figure 7.6 presents the trends over time of the number of mutants ϕ-killed by test suites generated using different methods. For clearer visualization and easier interpretation, the plots display the average number of mutants ϕ-killed. For a detailed view of the variability in the number of mutants ϕ-killed, readers can refer to Figure 7.7. For the mutant-killing trends for various test suites, there is significant overlap between the

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

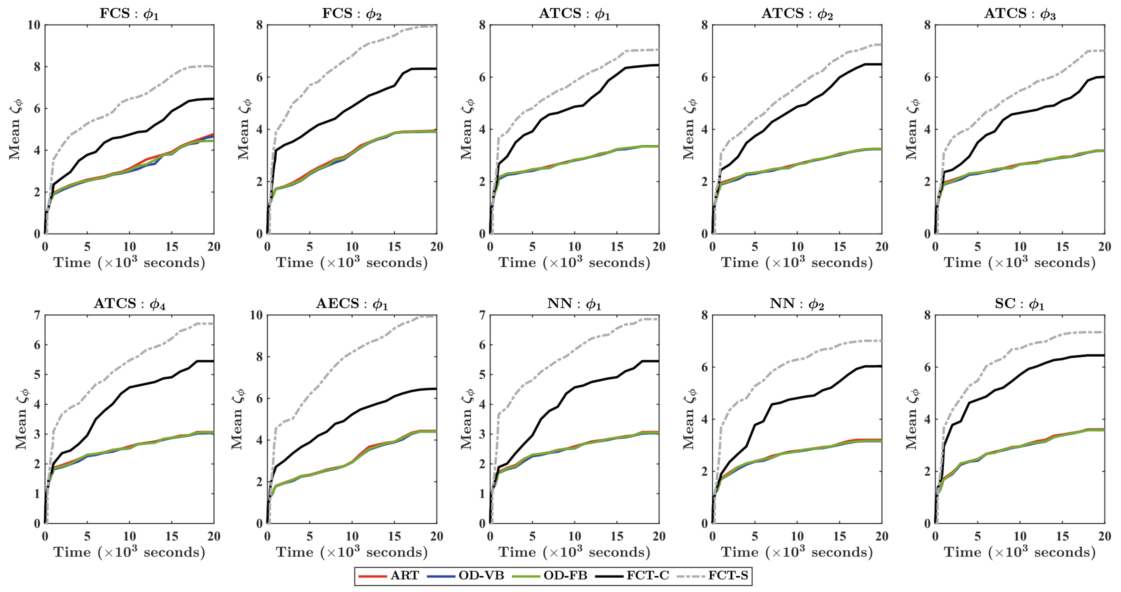


Figure 7.4: Trends over time of the mean ζ_ϕ-values computed for *ground truth* signals and features.

curves for ART and SDV-EC-based test suites. As a result, in each subplot of Figure 7.6 and Figure 7.7, only one of these curves is clearly visible at a time.

We verified the ϕ-killability of the mutants using the mutant-directed test generation strategy outlined in [BMNY23], and also outlined in Algorithm 6.1 discussed in Chapter 6. All generated mutants were confirmed to be ϕ-killable. As illustrated in Figure 7.6, distinct patterns are evident among the curves. The curves for FCT-C and FCT-S consistently show a significantly higher number of ϕ-killed mutants compared to all other state-of-the-art testing methods. This trend is consistent across all models and properties, providing strong evidence for the effectiveness of generating test cases that thoroughly exercise the features of the internal signals.

An interesting observation is that the FT approach (both BCA and S-TaLiRo), which focuses on property falsification, does not achieve a higher mutant kill count than other state-of-the-art methods. In fact, it consistently performs less effective compared to FCT-S and FCT-C across all system models and their respective properties. This suggests that targeting falsification may be less effective in identifying faults than the strategies employed by FCT-S and FCT-C. Similarly, the ART-based test generation, which prioritizes diversity among inputs without considering the property, also shows lower mutant-killing effectiveness. ART-generated test cases fail to sufficiently (i) explore the internal behavior of the software and (ii) detect basic faults, such as those seeded using FIM, that could impact the property. This outcome aligns with previous findings in the literature [BMNY23], further highlighting the inadequacy of ART-generated test cases for effective fault detection. It is also observed that the performance of SDV-EC-based

TU Bibliothek WIEN Your knowledge hub
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

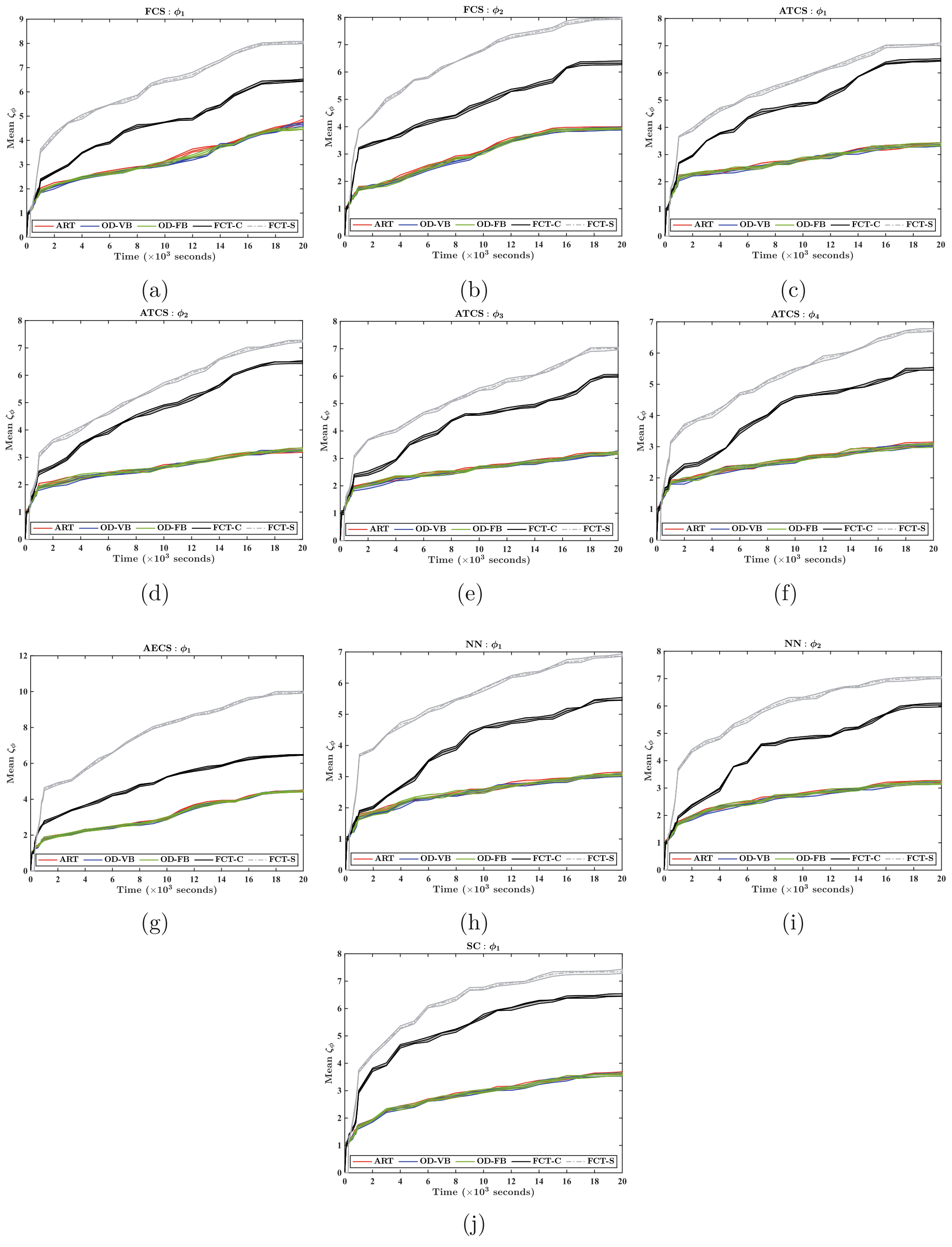


Figure 7.5: Trends over time of ζ_ϕ achieved by different test generation strategies: Tubular visualization indicating minimum, maximum and mean values.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

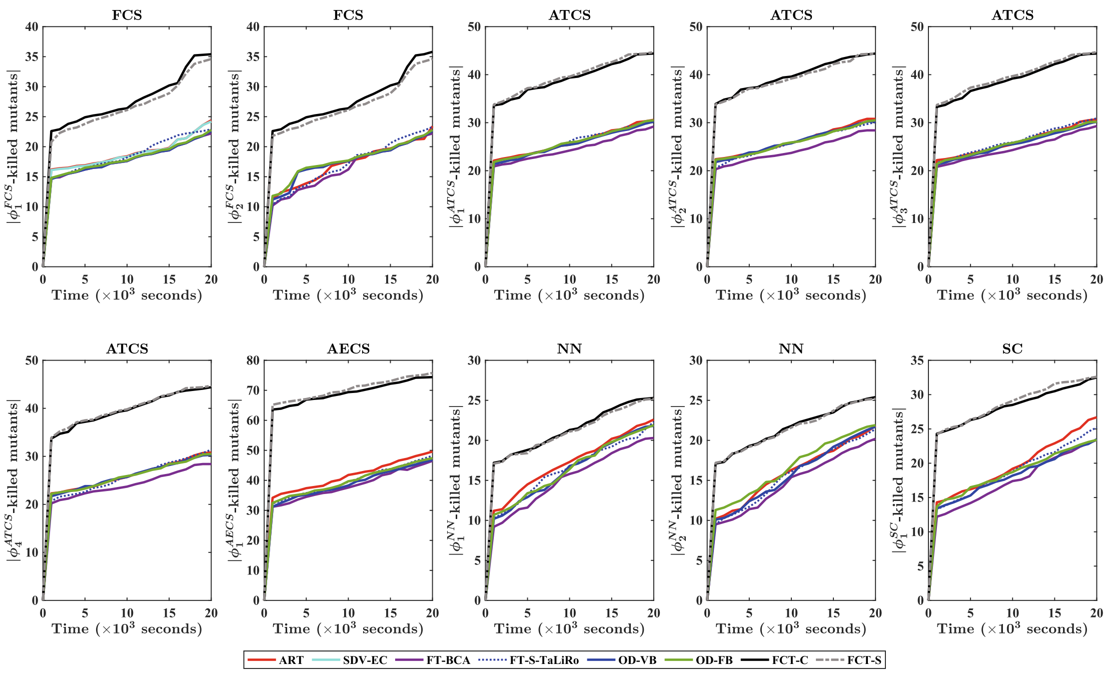


Figure 7.6: Trends over time of the average number of ϕ-killed mutants by different test generation strategies.

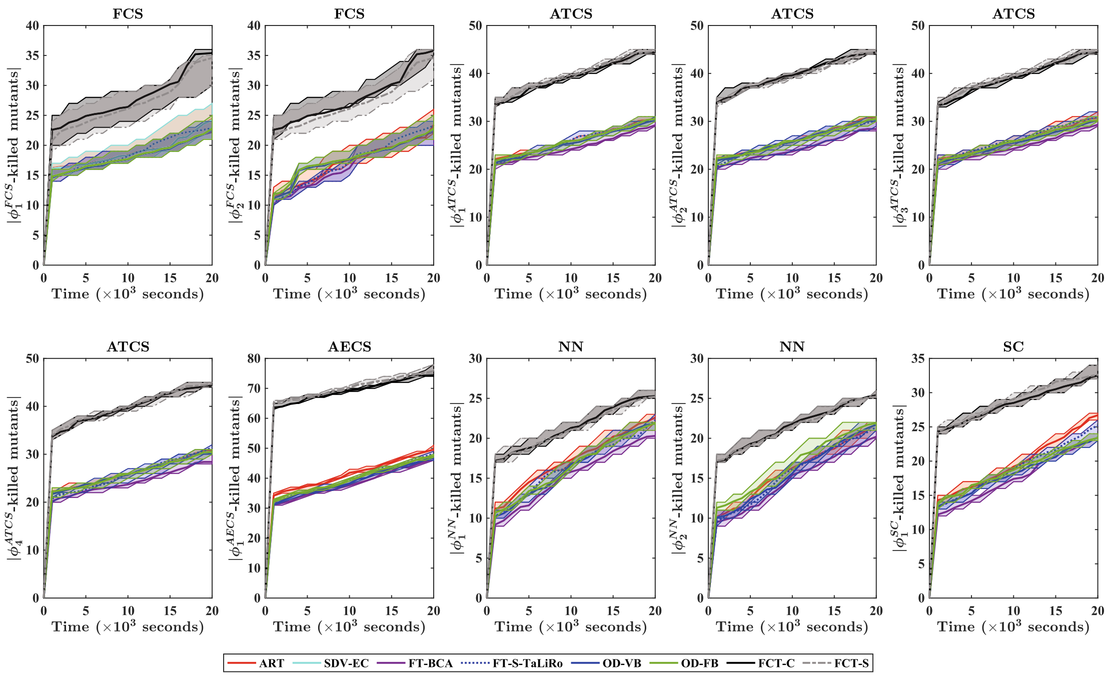


Figure 7.7: Trends over time of the number of ϕ-killed mutants by different test generation strategies: Tubular visualization indicating minimum, maximum and mean values.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

test suites in ϕ-killing the mutants is very similar to that of ART-based test suites, including their signal feature coverage.

Furthermore, output diversity-based testing strategies (OD-VB and OD-FB) show limited effectiveness in fault detection. These strategies aim to increase the likelihood of uncovering failures by diversifying test output signals, which enhance the chances of detecting significant discrepancies between expected and actual outputs. However, our PBMT evaluations across various models and properties suggest that the low mutant-killing performance of OD-VB and OD-FB results from insufficient exploration of the feature values within internal signals. This inadequate coverage prevents these strategies from effectively identifying faults. More specifically, the OD-FB approach focuses on diversifying the feature vectors of output signals but fails to diversify the features within internal signals. We believe this shortcoming is due to the complex interconnections and interactions between components and signals within the system. In essence, OD-VB and OD-FB testing methods fall short in thoroughly examining the internal behavior of the system model.

In contrast, FCT-S and FCT-C produce test cases that achieve greater feature coverage, significantly improving their effectiveness in detecting and capturing system errors, leading to a higher mutant-killing capability. These findings highlight the substantial benefits of generating tests that cover a wide range of features and feature values. Such tests are more likely to reveal faults by exercising them in ways that noticeably impact safety properties. By focusing on the features of internal signals, feature-coverage testing increases the chances of activating faults or erroneous behaviors linked to those features. This approach is particularly effective because faults in CPSs are often associated with specific signal characteristics or interactions between signals.

In summary, FCT-S and FCT-C proved to be the most effective test case generation strategies, achieving the highest mutant kill count. The tests generated by FCT-S and FCT-C exercise mutants in a manner that modifies the features of internal signals, causing failures to propagate to the observable interface and resulting in system property violations. Moreover, the superior performance of FCT-C and FCT-S underscores the importance of diversity in the feature values covered by test cases. This diversity enables a more thorough exploration of the system's response space, increasing the likelihood of uncovering various types of faults or corner cases that might be overlooked by feature-insensitive testing approaches.

In Figure 7.7, we present the trends over time for the number of mutants ϕ-killed with respect to a given property ϕ, showing minimum, maximum, and mean values. This visualization provides a comprehensive view of the variability in mutant killing across different scenarios, enhancing the understanding of overall performance. Examining Figure 7.7, we observe that test suites generated by FCT-S and FCT-C demonstrate the ability to kill the highest number of mutants, reaching a maximum mutation score of 100% (i.e., killing all ϕ-killable mutants) at around 18k seconds (approximately 5 hours) for most experimental subjects. Additionally, we notice significant overlap among the tubes representing the state-of-the-art testing strategies for certain subjects. In particular,

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

in the FCS (property 1), AECS, NN (property 1), and SC models, the maximum number of mutants killed appears slightly higher for ART-based and SDV-EC-based test suites. However, it is crucial to emphasize that the overall mutant killing count for ART-based and SDV-EC-based suites remains substantially lower compared to even the minimum number of mutants killed by our FCT-C and FCT-S techniques.

Another important finding is that the Specification-directed feature-coverage testing strategy, FCT-S, exhibits similar performance to the baseline FCT-C method. Across most of our experimental models and properties, the differences in the mean number of mutants ϕ-killed by FCT-C and FCT-S are marginal. This parity in performance between FCT-S and FCT-C-based testing methods implies that while the specification-directed approach thoroughly exercises signals and features strongly correlated with the tested property, it may not necessarily improve mutant killing compared to the feature-comprehensive strategy.

In our concluding analysis, we investigated the correlation between coverage values of individual signal feature types and mutation scores to assess if particular features notably impact mutation-killing capabilities. However, we did not observe a consistent pattern across all our experimental subjects concerning the influence of individual signal features on mutation-killing effectiveness. We aim to explore further the role of signal features in fault detection as part of our future research endeavors.

7.3.3 Threats to Validity

Our evaluation is influenced by the following threats to validity:

External Validity: Like many studies, ours faces an external validity concern regarding the generalizability of the results. To address this, we tested multiple properties and CPS models across diverse application domains. The consistency of the results across all properties and applications suggests that our findings provide a valuable and motivating foundation for further investigation into the concept of signal feature coverage. Another potential threat is the representativeness of the injected faults. Our results are based on common faults and mutation operators specific to Simulink® models, as supported by FIM (discussed in Chapter 3), to offer a fair and relevant evaluation with hundreds of injected faults. Conducting an initial study of this scale with real Simulink® faults would be challenging due to the necessity of a large dataset of actual faults.

Internal Validity: An internal validity concern involves the generated mutants. We focused on FOMs, consistent with common research practices in MT, and supported by evidence indicating their reliability as predictors of test suite effectiveness [JJI+14]. Exploring multi-fault Simulink® models remains an avenue for future research.

Conclusion Validity: One potential concern regarding conclusion validity relates to the randomness inherent in the test generation methods. To address this, we conducted each test case generation process 30 times. We reported both average results (as shown in Figure 7.2, Figure 7.4, and Figure 7.6) and results regarding the variance of the computed

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

metrics (depicted in Figure 7.3, Figure 7.5, and Figure 7.7). The uniformity of findings across all subjects and the minimal variation in results suggest that the presence of randomness does not pose a significant threat to our conclusions.

7.4 Related Work

In recent years, there has been a growing interest in coverage-based testing approaches for CPS dataflow models. One prominent metric is *input coverage*, which focuses on achieving diversity among input signals [LNLB19]. Similarly, *output coverage* aims to diversify system outputs [MNBB19]. The studies presented in [MNBB19, MNBB15, LNLB19] explore methods for generating tests that maximize output signal diversity in Simulink® models by guiding the process based on the output values that need to be tested. The underlying idea is that diversifying the output signals increases the chances of detecting significant discrepancies between expected and actual outputs, thereby enhancing failure detection. *Structural coverage* metrics for CPS dataflow models encompass block execution coverage, decision coverage, condition decision coverage, modified condition decision coverage, and node/edge/path coverage [Mat23b, TBAA15, AERP04]. While traditional coverage criteria typically focus on input values, output responses, and structural elements, they fall short of fully capturing the complexities and dynamic behaviors inherent in CPS dataflow models. The dynamics of CPS dataflow models are particularly influenced by internal signal features and interactions. Conventional testing strategies that rely solely on input/output values fail to ensure comprehensive testing of the internal behavior of the system, often missing critical faults, as highlighted by our empirical findings. *Signal feature coverage* addresses these shortcomings by explicitly considering the internal signal features within CPS dataflow models, thereby more effectively revealing system faults.

Several researchers have explored the use of signal features to improve the testing of CPS models. Zander [ZN09] introduced a detailed taxonomy of signal features tailored for dataflow models, advocating for their use in describing specifications and test oracles. Similarly, Matinnejad et al. [MNBB19] examined signal features, highlighting the importance of diversifying them to enhance test generation processes. Other researchers have argued that internal signals are valuable for fault identification and localization [LNLB19, GRS+16]. Liu et al. [LNLB19] suggested that considering internal signals can enhance fault localization in Simulink models, as faults become more detectable when analyzing internal signals or intermediary outputs. Similarly, Gay et al. [GRS+16] proposed that focusing on the final outputs of a Simulink model might mask the effects of internal faults. In contrast, examining subsystem outputs can make the impact of faults more evident. In this thesis, we contribute to this field by proposing a novel approach based on *signal feature coverage*. This approach guides test generation to focus on exercising the *features* of *internal* signals within CPS dataflow models, thus enhancing the effectiveness of the testing process.

Another research avenue explores integrating formal methods [MWR+15] and model checking techniques [NGM+19] with coverage-based testing. These methods use for-

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

mal specifications to guide test case generation and verify system properties [DDD+15, DYHF17, MYPF19]. For instance, Bartocci et al. [BBM+22] propose an adaptive test generation strategy to achieve specification coverage for temporal logic properties. Additionally, Dokhanchi et al. [DZS+15] enhance falsification methods by applying coverage metrics to the state space of hybrid systems. Our approach, however, leverages formal specifications (specifically, system safety properties expressed in STL) to pinpoint critical signals and features, which are then used to generate test cases focused on signal feature coverage. Furthermore, the automation of test case generation has been explored using techniques such as genetic algorithms [OHY11, AWM+18] and search-based testing [TAC+21, ZC08]. In line with these search-based strategies for achieving specific test objectives, we utilize BCA to generate test cases that maximize signal feature coverage.

Moreover, researchers have investigated mutation-based testing for CPS Simulink dataflow models, focusing on detecting seeded faults [BHM+09, HNT14]. As outlined in Chapter 6, we introduced Property-Based Mutation Testing [BMNY23] to address mutation testing for CPS Simulink models with respect to formal properties. In our experiments, we utilized this approach to showcase the mutant-killing effectiveness of the test suites generated by our feature-coverage testing methods.

7.5 Conclusion

We introduced signal feature coverage as a new coverage metric specifically designed for CPS dataflow models, based on commonly observed signal features. Additionally, we developed a test generation strategy aimed at maximizing signal feature coverage for a specified set of features and signals. Our feature-coverage testing approach was evaluated against four state-of-the-art methods, demonstrating superior fault detection by enhancing the coverage of different features of the internal signal within a system model. Our feature-coverage testing method consistently outperformed existing testing strategies across all properties and subjects in our experiments.

The experimental results yielded the following important insights: (1) Utilizing signal features leads to coverage criteria that more accurately reflect the behavior and intricacies of CPS models. (2) Test suites designed to maximize signal feature coverage achieve higher mutation scores.

Data Availability Statement. Our data package and all the details of our evaluation results are available at https://gitlab.com/DrishtiYadav/fct_master.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

CHAPTER 8

Summary and Future Work

In this chapter, we encapsulate the essence of the thesis and illuminate potential paths for future research. First, by distilling the core themes and findings, we offer a brief overview of the thesis contributions in Section 8.1. Next, we identify unexplored areas and propose avenues (in Section 8.2) for further investigation, paving the way for continued scholarly inquiry and advancement in the field.

8.1 Summary

The primary contribution of this thesis, as indicated by its title, lies in the development of a fault-based testing framework tailored for safety-critical CPSs, facilitating swift and accurate fault diagnostics for such systems. The proposed framework integrates several techniques, including fault injection, global optimization, search-based testing for fault localization, mutation testing against formal properties, and feature coverage-based testing. This amalgamation offers distinct advantages over existing approaches and facilitates comprehensive testing across various scenarios. The focus of this thesis was specifically on CPS Simulink models governed by STL specifications. These specifications enable the encoding of intricate task requirements, encompassing both spatial and temporal constraints within the system.

Chapter 3 introduced FIM, a toolkit designed for automating fault injection and mutant generation tasks within Simulink models. With FIM, testers have the flexibility to control the activation of fault blocks according to their testing objectives, and fine-tune fault parameters to rigorously evaluate the system under test against failures, assessing its fault tolerance. Verification experts and testers can adapt and customize our existing open-source toolkit to suit their safety evaluation and testing requirements. The fault injection tool FIM developed in this chapter forms the foundation for fault seeding and mutant generation in the later chapters.

Chapter 4 introduced a novel optimization algorithm termed *Blood Coagulation Algorithm* (BCA), inspired by the mechanism of blood coagulation in the human body. Drawing from the cooperative behavior of thrombocytes and their intelligent clot formation strategy, BCA emphasizes intensification and diversification within a search space. Through extensive testing on a benchmark of well-known functions and real-world optimization problems, we compared BCA with several state-of-the-art meta-heuristic algorithms, revealing its competitive and statistically significant performance. The BCA optimizer introduced in this chapter plays a crucial role in the subsequent chapters, handling optimization tasks efficiently.

Chapter 5 introduced a new technique for fault localization in Simulink models of safety-critical CPS. Our method, guided by an STL property, utilizes both failing executions and automatically generated passing executions resembling the failed ones to pinpoint anomalous signals and, consequently, the probable faulty blocks. We also applied our technique using equivalence testing, where fault localization aligns with implicit specifications. We then showcased the efficacy of our approach, demonstrating that our method effectively localizes multiple faults within an acceptable cost, surpassing the performance of `CPSDebug`, a commonly used approach in the field.

Chapter 6 introduced *Property-Based Mutation Testing* (PBMT), an innovative approach aimed at evaluating test suites based on software properties. We formalized the concept of mutant killability concerning the satisfaction and violation of a property by the original program and its mutated version, respectively. This formalization provided a robust foundation for PBMT and its associated mutant killing problem, facilitating the use of the BCA optimizer for test case generation. We employed various test generation strategies to create test suites and examined their effectiveness in mutant killability. Our investigation of PBMT utilized two Simulink models within the safety-critical CPS domain, highlighting the challenges and relevance of testing software against properties compared to traditional MT method. Additionally, our evaluation revealed that current state-of-the-art techniques like Adaptive Random Testing and Falsification Testing still have limitations in generating test suites capable of effectively killing mutants when tested against properties.

Chapter 7 detailed the introduction of *Signal Feature Coverage*, a coverage metric specifically designed for CPS dataflow models, focusing on common signal features. A search-based test generation strategy was also developed to maximize this coverage for a given set of features and signals. When compared to four leading test generation techniques, our proposed search-based strategy was shown to maximize signal feature coverage more effectively, resulting in superior mutant detection. This highlights the advantage of utilizing signal features to establish a coverage criterion that is better suited to capturing the behavior and intricacies of CPS models. Specifically, we demonstrated that encompassing a broad spectrum of signal features can effectively address various aspects of system functionality, thereby increasing the probability of detecting faults caused by mutations.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

8.2 Outlook and Future Research Directions

The methodologies developed in this thesis, though focused on CPS Simulink models governed by STL specifications, are applicable to other formal specification languages for signal-based temporal properties. The fault localization method in Chapter 5, which compares passing and failing test cases, can be used with any formal language that has well-defined qualitative and quantitative semantics. Similarly, PBMT (evaluating property satisfaction on the original program and their violation on mutants) and FCT-S (identifying features impacting properties) are language-agnostic, given the availability of these semantics. Fault injection is inherently language-independent. Additionally, the Blood Coagulation Algorithm (BCA) is not tied to any specific formal language, though its use in search-based tasks may involve formal specifications. Thus, with well-defined semantics, our proposed methods remain broadly applicable and effective.

This thesis does not comprehensively explore every possible aspect of the presented work. While this dissertation includes experimental validation, it is equally important to assess the practical effectiveness of the proposed tools and methodologies in real-world CPS development environments. To this end, we plan to conduct a user study with professional engineers actively working on CPSs to evaluate the tools' usability, efficiency, and overall impact during fault diagnosis tasks. By simulating realistic scenarios, we aim to gather feedback on workflow integration, identify areas for improvement, and refine our approach. If access to professional engineers is limited, an initial study with students specializing in CPS development will be considered for preliminary insights.

There are several promising avenues for future research that have yet to be explored in this thesis or within the research community. At a broader level, a potential avenue for future research involves exploring higher-order mutants within formal temporal logic specifications, along with investigating falsification and coverage-based testing in multi-fault CPS dataflow models. Developing more systematic tools that can offer formal safety assurances when multiple faults are introduced into the system is crucial, especially in scenarios involving complex CPS controllers with learning-based control laws or learning-enabled perception, which inherently introduce stochastic elements into the system.

Besides, on a more general level, a potential avenue for future research involves testing with Simulink models that contain documented *real* faults, although this task is notably challenging due to the scarcity of such models. Furthermore, future investigations could involve testing our proposed tools and techniques across a broader range of Simulink models and a larger set of STL properties. This approach aims to address potential validity threats by enhancing the generalizability of the results obtained.

More with respect to the themes explored in this thesis, we consider FIM (as discussed in Chapter 3) to be a significant advancement in the verification process for safety-critical systems. Presently, FIM efficiently generates a considerable number of fault-injected models, enabling scalable fault injection experiments. Future improvements may target specific application scenarios, like offering more refined testing options. Conducting

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

an empirical study to evaluate the costs and benefits of large-scale mutation testing in Simulink models could provide valuable insights. Moreover, expanding the tool to include additional options for fault events could enhance fault diagnosis and reasoning capabilities. Lastly, integrating FIM with simulation-based verification engines employing formal approaches for safety analysis holds promise for further enhancing its utility.

Another promising avenue for future research involves extending BCA to address multi-objective and many-objective optimization problems. Additionally, investigating the performance of BCA with various constraint handling approaches for solving constrained optimization problems is another promising area for research. The capability of BCA to deal with highly constrained problems also needs to be explored. Additionally, there is a need for continued research to evaluate the effectiveness of the proposed algorithm in addressing complex real-life applications characterized by complex search spaces and high dimensionality across diverse multidisciplinary domains. Given the simplicity of the BCA proposed in this thesis in Chapter 4, incorporating additional mechanisms such as evolutionary updating structures and chaos-based updating strategies could be beneficial for future research efforts.

Expanding upon the other subjects addressed in this thesis, there is potential for expanding upon our fault localization methodology, as outlined in Chapter 5. This expansion could involve integrating different test generation strategies and evaluating how they influence the accuracy of fault localization as a whole. Additionally, delving into the interaction among various failing executions within a system that violates a given specification presents an intriguing avenue for exploration. Also, developing nuanced strategies to select the most effective test cases from a test suite based on these interactions could be a valuable endeavor.

Another prospective research trajectory involves adaptation of PBMT to closely associated CPS modeling languages, encompassing Simulink models integrated with Stateflow Charts. Additionally, there will be a need for further exploration into HOMs, along with research endeavors aimed at addressing test case generation for exposing higher-order faults. Furthermore, complex LUTs (including n-D lookup tables and Prelookup blocks) are also used in model-based development of hybrid systems with Simulink. Mutations occurring within these LUTs could present greater interest to testers, thus representing a potential avenue for future research.

Another promising future research avenue involves expanding the study of signal feature coverage to HOMs and assessing its applicability in more complex scenarios. Furthermore, refining the testing approach by exploring the interactions of different signal features on system specifications could enhance its effectiveness. These initiatives are set to expand the knowledge in CPS testing, ultimately leading to the creation of more resilient and dependable CPSs in the future.

To conclude, we hope that certain sections of this thesis will serve as a foundation for further exploration and development, whether it is through expanding upon the points outlined in the future prospects or applying the concepts in different domains.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

List of Figures

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

List of Tables

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

List of Algorithms

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Bibliography

[5G 24] Austria 5G PLAYGROUND CARINTHIA. 5g playground - what is the playground?, 2024.

[AAJ+14] Bernhard K Aichernig, Jakob Auer, Elisabeth Jöbstl, Robert Korošec, Willibald Krenn, Rupert Schlick, and Birgit Vera Schmidt. Model-based mutation testing of an industrial measurement device. In *International Conference on Tests and Proofs*, pages 1–19. Springer, 2014.

[ABD+79] Allen Troy Acree, Timothy Alan Budd, Richard A. DeMillo, Richard J. Lipton, and Frederick Gerald Sayward. Mutation analysis. techreport GIT-ICS-79/08, Georgia Institute of Technology, Atlanta, Georgia, 1979.

[ABJ+15] Bernhard K Aichernig, Harald Brandl, Elisabeth Jöbstl, Willibald Krenn, Rupert Schlick, and Stefan Tiran. Killing strategies for model-based mutation testing. *Software Testing, Verification and Reliability*, 25(8):716–748, 2015.

[ABJK11] Bernhard K Aichernig, Harald Brandl, Elisabeth Jöbstl, and Willibald Krenn. Efficient mutation killers in action. In *2011 Fourth IEEE International Conference on Software Testing, Verification and Validation*, pages 120–129. IEEE, 2011.

[ABL15] J. H. Andrews, L. C. Briand, and Y. Labiche. Is mutation an appropriate tool for testing experiments? In *Proceedings of the International Conference on Software Engineering (ICSE)*, 2015.

[ABLN06] J. H. Andrews, L. C. Briand, Y. Labiche, and A. S. Namin. Using mutation analysis for assessing and comparing testing coverage criteria. *IEEE Transactions on Software Engineering*, 32(8):608–624, 2006.

[ADD+17] Arvind S. Adimoolam, Thao Dang, Alexandre Donzé, James Kapinski, and Xiaoqing Jin. Classification and coverage-based falsification for embedded control systems. In Rupak Majumdar and Viktor Kuncak,

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

editors, *Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I*, volume 10426 of *Lecture Notes in Computer Science*, pages 483–503. Springer, 2017.

[ADM$^+$21] Laith Abualigah, Ali Diabat, Seyedali Mirjalili, Mohamed Abd Elaziz, and Amir H Gandomi. The arithmetic optimization algorithm. *Computer methods in applied mechanics and engineering*, 376:113609, 2021.

[AERP04] Vadim Alyokhin, Benedikte Elbel, Martin Rothfelder, and Alexander Pretschner. Coverage metrics for continuous function charts. In *15th International Symposium on Software Reliability Engineering (ISSRE 2004), 2-5 November 2004, Saint-Malo, Bretagne, France*, pages 257–268, USA, 2004. IEEE Computer Society.

[AF12] Houssam Abbas and Georgios Fainekos. Convergence proofs for simulated annealing falsification of safety properties. In *50th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2012, Allerton Park & Retreat Center, Monticello, IL, USA, October 1-5, 2012*, pages 1594–1601. IEEE, 2012.

[AFI17] Hammoudi Abderazek, Djeddou Ferhat, and Atanasovska Ivana. Adaptive mixed differential evolution algorithm for bi-objective tooth profile spur gear optimization. *The International Journal of Advanced Manufacturing Technology*, 90:2063–2073, 2017.

[AGL07] Esmaeil Atashpaz-Gargari and Caro Lucas. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In *2007 IEEE congress on evolutionary computation*, pages 4661–4667. Ieee, 2007.

[AH15] Takumi Akazaki and Ichiro Hasuo. Time robustness in MTL and expressivity in hybrid system falsification. In Daniel Kroening and Corina S. Pasareanu, editors, *Computer Aided Verification - 27th International Conference, CAV 2015, San Francisco, CA, USA, July 18-24, 2015, Proceedings, Part II*, volume 9207 of *Lecture Notes in Computer Science*, pages 356–374. Springer, 2015.

[AK12] Bahriye Akay and Dervis Karaboga. Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of intelligent manufacturing*, 23:1001–1014, 2012.

[ALFS11] Yashwanth Annpureddy, Che Liu, Georgios Fainekos, and Sriram Sankaranarayanan. S-taliro: A tool for temporal logic falsification for hybrid systems. In *International Conference on Tools and Algorithms*

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

for the Construction and Analysis of Systems, pages 254–257. Springer, 2011.

[ALN13] Bernhard K Aichernig, Florian Lorber, and Dejan Ničković. Time for mutants—model-based mutation testing with timed automata. In *International Conference on Tests and Proofs*, pages 20–38. Springer, 2013.

[ALRL04] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl E. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33, 2004.

[ARM17] Arend Aerts, Michel Reniers, and Mohammad Reza Mousavi. Model-based testing of cyber-physical systems. In *Cyber-Physical Systems*, pages 287–304. Elsevier, 2017.

[Aro04] Jasbir Singh Arora. *Introduction to optimum design*. Elsevier, 2004.

[AS19] Sankalap Arora and Satvir Singh. Butterfly optimization algorithm: a novel approach for global optimization. *Soft computing*, 23:715–734, 2019.

[AWM+18] Aitor Arrieta, Shuai Wang, Urtzi Markiegi, Goiuria Sagardui, and Leire Etxeberria. Employing multi-objective search to enhance reactive test case generation and prioritization for testing industrial cyber-physical systems. *IEEE Trans. Ind. Informatics*, 14(3):1055–1066, 2018.

[AYS20] Qamar Askari, Irfan Younas, and Mehreen Saeed. Political optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowledge-based systems*, 195:105709, 2020.

[B+12] Nguyen Thanh Binh et al. Mutation operators for simulink models. In *2012 Fourth International Conference on Knowledge and Systems Engineering*, pages 54–59. IEEE, 2012.

[BA15] Adil Baykasoğlu and Şener Akpinar. Weighted superposition attraction (wsa): A swarm intelligence algorithm for optimization problems–part 2: Constrained optimization. *Applied Soft Computing*, 37:396–415, 2015.

[Bac91] Thomas Back. A survey of evolution strategies. In *Proc. of Fourth Internal. Conf. on Genetic Algorithms*, 1991.

[BBC+16] Benjamin Bittner, Marco Bozzano, Roberto Cavada, Alessandro Cimatti, Marco Gario, Alberto Griggio, Cristian Mattarei, Andrea

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Micheli, and Gianni Zampedri. The xsap safety analysis platform. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 533–539. Springer, 2016.

[BBL$^+$20] Ezio Bartocci, Luca Bortolussi, Michele Loreti, Laura Nenzi, and Simone Silvetti. Moonlight: A lightweight tool for monitoring spatio-temporal properties. In *International Conference on Runtime Verification*, pages 417–428. Springer, 2020.

[BBM$^+$22] Ezio Bartocci, Roderick Bloem, Benedikt Maderbacher, Niveditha Manjunath, and Dejan Ničković. Adaptive testing for specification coverage and refinement in cps models. *Nonlinear Analysis: Hybrid Systems*, 46:101254, 2022.

[BDT99] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence: from natural to artificial systems*. Oxford university press, 1999.

[BFMN18] Ezio Bartocci, Thomas Ferrère, Niveditha Manjunath, and Dejan Ničković. Localizing faults in simulink/stateflow models with STL. In *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week)*, pages 197–206, 2018.

[BG20] Seyede Marziyeh Ghoreshi Beyrami and Peyvand Ghaderyan. A robust, cost-effective and non-invasive computer-aided method for diagnosis three types of neurodegenerative diseases with gait signal analysis. *Measurement*, 156:107579, 2020.

[BH12] Richard Baker and Ibrahim Habli. An empirical evaluation of mutation testing for improving the test quality of safety-critical software. *IEEE Transactions on Software Engineering*, 39(6):787–805, 2012.

[BHM$^+$09] Angelo Brillout, Nannan He, Michele Mazzucchi, Daniel Kroening, Mitra Purandare, Philipp Rümmer, and Georg Weissenbacher. Mutation-based test case generation for simulink models. In Frank S. de Boer, Marcello M. Bonsangue, Stefan Hallerstede, and Michael Leuschel, editors, *Formal Methods for Components and Objects - 8th International Symposium, FMCO 2009, Eindhoven, The Netherlands, November 4-6, 2009. Revised Selected Papers*, volume 6286 of *Lecture Notes in Computer Science*, pages 208–227, Berlin, Heidelberg, 2009. Springer.

[BK11] Federico Bergero and Ernesto Kofman. Powerdevs: a tool for hybrid system modeling and real-time simulation. *Simul.*, 87(1-2):113–132, 2011.

[BLS13] Ilhem Boussaïd, Julien Lepagnot, and Patrick Siarry. A survey on optimization metaheuristics. *Information sciences*, 237:82–117, 2013.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[Blu05] Christian Blum. Ant colony optimization: Introduction and recent trends. *Physics of Life reviews*, 2(4):353–373, 2005.

[BMM+19] Ezio Bartocci, Niveditha Manjunath, Leonardo Mariani, Cristinel Mateis, and Dejan Ničković. Automatic failure explanation in CPS models. In *International Conference on Software Engineering and Formal Methods*, pages 69–86. Springer, 2019.

[BMM+21] Ezio Bartocci, Niveditha Manjunath, Leonardo Mariani, Cristinel Mateis, and Dejan Ničković. CPSdebug: Automatic failure explanation in CPS models. *International Journal on Software Tools for Technology Transfer*, 23(5):783–796, 2021.

[BMNY22a] Ezio Bartocci, Leonardo Mariani, Dejan Nickovic, and Drishti Yadav. FIM: fault injection and mutation for simulink. In Abhik Roychoudhury, Cristian Cadar, and Miryung Kim, editors, *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2022, Singapore, Singapore, November 14-18, 2022*, pages 1716–1720, USA, 2022. ACM.

[BMNY22b] Ezio Bartocci, Leonardo Mariani, Dejan Nickovic, and Drishti Yadav. FIM: Fault Injection and Mutation for Simulink, May 2022.

[BMNY22c] Ezio Bartocci, Leonardo Mariani, Dejan Nickovic, and Drishti Yadav. Search-based testing for accurate fault localization in CPS. In *IEEE 33rd International Symposium on Software Reliability Engineering, ISSRE 2022, Charlotte, NC, USA, October 31 - Nov. 3, 2022*, pages 145–156, USA, 2022. IEEE.

[BMNY23] Ezio Bartocci, Leonardo Mariani, Dejan Nickovic, and Drishti Yadav. Property-based mutation testing. In *IEEE Conference on Software Testing, Verification and Validation, ICST 2023, Dublin, Ireland, April 16-20, 2023*, pages 222–233, USA, 2023. IEEE.

[BO15] Adil Baykasoğlu and Fehmi Burcin Ozsoydan. Adaptive firefly algorithm with chaos for mechanical design optimization problems. *Applied soft computing*, 36:152–164, 2015.

[Boa92a] Boualem Boashash. Estimating and interpreting the instantaneous frequency of a signal. i. fundamentals. *Proceedings of the IEEE*, 80(4):520–538, 1992.

[Boa92b] Boualem Boashash. Estimating and interpreting the instantaneous frequency of a signal. ii. algorithms and applications. *Proceedings of the IEEE*, 80(4):540–568, 1992.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[BSAH21] Malik Braik, Alaa Sheta, and Heba Al-Hiary. A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm. *Neural computing and applications*, 33(7):2515–2547, 2021.

[BV07] Marco Bozzano and Adolfo Villafiorita. The fsap/nusmv-sa safety analysis platform. *Int. J. Softw. Tools Technol. Transf.*, 9(1):5–24, 2007.

[CÁS13] Xin Chen, Erika Ábrahám, and Sriram Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *International Conference on Computer Aided Verification*, pages 258–263. Springer, 2013.

[CCB04] Carlos A Coello Coello and Ricardo Landa Becerra. Efficient evolutionary optimization through the use of a cultural algorithm. *Engineering Optimization*, 36(2):219–236, 2004.

[CEC08] Leticia C Cagnina, Susana C Esquivel, and Carlos A Coello Coello. Solving engineering optimization problems with the simple constrained particle swarm optimizer. *Informatica*, 32(3), 2008.

[Cha14] S. Chamarthy. Normal coagulation and hemostasis. In Linda M. McManus and Richard N. Mitchell, editors, *Pathobiology of Human Disease*, pages 1544–1552. Academic Press, San Diego, 2014.

[CKLW08] Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, and W. Eric Wong. Does adaptive random testing deliver a higher confidence than random testing? In Hong Zhu, editor, *Proceedings of the Eighth International Conference on Quality Software, QSIC 2008, 12-13 August 2008, Oxford, UK*, pages 145–154, USA, 2008. IEEE Computer Society.

[CLM05] Tsong Yueh Chen, Hing Leung, and Ieng Kei Mak. Adaptive random testing. In *Advances in Computer Science-ASIAN 2004. Higher-Level Decision Making: 9th Asian Computing Science Conference. Dedicated to Jean-Louis Lassez on the Occasion of His 5th Birthday. Chiang Mai, Thailand, December 8-10, 2004. Proceedings 9*, pages 320–329, Berlin, Heidelberg, 2005. Springer, Springer.

[ČLM13] Matej Črepinšek, Shih-Hsi Liu, and Marjan Mernik. Exploration and exploitation in evolutionary algorithms: A survey. *ACM computing surveys (CSUR)*, 45(3):1–33, 2013.

[CM02] Carlos A Coello Coello and Efrén Mezura Montes. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*, 16(3):193–203, 2002.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

[CMM+18] Shafiul Azam Chowdhury, Soumik Mohian, Sidharth Mehra, Siddhant Gawsane, Taylor T Johnson, and Christoph Csallner. Automatically finding bugs in a commercial cyber-physical system development tool chain with slforge. In *Proceedings of the 40th International Conference on Software Engineering*, pages 981–992, 2018.

[Coe00] Carlos A Coello Coello. Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, 41(2):113–127, 2000.

[Coe02] Carlos A Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering*, 191(11-12):1245–1287, 2002.

[CW97] YJ Cao and QH Wu. Evolutionary programming. In *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, pages 443–446. IEEE, 1997.

[CZE17] Jacek M Czerniak, Hubert Zarzycki, and Dawid Ewald. Aao as a new strategy in modeling and simulation of constructional problems optimization. *Simulation Modelling Practice and Theory*, 76:22–33, 2017.

[DA94] René David and Hassane Alla. Petri nets for modeling of dynamic systems: A survey. *Autom.*, 30(2):175–202, 1994.

[DDD+15] Tommaso Dreossi, Thao Dang, Alexandre Donzé, James Kapinski, Xiaoqing Jin, and Jyotirmoy V. Deshmukh. Efficient guiding strategies for testing of temporal properties of hybrid systems. In Klaus Havelund, Gerard J. Holzmann, and Rajeev Joshi, editors, *NASA Formal Methods - 7th International Symposium, NFM 2015, Pasadena, CA, USA, April 27-29, 2015, Proceedings*, volume 9058 of *Lecture Notes in Computer Science*, pages 127–142, Cham, 2015. Springer.

[Deb91] Kalyanmoy Deb. Optimal design of a welded beam via genetic algorithms. *AIAA journal*, 29(11):2013–2015, 1991.

[DG+96] Kalyanmoy Deb, Mayank Goyal, et al. A combined genetic adaptive search (geneas) for engineering design. *Computer Science and informatics*, 26:30–45, 1996.

[DGMH11] Joaquín Derrac, Salvador García, Daniel Molina, and Francisco Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18, 2011.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[DGV96] Akash Deshpande, Aleks Göllü, and Pravin Varaiya. SHIFT: A formalism and a programming language for dynamic networks of hybrid automata. In Panos J. Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems IV, Proceedings of the Fourth International Workshop on Hybrid Systems, Ithaca, NY, USA, October 1996*, volume 1273 of *Lecture Notes in Computer Science*, pages 113–133. Springer, 1996.

[Dhi21] Gaurav Dhiman. Esa: a hybrid bio-inspired metaheuristic optimization approach for engineering problems. *Engineering with Computers*, 37:323–353, 2021.

[DJKM15] Jyotirmoy V. Deshmukh, Xiaoqing Jin, James Kapinski, and Oded Maler. Stochastic local search for falsification of hybrid systems. In Bernd Finkbeiner, Geguang Pu, and Lijun Zhang, editors, *Automated Technology for Verification and Analysis - 13th International Symposium, ATVA 2015, Shanghai, China, October 12-15, 2015, Proceedings*, volume 9364 of *Lecture Notes in Computer Science*, pages 500–517. Springer, 2015.

[DLS78] Richard A. DeMillo, Richard J. Lipton, and Frederick Gerald Sayward. Hints on test data selection: Help for the practicing programmer. *Computer*, 11(4):34–41, April 1978.

[DM10] Alexandre Donzé and Oded Maler. Robust satisfaction of temporal logic over real-valued signals. In Krishnendu Chatterjee and Thomas A. Henzinger, editors, *Formal Modeling and Analysis of Timed Systems - 8th International Conference, FORMATS 2010, Klosterneuburg, Austria, September 8-10, 2010. Proceedings*, volume 6246 of *Lecture Notes in Computer Science*, pages 92–106. Springer, 2010.

[DMVP15] Parasara Sridhar Duggirala, Sayan Mitra, Mahesh Viswanathan, and Matthew Potok. C2e2: A verification tool for stateflow models. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 68–82. Springer, 2015.

[Don10] Alexandre Donzé. Breach, a toolbox for verification and parameter synthesis of hybrid systems. In *International Conference on Computer Aided Verification*, pages 167–170. Springer, 2010.

[DPHG$^+$18] Pedro Delgado-Pérez, Ibrahim Habli, Steve Gregory, Rob Alexander, John Clark, and Inmaculada Medina-Bulo. Evaluation of mutation testing in a nuclear industry case study. *IEEE Transactions on Reliability*, 67(4):1406–1419, 2018.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

[DPV+20] Serge Demeyer, Ali Parsai, Sten Vercammen, Brent van Bladel, and Mehrdad Abdi. Formal verification of developer tests: a research agenda inspired by mutation testing. In *International Symposium on Leveraging Applications of Formal Methods*, pages 9–24. Springer, 2020.

[DR64] Earl W. Davie and Oscar D. Ratnoff. Waterfall sequence for intrinsic blood clotting. *Science*, 145(3638):1310–1312, 1964.

[dSC10] Leandro dos Santos Coelho. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Systems with Applications*, 37(2):1676–1683, 2010.

[DST17] Ram Das Diwakaran, Sriram Sankaranarayanan, and Ashutosh Trivedi. Analyzing neighborhoods of falsifying traces in cyber-physical systems. In *Proceedings of the 8th International Conference on Cyber-Physical Systems*, pages 109–119, 2017.

[DYHF17] Adel Dokhanchi, Shakiba Yaghoubi, Bardh Hoxha, and Georgios Fainekos. Vacuity aware falsification for MTL request-response specifications. In *13th IEEE Conference on Automation Science and Engineering, CASE 2017, Xi'an, China, August 20-23, 2017*, pages 1332–1337, USA, 2017. IEEE.

[DZS+15] Adel Dokhanchi, Aditya Zutshi, Rahul T. Sriniva, Sriram Sankaranarayanan, and Georgios Fainekos. Requirements driven falsification with coverage metrics. In Alain Girault and Nan Guan, editors, *2015 International Conference on Embedded Software, EMSOFT 2015, Amsterdam, Netherlands, October 4-9, 2015*, pages 31–40, USA, 2015. IEEE.

[EAF+22] Gidon Ernst, Paolo Arcaini, Georgios Fainekos, Federico Formica, Jun Inoue, Tanmay Khandait, Mohammad Mahdi Mahboob, Claudio Menghi, Giulia Pedrielli, Masaki Waga, et al. Arch-comp 2022 category report: Falsification with ubounded resources. *EPiC Series in Computing*, 90:204–221, 2022.

[ECGN01] Michael D. Ernst, Jake Cockrell, William G. Griswold, and David Notkin. Dynamically discovering likely program invariants to support program evolution. *IEEE Transactions on Software Engineering*, 27(2):99–123, 2001.

[EE06] Osman K Erol and Ibrahim Eksin. A new optimization method: big bang–big crunch. *Advances in engineering software*, 37(2):106–111, 2006.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[EL17] Jonathan Ezekiel and Andrea Lomuscio. Combining fault injection and model checking to verify fault tolerance, recoverability, and diagnosability in multi-agent systems. *Inf. Comput.*, 254:167–194, 2017.

[ESBH12] Hadi Eskandar, Ali Sadollah, Ardeshir Bahreininejad, and Mohd Hamdi. Water cycle algorithm–a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, 110:151–166, 2012.

[FA11] Gordon Fraser and Andrea Arcuri. Evosuite: automatic test suite generation for object-oriented software. In Tibor Gyimóthy and Andreas Zeller, editors, *SIGSOFT/FSE'11 19th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-19) and ESEC'11: 13th European Software Engineering Conference (ESEC-13), Szeged, Hungary, September 5-9, 2011*, pages 416–419, USA, 2011. ACM.

[FCA22] Daniel Fortunato, Jose Campos, and Rui Abreu. Mutation testing of quantum programs: A case study with qiskit. *IEEE Transactions on Quantum Engineering*, 3:1–17, 2022.

[FFC91] Jing-Fan Fu, Robert G Fenton, and William L Cleghorn. A mixed integer-discrete-continuous programming method and its application to engineering design optimization. *Engineering Optimization*, 17(4):263–280, 1991.

[FGD$^+$11] Goran Frehse, Colas Le Guernic, Alexandre Donzé, Scott Cotton, Rajarshi Ray, Olivier Lebeltel, Rodolfo Ripado, Antoine Girard, Thao Dang, and Oded Maler. SpaceEx: Scalable verification of hybrid systems. In *International Conference on Computer Aided Verification*, pages 379–395. Springer, 2011.

[FKS$^+$19] Andreas Fellner, Willibald Krenn, Rupert Schlick, Thorsten Tarrach, and Georg Weissenbacher. Model-based, mutation-driven test-case generation via heuristic-guided branching search. *ACM Transactions on Embedded Computing Systems (TECS)*, 18(1):1–28, 2019.

[FM24] Republic of Austria Federal Ministry. Austrian action programme on automated mobility 2019–2022, 2024.

[FMMJ21] Tagir Fabarisov, Ilshat Mamaev, Andrey Morozov, and Klaus Janschek. Model-based fault injection experiments for the safety analysis of exoskeleton system. *arXiv preprint arXiv:2101.01283*, 2021.

[FMN15] Thomas Ferrère, Oded Maler, and Dejan Nickovic. Trace diagnostics using temporal implicants. In Bernd Finkbeiner, Geguang Pu, and

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Lijun Zhang, editors, *Automated Technology for Verification and Analysis - 13th International Symposium, ATVA 2015, Shanghai, China, October 12-15, 2015, Proceedings*, volume 9364 of *Lecture Notes in Computer Science*, pages 241–258. Springer, 2015.

[FROC$^+$20] Fernando Fausto, Adolfo Reyna-Orta, Erik Cuevas, Ángel G Andrade, and Marco Perez-Cisneros. From ants to whales: metaheuristics for all tastes. *Artificial Intelligence Review*, 53:753–810, 2020.

[FSDC10] Cláudia Natália Ferreira, Marinez de Oliveira Sousa, Luci Maria Sant'Ana Dusse, and Maria das Graças Carvalho. A cell-based model of coagulation and its implications. *Revista brasileira de Hematologia e Hemoterapia*, 32:416–421, 2010.

[GA12] Amir Hossein Gandomi and Amir Hossein Alavi. Krill herd: a new bio-inspired optimization algorithm. *Communications in nonlinear science and numerical simulation*, 17(12):4831–4845, 2012.

[Gan14] Amir H Gandomi. Interior search algorithm (isa): a novel approach for global optimization. *ISA transactions*, 53(4):1168–1183, 2014.

[Gar14] Harish Garg. Solving structural engineering design optimization problems using an artificial bee colony algorithm. *J Ind Manag Optim*, 10(3):777–794, 2014.

[GB14] Naser Ghorbani and Ebrahim Babaei. Exchange market algorithm. *Applied soft computing*, 19:177–187, 2014.

[GBB20] Cyrus Garmo, Tanvir Bajwa, and Bracken Burns. Physiology, clotting mechanism. Online, 2020.

[GC14] Jean Dickinson Gibbons and Subhabrata Chakraborti. *Nonparametric statistical inference*. CRC press, 2014.

[GHJ07] Alex Groce, Gerard Holzmann, and Rajeev Joshi. Randomized differential testing as a prelude to formal verification. In *29th International Conference on Software Engineering (ICSE'07)*, pages 621–631. IEEE, 2007.

[GJG17] Rahul Gopinath, Carlos Jensen, and Alex Groce. The theory of composite faults. In *2017 IEEE International Conference on Software Testing, Verification and Validation (ICST)*, pages 47–57, 2017.

[GKL01] Zong Woo Geem, Joong Hoon Kim, and Gobichettipalayam Vasudevan Loganathan. A new heuristic optimization algorithm: harmony search. *simulation*, 76(2):60–68, 2001.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

[GM05] Jason Ghidella and Pieter Mosterman. Requirements-based testing in aircraft control design. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, page 5886, 2005.

[GRS+16] Gregory Gay, Ajitha Rajan, Matt Staats, Michael W. Whalen, and Mats Per Erik Heimdahl. The effect of program and model structure on the effectiveness of MC/DC test adequacy coverage. *ACM Trans. Softw. Eng. Methodol.*, 25(3):25:1–25:34, 2016.

[Gue16] Najeh Ben Guedria. Improved accelerated pso algorithm for mechanical engineering optimization problems. *Applied Soft Computing*, 40:455–467, 2016.

[GYA11] Amir Hossein Gandomi, Xin-She Yang, and Amir Hossein Alavi. Mixed variable structural optimization using firefly algorithm. *Computers & Structures*, 89(23-24):2325–2336, 2011.

[GYA13] Amir Hossein Gandomi, Xin-She Yang, and Amir Hossein Alavi. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*, 29:17–35, 2013.

[GYYT13] Amir Hossein Gandomi, Gun Jin Yun, Xin-She Yang, and Siamak Talatahari. Chaos-enhanced accelerated particle swarm optimization. *Communications in Nonlinear Science and Numerical Simulation*, 18(2):327–340, 2013.

[HAF14] Bardh Hoxha, Houssam Abbas, and Georgios Fainekos. Benchmarks for temporal logic requirements for automotive systems. In Goran Frehse and Matthias Althoff, editors, *1st and 2nd International Workshop on Applied veRification for Continuous and Hybrid Systems, ARCH@CPSWeek 2014, Berlin, Germany, April 14, 2014 / ARCH@CPSWeek 2015, Seattle, WA, USA, April 13, 2015*, volume 34 of *EPiC Series in Computing*, pages 25–30, USA, 2014. EasyChair.

[HAL+18] Ferhat Hamza, Hammoudi Abderazek, Smata Lakhdar, Djeddou Ferhat, and Ali Rıza Yıldız. Optimum design of cam-roller follower mechanism using a new evolutionary algorithm. *The International Journal of Advanced Manufacturing Technology*, 99:1267–1282, 2018.

[Hat13] Abdolreza Hatamlou. Black hole: A new heuristic optimization approach for data clustering. *Information sciences*, 222:175–184, 2013.

[Hev07] Alan R Hevner. A three cycle view of design science research. *Scandinavian journal of information systems*, 19(2):4, 2007.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[HF06] Abdel-Rahman Hedar and Masao Fukushima. Derivative-free filter simulated annealing method for constrained continuous global optimization. *Journal of Global optimization*, 35:521–549, 2006.

[HMF+19] Ali Asghar Heidari, Seyedali Mirjalili, Hossam Faris, Ibrahim Aljarah, Majdi Mafarja, and Huiling Chen. Harris hawks optimization: Algorithm and applications. *Future generation computer systems*, 97:849–872, 2019.

[HMI01] Maureane Hoffman and Dougald M Monroe III. A cell-based model of hemostasis. *Thrombosis and haemostasis*, 85(06):958–965, 2001.

[HNT14] Le Thi My Hanh, Thanh Binh Nguyen, and Khuat Thanh Tung. Applying the meta-heuristic algorithms for mutation-based test data generation for simulink models. In Nguyen Trong Giang, Huynh Quyet Thang, Ismal Khalil, Son Hong Ngo, Yves Deville, and Marc Bui, editors, *Proceedings of the Fifth Symposium on Information and Communication Technology, SoICT '14, Hanoi, Vietnam, December 4-5, 2014*, pages 102–109, USA, 2014. ACM.

[Hol92] John H Holland. Genetic algorithms. *Scientific american*, 267(1):66–73, 1992.

[HPW04] Shan He, E Prempain, and QH Wu. An improved particle swarm optimizer for mechanical design optimization problems. *Engineering optimization*, 36(5):585–605, 2004.

[HRK11] Nannan He, Philipp Rümmer, and Daniel Kroening. Test-case generation for embedded simulink via formal concept analysis. In *Proceedings of the 48th Design Automation Conference*, pages 224–229, 2011.

[HTI97] Mei-Chen Hsueh, Timothy K Tsai, and Ravishankar K Iyer. Fault injection techniques and tools. *Computer*, 30(4):75–82, 1997.

[HW07a] Qie He and Ling Wang. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering applications of artificial intelligence*, 20(1):89–99, 2007.

[HW07b] Qie He and Ling Wang. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Applied mathematics and computation*, 186(2):1407–1422, 2007.

[HWH07] Fu-zhuo Huang, Ling Wang, and Qie He. An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and computation*, 186(1):340–356, 2007.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[IH14] Laura Inozemtseva and Reid Holmes. Coverage is not strongly correlated with test suite effectiveness. In Pankaj Jalote, Lionel C. Briand, and André van der Hoek, editors, *36th International Conference on Software Engineering, ICSE '14, Hyderabad, India - May 31 - June 07, 2014*, pages 435–445. ACM, 2014.

[IPJF19] Marko Ivankovic, Goran Petrovic, René Just, and Gordon Fraser. Code coverage at google. In Marlon Dumas, Dietmar Pfahl, Sven Apel, and Alessandra Russo, editors, *Proceedings of the ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/SIGSOFT FSE 2019, Tallinn, Estonia, August 26-30, 2019*, pages 955–963, USA, 2019. ACM.

[ISRO24a] Government of India Indian Space Research Organisation, Department of Space. ADITYA-L1 Mission Details, 2024.

[ISRO24b] Government of India Indian Space Research Organisation, Department of Space. Chandrayaan-3 details, 2024.

[JB05] Jolyon Jesty and Edward Beltrami. Positive feedbacks of coagulation: their role in threshold regulation. *Arteriosclerosis, thrombosis, and vascular biology*, 25(12):2463–2469, 2005.

[JCL11] Jeff C. Jensen, Danica H. Chang, and Edward A. Lee. A model-based design methodology for cyber-physical systems. In *Proceedings of the 7th International Wireless Communications and Mobile Computing Conference, IWCMC 2011, Istanbul, Turkey, 4-8 July, 2011*, pages 1666–1671. IEEE, 2011.

[JDK$^+$14] Xiaoqing Jin, Jyotirmoy V Deshmukh, James Kapinski, Koichi Ueda, and Ken Butts. Powertrain control verification benchmark. In *Proceedings of the 17th international conference on Hybrid systems: computation and control*, pages 253–262, 2014.

[JH09] Yue Jia and Mark Harman. Higher order mutation testing. *Information and Software Technology*, 51(10):1379–1393, 2009.

[JH11] Yue Jia and Mark Harman. An analysis and survey of the development of mutation testing. *IEEE Trans. Software Eng.*, 37(5):649–678, 2011.

[JJI$^+$14] René Just, Darioush Jalali, Laura Inozemtseva, Michael D. Ernst, Reid Holmes, and Gordon Fraser. Are mutants a valid substitute for real faults in software testing? In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, FSE 2014, page 654–665, New York, NY, USA, 2014. Association for Computing Machinery.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[JSKW16] Nishtha Jatana, Bharti Suri, Prateek Kumar, and Bimlesh Wadhwa. Test suite reduction by mutation testing mapped to set cover problem. In *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, pages 1–6, 2016.

[KA18] Mehak Kohli and Sankalap Arora. Chaotic grey wolf optimization algorithm for constrained optimization problems. *Journal of computational design and engineering*, 5(4):458–472, 2018.

[KAC21] Nabeel Ali Khan, Sadiq Ali, and Kwonhue Choi. An instantaneous frequency and group delay based feature for classifying eeg signals. *Biomedical Signal Processing and Control*, 67:102562, 2021.

[KAO15] Bob Kurtz, Paul Ammann, and Jeff Offutt. Static analysis of mutant subsumption. In *2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 1–10. IEEE, 2015.

[KB08] Dervis Karaboga and Bahriye Basturk. On the performance of artificial bee colony (abc) algorithm. *Applied soft computing*, 8(1):687–697, 2008.

[KBB$^+$09] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology*, 51(1):7 – 15, 2009. Special Section - Most Cited Articles in 2002 and Regular Research Papers.

[KBG$^+$21] Meghdad Khazaee, Ahmad Banakar, Barat Ghobadian, Mostafa Agha Mirsalim, and Saeid Minaei. Remaining useful life (RUL) prediction of internal combustion engine timing belt based on vibration signals and artificial neural network. *Neural Comput. Appl.*, 33(13):7785–7801, 2021.

[KDN14] Maha Kooli and Giorgio Di Natale. A survey on simulation-based fault injection tools for complex systems. In *2014 9th IEEE International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, pages 1–6. IEEE, 2014.

[KdSC06] Renato A Krohling and Leandro dos Santos Coelho. Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(6):1407–1416, 2006.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[KE95] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. ieee, 1995.

[KGJV83] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[KISH20] Deok-Jae Kwon, Jun-Hyuk Im, Mudassir Raza Siddiqi, and Jin Hur. Detection technique for manufacturing imperefection of rare-earth magnets on ipmsm. In *2020 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 1407–1410, USA, 2020. IEEE.

[KK94] BK Kannan and Steven N Kramer. An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *J. Mech. Des.*, 1994.

[KKS18] Meeta Kumar, Anand J Kulkarni, and Suresh Chandra Satapathy. Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology. *Future Generation Computer Systems*, 81:252–272, 2018.

[Koz94] John R Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and computing*, 4:87–112, 1994.

[KS16] Willibald Krenn and Rupert Schlick. Mutation-driven test case generation using short-lived concurrent mutants–first results. *arXiv preprint arXiv:1601.06974*, 2016.

[KST$^+$15] Willibald Krenn, Rupert Schlick, Stefan Tiran, Bernhard Aichernig, Elisabeth Jobstl, and Harald Brandl. Momut:: Uml model-based mutation testing for uml. In *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, pages 1–8. IEEE, 2015.

[KT10a] A Kaveh and S Talatahari. An improved ant colony optimization for constrained engineering design problems. *Engineering Computations*, 27(1):155–182, 2010.

[KT10b] Ali Kaveh and Siamak Talatahari. A novel heuristic optimization method: charged system search. *Acta mechanica*, 213(3):267–289, 2010.

[LCW10] Hui Liu, Zixing Cai, and Yong Wang. Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Applied Soft Computing*, 10(2):629–640, 2010.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

[Lee15] Edward A. Lee. The past, present and future of cyber-physical systems: A focus on models. *Sensors*, 15(3):4837–4869, 2015.

[LG04] Kang Seok Lee and Zong Woo Geem. A new structural optimization method based on the harmony search algorithm. *Computers & structures*, 82(9-10):781–798, 2004.

[LG05] Kang Seok Lee and Zong Woo Geem. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Computer methods in applied mechanics and engineering*, 194(36-38):3902–3933, 2005.

[LG15] Liping Li and Honghao Gao. Test suite reduction for mutation testing based on formal concept analysis. In *2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 1–5. IEEE, 2015.

[LLN$^+$16] Bing Liu, Lucia, Shiva Nejati, Lionel C Briand, and Thomas Bruckmann. Simulink fault localization: an iterative statistical debugging approach. *Software Testing, Verification and Reliability*, 26(6):431–459, 2016.

[LNB$^+$16] Bing Liu, Shiva Nejati, Lionel Briand, Thomas Bruckmann, et al. Localizing multiple faults in simulink models. In *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, volume 1, pages 146–156. IEEE, 2016.

[LNB$^+$17] Bing Liu, Shiva Nejati, Lionel C Briand, et al. Improving fault localization for simulink models using search-based testing and prediction models. In *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 359–370. IEEE, 2017.

[LNLB19] Bing Liu, Shiva Nejati, Lucia, and Lionel C. Briand. Effective fault localization of automotive simulink models: achieving the trade-off between test oracle effort and fault localization accuracy. *Empir. Softw. Eng.*, 24(1):444–490, 2019.

[LO16] Nan Li and Jeff Offutt. Test oracle strategies for model-based testing. *IEEE Transactions on Software Engineering*, 43(4):372–395, 2016.

[LP91] Han Tong Loh and Panos Y Papalambros. A sequential linearization approach for solving mixed-discrete nonlinear design optimization problems. *J. Mech. Des.*, 1991.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

[LS04] J Lampinen and R Storn. New optimization techniques in engineering. *Studies in Fuzziness and Soft Computing. Springer-Verlag*, 141:123–166, 2004.

[LZ06] Edward A. Lee and Haiyang Zheng. Hyvisual: a hybrid system modeling framework based on ptolemy II. In Christos G. Cassandras, Alessandro Giua, Carla Seatzu, and Janan Zaytoon, editors, *2nd IFAC Conference on Analysis and Design of Hybrid Systems, ADHS 2006, Alghero, Italy, June 7-9, 2006*, volume 39 of *IFAC Proceedings Volumes*, pages 270–271. Elsevier, 2006.

[Mac64] R. G. Macfarlane. An enzyme cascade in the blood clotting mechanism, and its function as a biochemical amplifier. *Nature*, 202:498–499, 1964.

[Mat22a] Mathworks. Detect faults in aircraft elevator control system, 2022.

[Mat22b] Mathworks. Model-based systems engineering (mbse), 2022.

[Mat22c] Mathworks. Modeling a fault-tolerant fuel control system, 2022.

[Mat22d] Mathworks. Modeling an automatic transmission controller, 2022.

[Mat22e] Mathworks. Simulink — simulation and model-based design, 2022.

[Mat23a] Mathworks. Signal features, 2023.

[Mat23b] Mathworks. Structural coverage metrics, 2023.

[Mat24] Mathworks. Design narma-l2 neural controller in simulink, 2024.

[MGM$^+$17] Seyedali Mirjalili, Amir H Gandomi, Seyedeh Zahra Mirjalili, Shahrzad Saremi, Hossam Faris, and Seyed Mohammad Mirjalili. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in engineering software*, 114:163–191, 2017.

[Mir15] Seyedali Mirjalili. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-based systems*, 89:228–249, 2015.

[Mir16a] Seyedali Mirjalili. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural computing and applications*, 27:1053–1073, 2016.

[Mir16b] Seyedali Mirjalili. Sca: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems*, 96:120–133, 2016.

[ML16] Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in engineering software*, 95:51–67, 2016.

[MMC05] Efrén Mezura-Montes and Carlos A Coello Coello. A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Transactions on Evolutionary computation*, 9(1):1–17, 2005.

[MMC08] Efrén Mezura-Montes and Carlos A Coello Coello. An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *International Journal of General Systems*, 37(4):443–473, 2008.

[MMCCVRMD07] Efrén Mezura-Montes, CA Coello Coello, Jesús Velázquez-Reyes, and Lucıa Munoz-Dávila. Multiple trial vectors in differential evolution for engineering design. *Engineering Optimization*, 39(5):567–589, 2007.

[MMH16] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Abdolreza Hatamlou. Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications*, 27:495–513, 2016.

[MML14] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in engineering software*, 69:46–61, 2014.

[MMVRC06] Efrén Mezura-Montes, Jesús Velázquez-Reyes, and CA Coello Coello. Modified differential evolution for constrained optimization. In *2006 IEEE International Conference on Evolutionary Computation*, pages 25–32. IEEE, 2006.

[MN04] Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pages 152–166. Springer, 2004.

[MN13] Oded Maler and Dejan Ničković. Monitoring properties of analog and mixed-signal circuits. *International Journal on Software Tools for Technology Transfer*, 15(3):247–268, 2013.

[MNBB15] Reza Matinnejad, Shiva Nejati, Lionel C. Briand, and Thomas Bruckmann. Effective test suites for mixed discrete-continuous stateflow controllers. In Elisabetta Di Nitto, Mark Harman, and Patrick Heymans, editors, *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, ESEC/FSE 2015, Bergamo, Italy, August 30 - September 4, 2015*, pages 84–95, USA, 2015. ACM.

[MNBB19] Reza Matinnejad, Shiva Nejati, Lionel C. Briand, and Thomas Bruckmann. Test generation and test prioritization for simulink models with dynamic behavior. *IEEE Trans. Software Eng.*, 45(9):919–944, 2019.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[MoHUAGoI24] Ministry of Housing and Urban Affairs Government of India. Smartcities, 2024.

[MR14] Naser Moosavian and Babak Kasaee Roodsari. Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks. *Swarm and Evolutionary Computation*, 17:14–24, 2014.

[MS12] Ali Wagdy Mohamed and Hegazy Zaher Sabry. Constrained optimization based on modified differential evolution algorithm. *Information Sciences*, 194:171–208, 2012.

[MWR+15] Anitha Murugesan, Michael W. Whalen, Neha Rungta, Oksana Tkachuk, Suzette Person, Mats Per Erik Heimdahl, and Dongjiang You. Are we there yet? determining the adequacy of formalized requirements and test suites. In Klaus Havelund, Gerard J. Holzmann, and Rajeev Joshi, editors, *NASA Formal Methods - 7th International Symposium, NFM 2015, Pasadena, CA, USA, April 27-29, 2015, Proceedings*, volume 9058 of *Lecture Notes in Computer Science*, pages 279–294, Cham, 2015. Springer.

[MYPF19] Logan Mathesen, Shakiba Yaghoubi, Giulia Pedrielli, and Georgios Fainekos. Falsification of cyber-physical systems with robustness uncertainty quantification through stochastic optimization with adaptive restart. In *15th IEEE International Conference on Automation Science and Engineering, CASE 2019, Vancouver, BC, Canada, August 22-26, 2019*, pages 991–997, USA, 2019. IEEE.

[MZC+20] Chengying Mao, Xuzheng Zhan, Jinfu Chen, Jifu Chen, and Rubing Huang. Adaptive random testing based on flexible partitioning. *IET Softw.*, 14(5):493–505, 2020.

[NA09] Akbar Siami Namin and James H. Andrews. The influence of size and coverage on test suite effectiveness. In Gregg Rothermel and Laura K. Dillon, editors, *Proceedings of the Eighteenth International Symposium on Software Testing and Analysis, ISSTA 2009, Chicago, IL, USA, July 19-23, 2009*, pages 57–68. ACM, 2009.

[NCM16] Roberto Natella, Domenico Cotroneo, and Henrique S Madeira. Assessing dependability with software fault injection: A survey. *ACM Computing Surveys (CSUR)*, 48(3):1–55, 2016.

[NGM+19] Shiva Nejati, Khouloud Gaaloul, Claudio Menghi, Lionel C. Briand, Stephen Foster, and David Wolfe. Evaluating model testing and model checking for finding requirements violations in simulink models. In Marlon Dumas, Dietmar Pfahl, Sven Apel, and Alessandra Russo,

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

editors, *Proceedings of the ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/SIGSOFT FSE 2019, Tallinn, Estonia, August 26-30, 2019*, pages 1015–1025, USA, 2019. ACM.

[NKJ+17] Luan Viet Nguyen, James Kapinski, Xiaoqing Jin, Jyotirmoy V. Deshmukh, and Taylor T. Johnson. Hyperproperties of real-valued signals. In Jean-Pierre Talpin, Patricia Derler, and Klaus Schneider, editors, *Proceedings of the 15th ACM-IEEE International Conference on Formal Methods and Models for System Design, MEMOCODE 2017, Vienna, Austria, September 29 - October 02, 2017*, pages 104–113, USA, 2017. ACM.

[NM18] Gabriela Nicolescu and Pieter J Mosterman. *Model-based design for embedded systems*. Crc Press, 2018.

[NSF+10] Truong Nghiem, Sriram Sankaranarayanan, Georgios Fainekos, Franjo Ivancic, Aarti Gupta, and George J. Pappas. Monte-carlo techniques for falsification of temporal properties of non-linear hybrid systems. In Karl Henrik Johansson and Wang Yi, editors, *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2010, Stockholm, Sweden, April 12-15, 2010*, pages 211–220. ACM, 2010.

[NY20] Dejan Nickovic and Tomoya Yamaguchi. RTAMT: online robustness monitors from STL. In *Automated Technology for Verification and Analysis - 18th International Symposium, ATVA 2020, Hanoi, Vietnam, October 19-23, 2020, Proceedings*, pages 564–571, 2020.

[OHY11] Jungsup Oh, Mark Harman, and Shin Yoo. Transition coverage testing for simulink/stateflow models using messy genetic algorithms. In Natalio Krasnogor and Pier Luca Lanzi, editors, *13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011, Proceedings, Dublin, Ireland, July 12-16, 2011*, pages 1851–1858, USA, 2011. ACM.

[OJL+22] Sascha Julian Oks, Max Jalowski, Michael Lechner, Stefan Mirschberger, Marion Merklein, Birgit Vogel-Heuser, and Kathrin M Möslein. Cyber-physical systems in the context of industry 4.0: A review, categorization and outlook. *Information Systems Frontiers*, pages 1–42, 2022.

[OL11] Carol Oakley and Hannu Larjava. Hemostasis, coagulation, and complications. *Endodontic Topics*, 24(1):4–25, 2011.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[OP97] A Jefferson Offutt and Jie Pan. Automatically detecting equivalent mutants and infeasible paths. *Software testing, verification and reliability*, 7(3):165–192, 1997.

[OPB21] Oscar Eduardo Cornejo Olivares, Fabrizio Pastore, and Lionel Briand. Mutation analysis for cyber-physical systems: Scalable solutions and results in the space domain. *IEEE Transactions on Software Engineering*, 2021.

[oTG24] AIT Austrian Institute of Technology GMBH. Digibus Austria Automated Driving in local public transport, 2024.

[OU01] A Jefferson Offutt and Roland H Untch. Mutation 2000: Uniting the orthogonal. *Mutation testing for the new century*, pages 34–44, 2001.

[PCJ+17] Spencer Pearson, José Campos, René Just, Gordon Fraser, Rui Abreu, Michael D Ernst, Deric Pang, and Benjamin Keller. Evaluating and improving fault localization. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, pages 609–620. IEEE, 2017.

[PFK+13] Ludovic Pintard, Jean-Charles Fabre, Karama Kanoun, Michel Leeman, and Matthieu Roy. Fault injection in the automotive standard ISO 26262: An initial approach. In Marco Vieira and João Carlos Cunha, editors, *Dependable Computing - 14th European Workshop, EWDC 2013, Coimbra, Portugal, May 15-16, 2013. Proceedings*, volume 7869 of *Lecture Notes in Computer Science*, pages 126–133. Springer, 2013.

[PGZ+23] Alexandre Parant, François Gellot, Damien Zander, Véronique Carré-Ménétrier, and Alexandre Philippot. Model-based engineering for designing cyber-physical systems from product specifications. *Computers in Industry*, 145:103808, 2023.

[PIFJ21] Goran Petrovic, Marko Ivankovic, Gordon Fraser, and R. Just. Does mutation testing improve testing practices? In *Proceedings of the International Conference on Software Engineering (ICSE)*, 2021.

[PKZ+19] Mike Papadakis, Marinos Kintis, Jie Zhang, Yue Jia, Yves Le Traon, and Mark Harman. Mutation testing advances: an analysis and survey. In *Advances in Computers*, volume 112, pages 275–378. Elsevier, 2019.

[PLEBMBN18] Francisco Palomo-Lozano, Antonia Estero-Botaro, Inmaculada Medina-Bulo, and Manuel Núñez. Test suite minimization for mutation testing of ws-bpel compositions. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1427–1434, 2018.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

[PM11] Mike Papadakis and Nicos Malevris. Automatically performing weak mutation with the aid of symbolic execution, concolic testing and search-based testing. *Software Quality Journal*, 19(4):691–723, 2011.

[PM12] Mike Papadakis and Nicos Malevris. Mutation based test case generation via a path selection strategy. *Information and Software Technology*, 54(9):915–932, 2012.

[PMGM22] Fabrizio Pastore, Daniela Micucci, Michell Guzman, and Leonardo Mariani. TkT: Automatic inference of timed and extended pushdown automata. *IEEE Transactions on Software Engineering*, 48(2):617–636, 2022.

[PRWN16] Ingo Pill, Ivan Rubil, Franz Wotawa, and Mihai Nica. Simultate: A toolset for fault injection and mutation testing of simulink models. In *2016 IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 168–173. IEEE, 2016.

[PURMPL12] Macario Polo Usaola, Pedro Reales Mateo, and Beatriz Pérez Lamancha. Reduction of test suites using mutation. In *International Conference on Fundamental Approaches to Software Engineering*, pages 425–438. Springer, 2012.

[PV05] Konstantinos E Parsopoulos and Michael N Vrahatis. Unified particle swarm optimization for solving constrained engineering optimization problems. In *International conference on natural computation*, pages 582–591. Springer, 2005.

[PZL12] Kyung-Joon Park, Rong Zheng, and Xue Liu. Cyber-physical systems: Milestones and research challenges. *Computer Communications*, 36(1):1–7, 2012.

[Res23] Zion Market Research. Cyber-physical systems (cps) market size, share, trends & research 2030, 2023.

[RL03] Tapabrata Ray and Kim-Meow Liew. Society and civilization: an optimization algorithm based on the simulation of social behavior. *IEEE Transactions on Evolutionary Computation*, 7(4):386–396, 2003.

[RMP20] Kalaivani Rathakrishnan, Seung-Nam Min, and Se Jin Park. Evaluation of ecg features for the classification of post-stroke survivors with a diagnostic approach. *Applied Sciences*, 11(1):192, 2020.

[RNPS09] Esmat Rashedi, Hossein Nezamabadi-Pour, and Saeid Saryazdi. Gsa: a gravitational search algorithm. *Information sciences*, 179(13):2232–2248, 2009.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[RP76] KM Ragsdell and DT Phillips. Optimal design of a class of welded structures using geometric programming. *Journal of Engineering for Industry*, 1976.

[RR03] Manos Renieres and Steven P Reiss. Fault localization with nearest neighbor queries. In *18th IEEE International Conference on Automated Software Engineering, 2003. Proceedings.*, pages 30–39. IEEE, 2003.

[RS01] Tapabrata Ray and Pankaj Saini. Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Engineering Optimization*, 33(6):735–748, 2001.

[RSV11] R Venkata Rao, Vimal J Savsani, and Dipakkumar P Vakharia. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-aided design*, 43(3):303–315, 2011.

[RWK17] Rudolf Ramler, Thomas Wetzlmaier, and Claus Klammer. An empirical study on the application of mutation testing for a safety-critical industrial software system. In *Proceedings of the Symposium on Applied Computing*, pages 1401–1408, 2017.

[San90] E Sandgren. Nonlinear integer and discrete programming in mechanical design optimization. *J. Mech. Des.*, 1990.

[SBEH13] Ali Sadollah, Ardeshir Bahreininejad, Hadi Eskandar, and Mohd Hamdi. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing*, 13(5):2592–2612, 2013.

[SdSdS17] Rodolfo Adamshuk Silva, Simone do Rocio Senger de Souza, and Paulo Sérgio Lopes de Souza. A systematic review on search based mutation testing. *Information and Software Technology*, 81:19–35, 2017.

[SF12] Sriram Sankaranarayanan and Georgios Fainekos. Falsification of temporal properties of hybrid systems using the cross-entropy method. In *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, pages 125–134, 2012.

[SG23] Masume Saljuqi and Peyvand Ghaderyan. Combining homomorphic filtering and recurrent neural network in gait signal analysis for neurodegenerative diseases detection. *Biocybernetics and Biomedical Engineering*, 43(2):476–493, 2023.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

[Sim08] Dan Simon. Biogeography-based optimization. *IEEE transactions on evolutionary computation*, 12(6):702–713, 2008.

[SML17] Shahrzad Saremi, Seyedali Mirjalili, and Andrew Lewis. Grasshopper optimisation algorithm: theory and application. *Advances in engineering software*, 105:30–47, 2017.

[SMSJ17] Mustafa Saraoğlu, Andrey Morozov, Mehmet Turan Söylemez, and Klaus Janschek. Errorsim: A tool for error propagation analysis of simulink models. In *International Conference on Computer Safety, Reliability, and Security*, pages 245–254. Springer, 2017.

[SP97] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11:341–359, 1997.

[SP10] Matt Staats and Corina S. Pasareanu. Parallel symbolic execution for structural test generation. In Paolo Tonella and Alessandro Orso, editors, *Proceedings of the Nineteenth International Symposium on Software Testing and Analysis, ISSTA 2010, Trento, Italy, July 12-16, 2010*, pages 183–194, USA, 2010. ACM.

[SS20] Nikhil Kumar Singh and Indranil Saha. Specification-guided automated debugging of CPS models. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 39(11):4142–4153, 2020.

[SS22] R Sharma and N Sharma. Applications of artificial intelligence in cyber-physical systems. In *Cyber-Physical Systems*, pages 1–14. Chapman and Hall/CRC, 2022.

[SSK22] Harpreet Singh, Birmohan Singh, and Manpreet Kaur. An improved elephant herding optimization for global optimization problems. *Engineering with Computers*, pages 1–33, 2022.

[SVET10] Rickard Svenningsson, Jonny Vinter, Henrik Eriksson, and Martin Törngren. Modifi: a model-implemented fault injection tool. In Erwin Schoitsch, editor, *International Conference on Computer Safety, Reliability, and Security*, pages 210–222, Berlin, Heidelberg, 2010. Springer.

[SVS14] Daniel Skarin, Jonny Vinter, and Rickard Svenningsson. Visualization of model-implemented fault injection experiments. In *International Conference on Computer Safety, Reliability, and Security*, pages 219–230. Springer, 2014.

[TAC$^+$21] Quinn Thibeault, Jacob Anderson, Aniruddh Chandratre, Giulia Pedrielli, and Georgios Fainekos. Psy-taliro: A python toolbox for

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

search-based test generation for cyber-physical systems. In Alberto Lluch-Lafuente and Anastasia Mavridou, editors, *Formal Methods for Industrial Critical Systems - 26th International Conference, FMICS 2021, Paris, France, August 24-26, 2021, Proceedings*, volume 12863 of *Lecture Notes in Computer Science*, pages 223–231, Cham, 2021. Springer.

[TBAA15] Manel Tekaya, Mohamed Taha Bennani, Mohamed Abidi Alagui, and Samir Ben Ahmed. Aspect-oriented test case generation from matlab/simulink models. In Wojciech Zamojski, Jacek Mazurkiewicz, Jaroslaw Sugier, Tomasz Walkowiak, and Janusz Kacprzyk, editors, *Theory and Engineering of Complex Systems and Dependability - Proceedings of the Tenth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, June 29 - July 3 2015, Brunów, Poland*, volume 365 of *Advances in Intelligent Systems and Computing*, pages 495–504, Cham, 2015. Springer.

[TCPB+20] Thierry Titcheu Chekam, Mike Papadakis, Tegawendé F Bissyandé, Yves Le Traon, and Koushik Sen. Selecting fault revealing mutants. *Empirical Software Engineering*, 25(1):434–487, 2020.

[Tsa05] Jung-Fa Tsai. Global optimization of nonlinear fractional programming problems in engineering design. *Engineering Optimization*, 37(4):399–409, 2005.

[TTT21] Walid M Taha, Abd-Elhamid M Taha, and Johan Thunberg. *Cyber-Physical Systems: A Model-Based Approach*. Springer Nature, 2021.

[Um19] J Sup Um. Drones as cyber-physical systems. *Springer*, 10:978–981, 2019.

[VDB+22] Sten Vercammen, Serge Demeyer, Markus Borg, Niklas Pettersson, and Görel Hedin. Mutation testing optimisations using the clang front-end. *arXiv preprint arXiv:2210.17215*, 2022.

[VS05] Saša Vulinovic and Bernd-Holger Schlingloff. Model based dependability evaluation for automotive control functions. In *Invited Session: Model-Based Design and Test, 9th World Multi-Conference on Systemics, Cybernetics and Informatics, Florida*, 2005.

[WC95] Shyue-Jian Wu and Pei-Tse Chow. Genetic algorithms for nonlinear mixed discrete-integer optimization problems via meta-genetic parameter optimization. *Engineering Optimization+ A35*, 24(2):137–159, 1995.

[WCZF09] Yong Wang, Zixing Cai, Yuren Zhou, and Zhun Fan. Constrained optimization based on hybrid evolutionary algorithm and adaptive

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

constraint-handling technique. *Structural and Multidisciplinary Optimization*, 37:395–413, 2009.

[WGL+16] W Eric Wong, Ruizhi Gao, Yihao Li, Rui Abreu, and Franz Wotawa. A survey on software fault localization. *IEEE Transactions on Software Engineering*, 42(8):707–740, 2016.

[Wil92] Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer, 1992.

[WL10] Ling Wang and Ling-po Li. An effective differential evolution with level comparison for constrained engineering design. *Structural and Multidisciplinary Optimization*, 41:947–963, 2010.

[WM97] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[Y+19] Anupam Yadav et al. Aefa: Artificial electric field algorithm for global optimization. *Swarm and Evolutionary Computation*, 48:93–108, 2019.

[Yad21] Drishti Yadav. Blood coagulation algorithm: A novel bio-inspired meta-heuristic algorithm for global optimization. *Mathematics*, 9(23):3011, 2021.

[Yad24] Drishti Yadav. From fault injection to formal verification: A holistic approach to fault diagnosis in cyber-physical systems. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2024, page 1896–1900, New York, NY, USA, 2024. ACM.

[Yan10] Xin-She Yang. A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)*, pages 65–74. Springer, 2010.

[YD09] Xin-She Yang and Suash Deb. Cuckoo search via lévy flights. In *2009 World congress on nature & biologically inspired computing (NaBIC)*, pages 210–214. Ieee, 2009.

[YET20] Orhan Yaman, Fatih Ertam, and Turker Tuncer. Automated parkinson's disease recognition based on statistical pooling method using acoustic features. *Medical Hypotheses*, 135:109483, 2020.

[YF19] Shakiba Yaghoubi and Georgios Fainekos. Gray-box adversarial testing for control systems with machine learning components. In Necmiye Ozay and Pavithra Prabhakar, editors, *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation*

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

and Control, HSCC 2019, Montreal, QC, Canada, April 16-18, 2019, pages 179–184, USA, 2019. ACM.

[YHG12] Xin-She Yang and Amir Hossein Gandomi. Bat algorithm: a novel approach for global engineering optimization. *Engineering computations*, 29(5):464–483, 2012.

[YKH14] Xin-She Yang, Mehmet Karamanoglu, and Xingshi He. Flower pollination algorithm: a novel approach for multiobjective optimization. *Engineering optimization*, 46(9):1222–1237, 2014.

[YLW06] Qian Yang, J. Jenny Li, and David M. Weiss. A survey of coverage based testing tools. In Hong Zhu, Joseph R. Horgan, Shing-Chi Cheung, and J. Jenny Li, editors, *Proceedings of the 2006 International Workshop on Automation of Software Test, AST 2006, Shanghai, China, May 23, 2006*, pages 99–103, USA, 2006. ACM.

[YQ10] Quan Yuan and Feng Qian. A hybrid genetic algorithm for twice continuously differentiable nlp problems. *Computers & chemical engineering*, 34(1):36–41, 2010.

[YZH20] Bei Yu, Junlong Zhou, and Shiyan Hu. Cyber-physical systems: An overview. *Big data analytics for cyber-physical systems*, pages 1–11, 2020.

[ZAV$^+$04] Haissam Ziade, Rafic A Ayoubi, Raoul Velazco, et al. A survey on fault injection techniques. *Int. Arab J. Inf. Technol.*, 1(2):171–186, 2004.

[ZC05] Yuan Zhan and John A Clark. Search-based mutation testing for simulink models. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pages 1061–1068, 2005.

[ZC08] Yuan Zhan and John A. Clark. A search-based framework for automatic testing of matlab/simulink models. *J. Syst. Softw.*, 81(2):262–285, 2008.

[ZHA19] Zhenya Zhang, Ichiro Hasuo, and Paolo Arcaini. Multi-armed bandits for boolean connectives in hybrid system falsification. In Isil Dillig and Serdar Tasiran, editors, *Computer Aided Verification - 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I*, volume 11561 of *Lecture Notes in Computer Science*, pages 401–420. Springer, 2019.

[ZJP$^+$14] Daniele Zuddas, Wei Jin, Fabrizio Pastore, Leonardo Mariani, and Alessandro Orso. Mimic: locating and understanding bugs by analyzing mimicked executions. In *Proceedings of the 29th ACM/IEEE*

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

international conference on Automated software engineering, pages 815–826, 2014.

[ZK09] Erwie Zahara and Yi-Tung Kao. Hybrid nelder–mead simplex search and particle swarm optimization for constrained engineering design problems. *Expert Systems with Applications*, 36(2):3880–3886, 2009.

[ZKH03] Qianchuan Zhao, Bruce H Krogh, and Paul Hubbard. Generating test inputs for embedded control systems. *IEEE Control Systems Magazine*, 23(4):49–57, 2003.

[ZLA$^+$21] Zhenya Zhang, Deyun Lyu, Paolo Arcaini, Lei Ma, Ichiro Hasuo, and Jianjun Zhao. Effective hybrid system falsification using monte carlo tree search guided by qb-robustness. In Alexandra Silva and K. Rustan M. Leino, editors, *Computer Aided Verification - 33rd International Conference, CAV 2021, Virtual Event, July 20-23, 2021, Proceedings, Part I*, volume 12759 of *Lecture Notes in Computer Science*, pages 595–618. Springer, 2021.

[ZLW08] Min Zhang, Wenjian Luo, and Xufa Wang. Differential evolution with dynamic stochastic selection for constrained optimization. *Information Sciences*, 178(15):3043–3074, 2008.

[ZN09] Justyna Zander-Nowicka. *Model-based testing of real-time embedded systems in the automotive domain*. Technical University Berlin, Berlin, Germany, 2009.

[ZSD$^+$15] Aditya Zutshi, Sriram Sankaranarayanan, Jyotirmoy V Deshmukh, James Kapinski, and Xiaoqing Jin. Falsification of safety properties for closed loop control systems. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pages 299–300, 2015.

[ZW93] Chun Zhang and Hsu-Pin Wang. Mixed-discrete nonlinear optimization with simulated annealing. *Engineering Optimization*, 21(4):277–291, 1993.

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub