Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

R&D Project Proposal

Automated Test Generation for Robot Self-Examination

Salman Omar Sohail

Supervised by

Prof. Dr. Paul G. Plöger
Alex Mitrevski

May 2020

1 Introduction

Humanity has entered the era of cyber-physical-systems in which robotics is at the heart of it all. During the last 60 years robots have been continuously evolving and have expanded beyond the scope of industrial manufacturing which was its original purpose[6].

Robots are now being extensively used in domestic applications, agriculture, civic departments, etc. Now more than ever there has been a sharp increase in the implementation of robots across various domains which mandates the increase of testing, verification, and validation is done in systems to ensure dependability.

In the past, tests in robotic software were coded only when necessary and without structure, and if it did follow any structure, it commonly followed the standard software engineering practices but in the recent decade, things have changed, specifically that now, independent testing methodologies exclusively for robotics have been developed like the robotic unit tests [2].

As new standard testing methodologies and systems are developed for robots, [4][5][3] it must be kept in mind that importing practices from software engineering are simply not enough to verify complex and sophisticated robotic systems, in fact, implementing them would be completely obsolete in the context of interactions with the dynamic environment as they cannot encapsulate the global behavior of the robot [5].

This R&D aims to facilitate testing of complex robotic systems by automatically generating a set of simulated test case scenarios in which a robot assesses its own performance. The primary tasks that will be assessed are of robotic manipulation (i.e. testing whether a cup has been successfully grasped in a generated scenario). The test case scenarios will be generated by the stochastic yet plausible placement of different objects like a cup in a random pose within a simulated environment. These generated test case scenarios will vary in complexity (e.g. one scenario might chain multiple tasks together for testing). The core tools that will be utilized includes 'Robot Operating System' (ROS) and 'Gazebo'. As for the actual tests, actions-tests will be implemented which basically means that it will check if certain actions have been successfully completed (e.g. if the robot is actually picked a cup).

1.1 Motivation

One of the greatest challenges faced by companies that wish to use a fully autonomous robot is the prevention of faults that disrupt the robot from accomplishing its task[7]; these faults can range from faulty actuators and sensors to wrong decisions given by the task planner.

The most reliable way to prevent these faults is by intensively testing the robot which is a big-time sink, not to mention, time spent on testing it is time not spent by the robot working and time not spent working means time not spent earning profit.

This R&D will save companies, time, and money by enabling their robots to examine themselves for these faults in a variety of test case scenarios that range from grasping actions to the execution of complex scenarios automatically; all within a simulated environment ensuring reduced downtime and increased productivity! Once these faults have been identified, preventive measures can be implemented manually or by utilizing fault detection and diagnosis methods[8].

Additional benefits are extensive and repeated testing of robots in a large variety of scenarios, testing the robots with different objects and environments which may not be accessible, testing edge cases and finally providing the training data required by machine learning algorithms to enables the robot to learn from failed scenarios and apply corrected behavior.

1.2 Problem Statement

Research has been done in different domains like autonomous driving systems [9][1][13] to automatically generate test case scenarios for testing their systems, in this case, being their vehicles to solve problems like faulty decision making, insufficient test data, lack of equipment, time constraints, limited scenarios, resource strain, etc. This R&D builds upon these previous researches by applying their approaches to the robotic domain specifically for robotic manipulation to solve these same problems.

The evaluation for the development phase of the project will be based on the following:

1. Plausibility of the generated scenario.

2. Complexity of the generated scenario.

2 Related Work

In this section we will review some of the previously published work to this project as well as the current state of the art.

Fast temporal projection using accurate physics-based geometric reasoning[12]

The target of this research paper is to create a possible orientation for the robot's base framework while it uses manipulators arms for picking and placing objects while considering the future.

The approach being used by the paper is to create a projection of the future which is then analyzed to see if there is a violation that may disrupt the current solution. The projection for the future is done by a lightweight simulation. The projection generates a timeline for each of the different world states. From the generated world scenarios if one solution is better than another, it is taken as the current solution. Each failure is analyzed and recorded to match with future references. This is done by selecting samples from a probability distribution. The environment is defined by a 3D representation. An inference engine goes through various predicates to attempt to find a legal solution.

The proposed method seems to be the future of robotic planning. Its contributions are that is presents a hybrid approach that utilizes physics-based reasoning and inference-based reasoning to generate a timeline enabling accurate and effective positioning of a robotic base while a manipulator arm picks and places objects. It deals with the problem of abstraction when using only the first-order logic for robotic planning. It creates and utilizes timelines for planning in combination with the inference engine, physics engine, and the simulation engine.

Its shortcomings are that it is computationally expensive and a world database is imperative and it cannot do without one. This research paper is very similar to what this R&D plans to achieve. The key difference is that the approach used by this research paper has been developed specifically to verify and predict faults in the planning model of the robot, additionally, it does not simulate the complete trajectory of the robotic manipulator arm instead it just simulates key poses. The proposed R&D will be developed for testing purposes which includes the simulation of the complete trajectory of the robotic manipulator.

Paracosm: A Language and Tool for Testing Autonomous Driving Systems[10]

This paper presents a programming language PARACOSM that creates realistic environment simulation based on the unity physics engine. PARACOSM provides support for dynamically changing test case scenarios and environments. It generates meshes that can be read by a car's camera for test cases. It utilizes the quasi-Monte Carlo method to ensure continuous parameter creation with pure randomness. PARACOSM is based on reactive objects i.e. reactive objects store geometric and graphical characteristics of an object. Each reactive object has its own separate input stream, these streams provide the value of the environment object through sensors. Complex assemblies are created by the assembly of smaller simpler objects. This engine has a key feature that allows for the detection of collision amongst the generated objects.

The PARACOSM has the ability to do quantitative as well as qualitative analysis ensuring that just because a solution is reached means that the autonomous driving system has passed the test. Test case scenarios are built both for the static and dynamic reactive objects by the configuration of the input streams and variation of parameters. Test case scenario's using single deep neural network, history-oriented gradients, and (you only look once: Yolo) neural networks have shown promising results for PARACOSM.

The contributions of this paper are that it provides a programmable simulation framework that depicts the real world extensively for autonomous driving systems, additionally, it displays the coverage of its system by creating intricate test scenarios for the autonomous driving agent to go through.

The drawbacks are that there is no mechanic that computes the deformation of the vehicle after the collision and the test case is simply treated as a failure. This research paper provides a strong guideline for the development of dynamic scenarios that will be utilized in this R&D.

Abstract Simulation Scenario Generation for Autonomous Vehicle Verification[11]

This paper proposes a system that creates simulation scenarios to test the decision-making capability of autonomous vehicles through approaches of hardware verification. The framework uses a language that takes a series of inputs and makes a logical driving scenario while rejecting illegitimate simulation scenarios. The modules within the framework of this paper utilize inputs to generate the required random scenarios models in which the autonomous vehicle is let loose and its decisions are recorded. From the recorded data of the simulation, certain key data are additionally recorded which include safety as well as scenario definition. The scenarios models are created using the semantic language defined by the author which expresses certain parameters such as road generation, road types, parameter values, etc. The developed language is used as inputs and generates tokens that have additional attributes (e.g. B ¿ Bend Road ¿ attributes radius, degrees). The collection of these tokens is then used to generate an XML file that keeps track of all the components within the simulation that are to take place. A validation is check for inconsistencies and prevent invalid routes, this allows for a solid scenario formation and prevents wastage of time during the simulation.

The contribution of this research paper can be enumerated as the development of a semantic language for defining scenes in a simulation, development of techniques for connecting different sectors of the road, and a self-verification approach. Similar to the research paper [10], this paper opens up yet another approach for the generation of simulated scenarios and unlike previous research papers, it talks about testing and verification.

The drawback is that unlike the autonomous vehicles, robots with manipulators have an additional layer of complexity and are much more challenging, so, the scenarios and testing required are different in nature which is where this R&D come into play.

3 Project Plan

3.1 Work Packages

The bare minimum will include the following packages:

WP1 Literature Review

The aim of this work package will be to extensively search for related literature focused on simulation generation and robotic testing.

T1.1 Search for papers related to the project.

In this task the aim will be to search for papers related to the on simulation generation and robotic testing and what types of approaches they used.

T1.2 Analysis of reference list.

In this task we will go through the reference list and shortlist all the relevant research papers.

T1.3 Analysis of state of the art.

In this task we will go through the reference list and analyse all the state of the art.

WP2 Development package.

The aim of this work package is to provide an the required working components for the random scenario generator, action-tests and complex scenario tests.

T2.1 Random scenario generator.

In this task we will design and implement a random scenario generator in which the robot is required to a perform a set assignment in a simulated environment (gazebo).

T2.2 Action-tests.

In this task we will design and implement an action test in which the actions of the robotic manipulator will be tested.

T2.3 Complex scenario tests.

In this task we will design and implement complex scenario tests in which a number of actions tests will be performed by the robot in the randomly generated scenario.

WP3 Project Report.

This work package involves writing the project report. This work package will be scheduled in parallel to all the previous work packages.

T3.1 Introduction and related work.

This task deals with answering the first four W's. A detailed analysis of the state of the art should be documented in the project report.

T3.2 Design and development.

In this task we will list down all the steps we taken to design and implement the random scenario generator, action-test and complex scenario test.

T3.3 Evaluation report

In this task we will compile the result of our work.

T3.4 Conclusion and future work

In this task we add our concluding remarks and list down our suggestions on how performance of sensors/sensing modules can be improved and add details of any future work to the project report.

3.2 Milestones

M1 Literature search and Analysis - 30/Sep/2020

M2 Design and implementation of the random scenario generator - 30/Nov/2020

M3 Design and implementation of the action testing - 31/Jan/2020

M4 Design and implementation of the complex scenario tester - 30/Apr/2021

M5 Report submission - 15/May/2021

3.3 Project Schedule

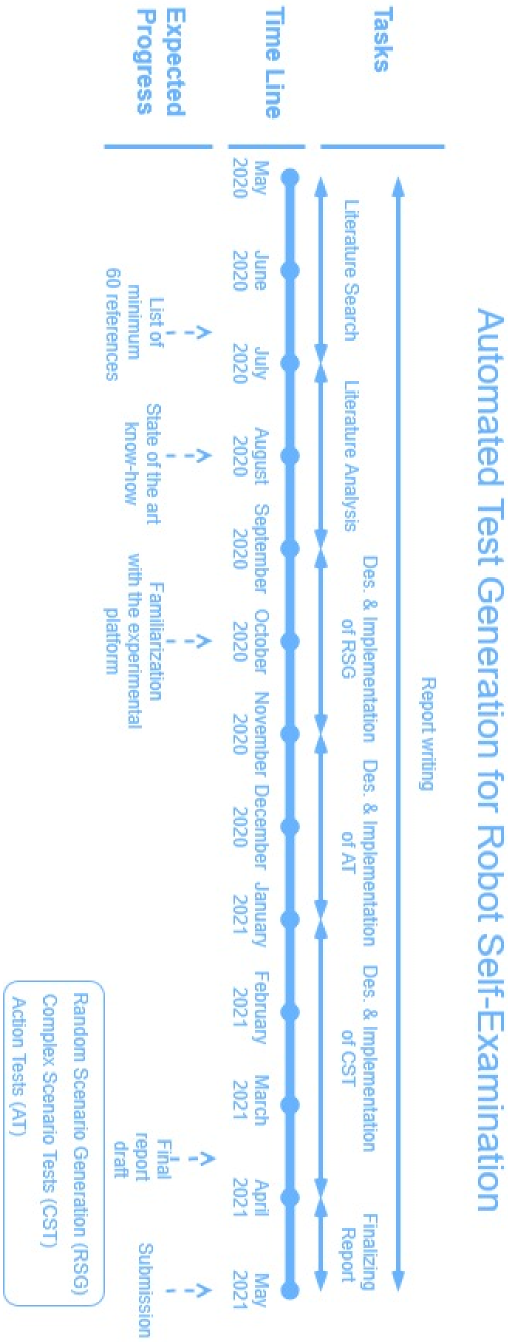

Figure 1: Project Schedule

3.4 Deliverables

Minimum Viable

- A survey on the related literature
- A detailed analysis of the state of the art.
- A fully implemented random scenario generator.
- Project report.

Expected

- Design and implementation of at least one action-test.

Desired

- Design and implementation of complex scenario tests.

References

[1] James Arnold and Rob Alexander. Testing autonomous robot control software using procedural content generation. In Friedemann Bitsch, Jérémie Guiochet, and Mohamed Kaâniche, editors, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8153 LNCS, pages 33–44, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 9783642407925. doi: 10.1007/978-3-642-40793-2_4.

[2] Andreas Bihlmaier and Heinz Wörn. Robot unit testing. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8810, pages 255–266, 2014.

[3] Andreas Bihlmaier, Fabian Stein, and Heinz Wörn. Towards a generic BRDF/BTF measurement system: Improving visual realism in robot simulators using robots. In *2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies, EmergiTech 2016*, pages 410–416, 2016. ISBN 9781509007066. doi: 10.1109/EmergiTech.2016.7737376.

[4] Juhan Ernits, Evelin Halling, Gert Kanter, and Jüri Vain. Model-based integration testing of ROS packages: A mobile robot case study. In *2015 European Conference on Mobile Robots, ECMR 2015 - Proceedings*, pages 1–7, 2015. ISBN 9781467391634. doi: 10.1109/ECMR.2015.7324210.

[5] Vladimir Estivill-Castro, René Hexel, and Carl Lusty. Continuous integration for testing full robotic behaviours in a GUI-stripped simulation. In *CEUR Workshop Proceedings*, volume 2245, pages 453–464, 2018.

[6] Elena Garcia, Maria Antonia Jimenez, Pablo Gonzalez De Santos, and Manuel Armada. The evolution of robotics research. *IEEE Robotics and Automation Magazine*, 14(1):90–103, 2007. ISSN 10709932. doi: 10.1109/MRA.2007.339608.

[7] Abdelfetah Hentout, Aouache Mustapha, Abderraouf Maoudj, and Isma Akli. Key challenges and open issues of industrial collaborative robotics. In *RO-*

MAN 2018: Workshop on Human-Robot Interaction: From Service to Industry, number August, 2018.

[8] Eliahu Khalastchi and Meir Kalech. On fault detection and diagnosis in robotic systems. *ACM Computing Surveys*, 51(1), jan 2018. ISSN 15577341. doi: 10.1145/3146389.

[9] Philip Koopman and Michael Wagner. Challenges in Autonomous Vehicle Testing and Validation. *SAE International Journal of Transportation Safety*, 4(1):15–24, 2016. ISSN 23275634. doi: 10.4271/2016-01-0128.

[10] Rupak Majumdar, Aman Mathur, Marcus Pirron, Laura Stegner, and Damien Zufferey. Paracosm: A language and tool for testing autonomous driving systems. 02 2019.

[11] Christopher Medrano-Berumen and Mustafa Akbaş. Abstract simulation scenario generation for autonomous vehicle verification. 04 2019.

[12] L. Mösenlechner and M. Beetz. Fast temporal projection using accurate physics-based geometric reasoning. In *2013 IEEE International Conference on Robotics and Automation*, pages 1821–1827, 2013.

[13] Kenta Takaya, Toshinori Asai, Valeri Kroumov, and Florentin Smarandache. Simulation environment for mobile robots testing using ROS and Gazebo. In *2016 20th International Conference on System Theory, Control and Computing, ICSTCC 2016 - Joint Conference of SINTES 20, SACCS 16, SIMSIS 20 - Proceedings*, pages 96–101, 2016. ISBN 9781509027200. doi: 10.1109/ICSTCC.2016.7790647.