Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

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 15, 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 [5].

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 different 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 [1].

As new standard testing methodologies and systems are developed for robots [3] [4] [2], it must be kept in mind that importing practices from software engineering is simply not enough to verify complex and sophisticated robotic systems especially in the context of the interactions with the environment [4].

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 (e.g. 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 has actually picked a cup).

1.1 Motivation

One of the greatest challenges faced by users that wish to use a fully autonomous robot is the prevention of faults that disrupt the robot from accomplishing its task [6]; 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 consumes a lot of time and resources.

The aim of this R&D is to enable the user's 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 saving time and resources.

Additional benefits include exhaustive testing of the 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

This R&D proposal addresses the problem of how to expansively test a robotic-manipulators interaction with the environment to identify faults, faults like erroneous decisions given by the task planner, or just faulty components.

The objective of this R&D is to enable robotic-manipulators to extensively examine themselves for these faults automatically within a simulated environment. The data gained from testing will then be processed by a machine learning algorithms that will enable the robot to apply corrected 'behavior' learned from failed scenarios due to faults (if possible).

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[9]

The target of this research paper is to identify a set of valid positions and orientations for a mobile robot's base while it uses its manipulators for picking and placing objects by considering predictions from simulated events.

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 contributions of this research paper is that it presents hybrid approach that utilizes physics-based reasoning and inference-based reasoning to generate a timeline for detecting faults in a robots planning model. This enables accurate and effective positioning of a robotic base while the manipulator arm picks and places objects. Additionally, it utilizes the generated timelines for path-planning and to create key positions for the robotic-manipulators trajectory.

Its shortcomings are that it is computationally expensive and requires a world database. This research paper is very similar to what we plan to achieve with our R&D. The key differences are that the approach used by this research paper has been developed specifically to detect faults in the planning model of a robot, additionally, it does not simulate the complete trajectory of the robotic manipulator but only the key positions. The proposed R&D will be developed not only for testing the planning model but detecting the point of failure in the robotic component as well, moreover, we plan to simulate the complete trajectory of the robotic manipulator as it can reveal additional points of failure.

Paracosm: A Language and Tool for Testing Autonomous Driving Systems[7]

This paper presents a programming language PARACOSM that creates realistic environment simulation based on the Unity physics engine for autonomous driving systems. 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.

PARACOSM has the ability to do quantitative as well as qualitative analysis ensuring that because a solution is reached, the autonomous driving system has passed the test. Test case scenarios are built both for static and dynamic reactive objects by the configuration of the input streams and variation of parameters. Test case scenarios 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[8]

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 scenario models in which the autonomous vehicle performs its assigned task 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 scenario models are created using a 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 input 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 performed for checking inconsistencies and preventing invalid routes; this allows for a solid scenario formation and prevents wasting time during the simulation.

The contribution of this research paper can be summarized, as the development of a semantic language for defining scenes in a simulation and developing an approach for checking the plausibility of a generated scenario. Similar to [7], 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 Proposal Approach

Our approach for expansively testing a robotic-manipulators interaction with the environment to identify faults will be as follow:

- Designing and implementing random scenario generator.
- Designing and implementing action-tests.
- Designing and implementing complex scenario tests.

3.1 Tools and Environment

- Device Under Test(DUT) - Toyota's Human Support Robot (HSR)
- Simulation Environment - Gazebo
- Robot Application - Robot Operating System (ROS)

3.2 Random Scenario Generator

The starting point for the development of the random simulator generator will be to expand the existing 3D model data base. This will allow for more variety of test case scenarios to be generated. The next step would be to design and implement a parameter randomizer to create diverse scenarios.

3.3 Action-Tests

The action tests will be developed in a very similar manner to property based tests. These tests will be developed for the Toyota (HSR) robot.

3.4 Complex Scenario Tests

The complex scenario tests will be developed by chaining a series of action tests.

3.5 Proposed Tasks

The robot will have to perform these tasks in a randomly generated environment and be able perform the action-tests.

1. Picking an object from the table.
2. Picking an object from a cluttered table.
3. Picking and placing an object on the same table.

4 Project Plan

4.1 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 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 introducing the reader to the topic of automated test generation for robot self-examination as well as its related literature.

T3.3 State-of-the-art

In this task we will provide detailed analysis of the state-of-the-art.

T3.4 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.

T3.5 Evaluation report

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

T3.6 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.

4.2 Milestones

M1 Literature search and Analysis - 31/Jul/2020

M2 Design and implementation of the random scenario generator - 15/Sep/2020

M3 Design and implementation of the action testing - 15/Oct/2020

M4 Design and implementation of the complex scenario tester - 15/Dec/2021

M5 Report submission - 15/Jan/2021

4.3 Project Schedule

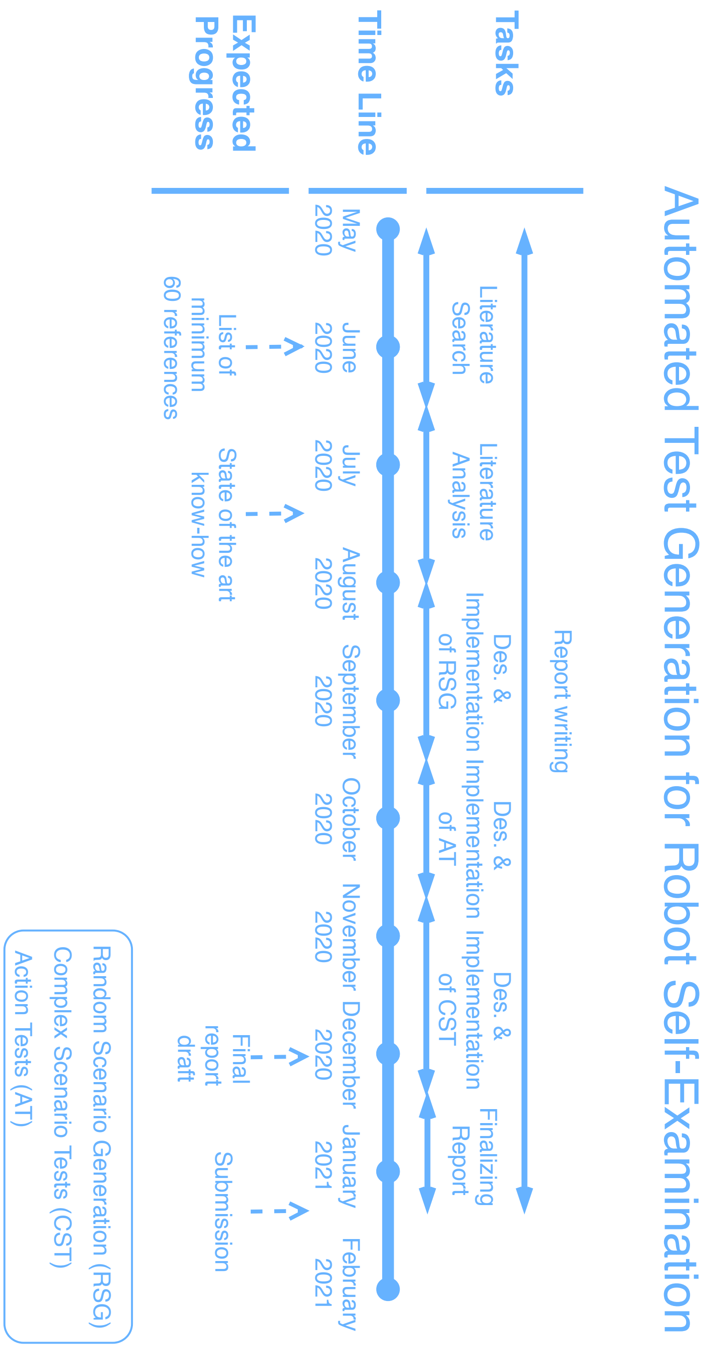

Figure 1: Project Schedule

4.4 Deliverables

Minimum Viable

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

Expected

- Design and implementation of at least one action-test.
- Design and implementation of complex scenario tests.

Desired

- Generating reports for each task that the robot has performed which can than be used to teach the robot.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

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

[9] 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.