Master's Thesis

Property-Based Testing: Formalized Robotic Testing for Standard Compliance

Salman Omar Sohail

Submitted to Hochschule Bonn-Rhein-Sieg,
Department of Computer Science
in partial fullfilment of the requirements for the degree
of Master of Science in Autonomous Systems

Supervised by

Prof. Dr. Nico Hochgeschwender
Prof. Dr. Paul G. Plöger
Sven Schneider

December 2022

I, the undersigned below, declare that this work has not previously been submitted to this or any other university and that it is, unless otherwise stated, entirely my own work.

Date

Salman Omar Sohail

Abstract

Robotic safety systems and standards are used to preserve and ensure the safety between a robot and its interaction with the environment. Prior robotic standards ensured safety via enclosing the robot in cages, specifically for industrial robots. Nowadays, with the rapid development of cognition and intelligence based robots that perform complex tasks, it has become a recurrent requirement for them to work along side humans, and these standards are evolving to meet these requirements. However, verifying and validating these standards on a robot remain a cumbersome process and un-automated.

To accelerate the verification and validation process of existing standards in a hierarchical and logical manner, we investigate current relevant robotic standards and develop a machine semantic language that translates the key information from these standards formalizing them into properties for a robotic system. These properties will then be used to automatically verify and validate them using property-based testing framework [1]. Furthermore, we will examine the formulation of several applicable standards into use-cases and vice versa.

Acknowledgements

Thanks to ....

Contents

List of Figures

List of Tables

1

Introduction

Autonomous robots are machines that perform designated tasks automatically as well as independently [11]. The designated tasks that they perform range from simple navigation tasks to complex assembly tasks. Industries nowadays are heavily adopting and utilizing robotics to enhance their efficiency in terms of production and not only industries but commercial and domestic markets as well [12].

Safety is a great concern when it comes to these autonomous robots due to which new standards are continuously being proposed and updated. Manual methods of testing the safety of these autonomous machines become in feasible due to the wide range of scenarios available [13, 14], hence, a shift towards virtual test drives is being made. Virtual test drives ensure good coverage of testing safety in robotics. Despite the simulation-reality gap, virtual testing provides a minimum theoretical guarantee of a working system i.e. if a system passes a virtual test, it may or may not pass a comparable reality test due to the simulation-gap, however, if a system fails a virtual test, then it is highly unlikely to pass a comparable reality test.

This Master Thesis aims enable automated robotic standard compliance testing (e.g. ISO/TR 23482-1) using an evolved form of property-based testing [1]. The evolved form of property-based testing has been restructured and modularized, moreover, it utilizes a domain specific language () for testing. The primary compliance tests will focus on standards that are simulation compatible and feasible. The use-case includes an un-manned ground vehicle (i.e. Clearpath Husky) that is automatically tested for several standard compliance tests as well custom user-based tests (e.g. testing person collision). The project utilizes an assortment of software-tools some of the core ones being ROS, Gazebo, Hypothesis, textX, and Allure.

1.1 Motivation

1.1.1 Safety

1.1.2 Security

1.1.3 Reliability

1.1.4 Performance

1.2 Challenges and Difficulties

1.2.1 ...

1.2.2 ...

1.2.3 Simulation

sim gap

1.3 Problem Statement

This master thesis addresses the problem of how to extensively verify and validate standard compliance in robotic actions and behaviours via property-based testing?

Two aspects are queried:

1. Investigation and formalization of the systematic process of identifying and converting key features of existing robotic standards into **properties** that can be tested and validated via the property-based testing framework [1]?

2. Developing a process of translating existing robotic standards into verifiable use-cases and vice versa?

1.3.1 ...

1.3.2 ...

2
Related Work

2.1 Standard Compliant Robotic Frameworks

A work that focuses on standard compliance for robotic testing has been presented by Cosmo et al. [15] in which they proposed a verification methodology for ensuring compliance with the protective separation distance (PSD) requirement issued by the ISO standards [8, 16]. The verification methodology utilizes velocity dependent safety zones (dynamic safety zones) which are achieved via bounding volumes (BV). The BV calculates whether an assigned imaginary volume that encompasses a robot and a personnel intersect. Dependent on the intersection, the robot either lowers its speed or completely halts. The motivation of this work stems for increasing safety in human-robot collaboration (HRC) tasks. Experiments have been conducted with the Franka-Emika robot in which comparisons were made with the recommended static safety zones by ISO [4] and the proposed dynamic safety zones. The result of the comparison demonstrated a 10% efficiency increase.

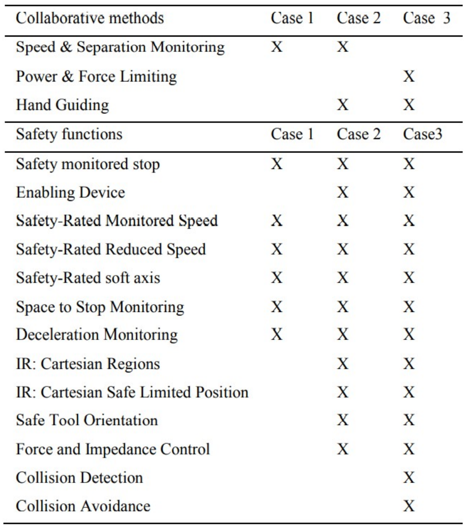
A different approach for the verification of standard compliance was performed by Guiochet et al. [17] in which they investigated the steps required to create a safe and reliable Multimodal Interactive Robot for Assistance in Strolling (MIRAS)[1]. Based on the investigation, steps were taken for ensuring the safety of the human-user and environment which included designing, risk-assessment, hazard-operability, and safety-cases. The motivation for this work was to create a human-rehabilitation robot that encompasses all necessary safety standards provided by various regional as well as international regulatory organizations. In this work, non-medical robotic standards have been phased out due to non-applicability in MIRAS such as ISO10218 and ISO13482. The standards that have been utilized are ISO guide 51, ISO guide 63, ISO31000, ISO12100, ISO14971, ISO7176-5, ISO11199-2, and Machinery Directive 2006/42/EC. Moreover, medical experts feedback was used in the development of MIRAS. The safety validation of the robot was achieved via a risk-acceptability matrix as well as through safety-cases.

Rosenstrauch et al. presents an overview of the ISO/TS 15066 standard as well as an implemented demonstrative use-case. The standard ISO/TS 15066 itself extends and updates the ISO 10218 standard in context of collaborative robots fleshing out its details. An example is that ISO 10218 requires that robotic movement ”should not create an injury” and the ISO /TS 15066 provides parameters and limits for force and speed to act in accordance with ISO 10218. Similar to this example, several other safety-based skills

[1] A robotic rollator.

are given in the ISO/TS 15066. The use-case includes a 6 DOF manipulator arm consisting of *Schunk Amtec PowerCube* with the safety-skill of power and speed limiting extracted from ISO/TS 15066. The method of validation of this experiment was by measuring the deformation of a subjects flesh when the manipulators end-of-effector came in contact with it. The result of this use-case was that the parameters issued by the ISO/TS 15066 proved to be insufficient as the flesh was penetrated beyond reasonable limits stated by the standard. It seems to the authors that the equations and parameters provided by ISO/TS 15066 are suited more to ”modern light weight” robots and not to heavy industrial robots. This research strengthens our own proposed framework of property-based tests which validates safety-requirement use-cases and has the ability to vary parameters in accordance to the robot being used and verifying that no constraints or limitations have been being violated.

Michalos et al. [2] present the result of their investigation of safety in HRI in accordance to the european directive as well as international standards. They narrow down the relevant safety features from several standards some of which are listed in table 2.1 and 2.2 and provide a brief and descriptive overview of these safety features. In addition to the identification of relevant safety mechanisms from the standards, they present three use-cases which utilize the safety-mechanisms. Three of the use-cases are robotic manipulator arms that are working in an industrial setting with the task of assembling of a given component; the first being a automotive dashboard assembly, the second being an automotive rear suspension assembly, and finally the third being a refrigerators assembly. An evaluation was performed on the three use-cases which had implementation of various safety-mechanisms. The result of the evaluation can be viewed in Fig. 2.1. One of the conclusions that can be taken from their work is that the safety-mechanisms and the their final goals cannot be set in stone and have to be adapted to the work requirement. For example in use-case:1 and use-case:2, separation of the robot to the human was acceptable but for use-case:3 separation was unavoidable. Flexibility in safety-verification and validation methodology plays a huge role in the deployment of robotics commercially.



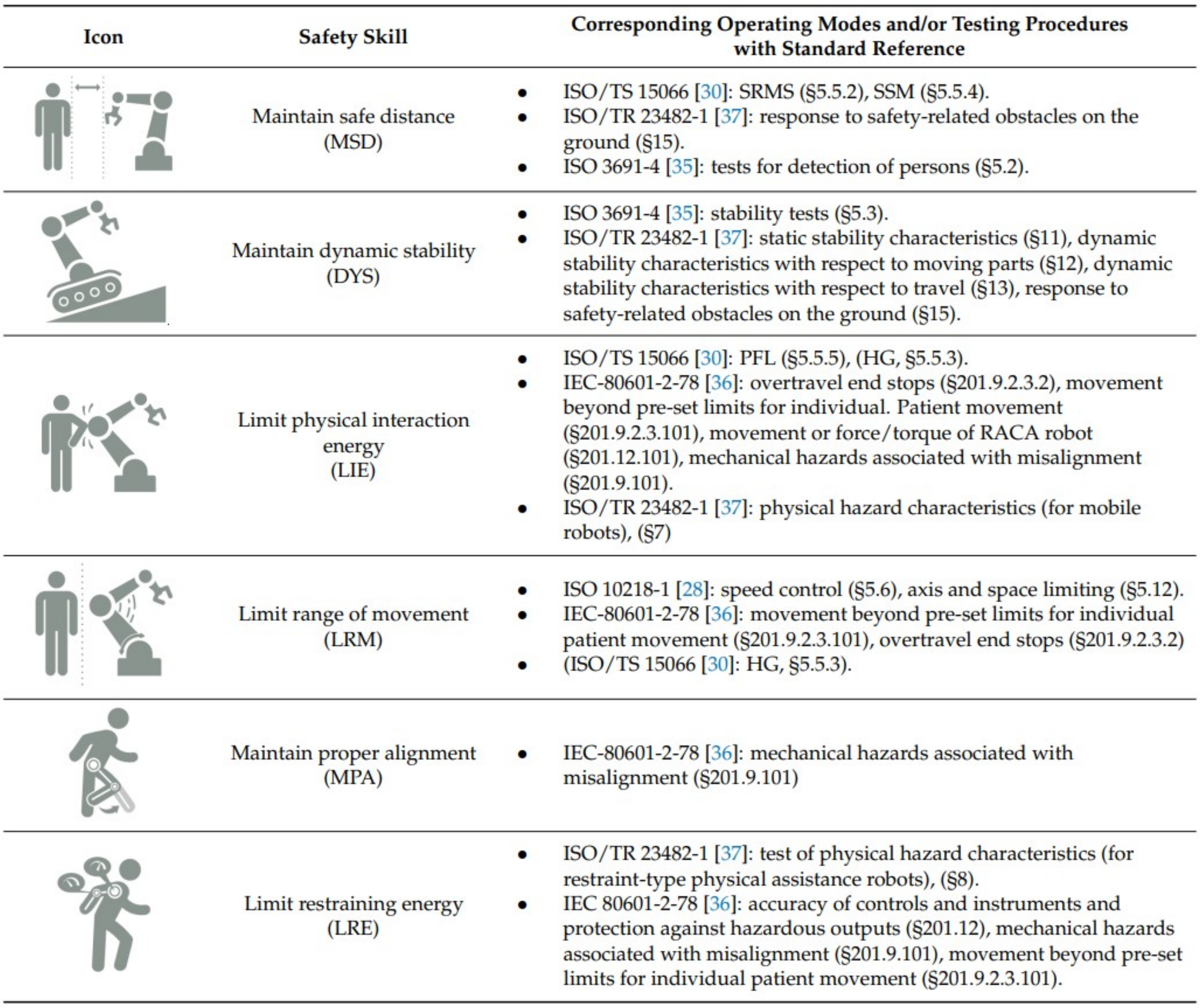
Collaborative methods	Case 1	Case 2	Case 3
Speed & Separation Monitoring	X	X	
Power & Force Limiting			X
Hand Guiding		X	X
Safety functions	Case 1	Case 2	Case3
Safety monitored stop	X	X	X
Enabling Device		X	X
Safety-Rated Monitored Speed	X	X	X
Safety-Rated Reduced Speed	X	X	X
Safety-Rated soft axis	X	X	X
Space to Stop Monitoring	X	X	X
Deceleration Monitoring	X	X	X
IR: Cartesian Regions		X	X
IR: Cartesian Safe Limited Position		X	X
Safe Tool Orientation		X	X
Force and Impedance Control		X	X
Collision Detection			X
Collision Avoidance			X

Figure 2.1: Figure by Michalos et al. [2]. The figure illustrates the result of the application of several safety-mechanisms in the three use-cases that were investigated.

A comprehensive survey conducted by [18] Herrmann et al. shows that many of the testing methodologies employed are not generalizable outside their designated scenarios. The work describes the development of safety-related frameworks that ensure standard compliance in certain scenarios. The testing methodologies discussed pertain to safety-planners, auxiliary human-monitoring sensors, risk-acceptability matrix, etc.

2.2 Robotic Standards

A comprehensive and in-depth article has been published by Valori et al. [3] which covers standard safety-tests and their applications in context of human robot interaction (HRI).



Icon	Safety Skill	Corresponding Operating Modes and/or Testing Procedures with Standard Reference
	Maintain safe distance (MSD)	• ISO/TS 15066 [30]: SRMS (§5.5.2), SSM (§5.5.4). • ISO/TR 23482-1 [37]: response to safety-related obstacles on the ground (§15). • ISO 3691-4 [35]: tests for detection of persons (§5.2).
	Maintain dynamic stability (DYS)	• ISO 3691-4 [35]: stability tests (§5.3). • ISO/TR 23482-1 [37]: static stability characteristics (§11), dynamic stability characteristics with respect to moving parts (§12), dynamic stability characteristics with respect to travel (§13), response to safety-related obstacles on the ground (§15).
	Limit physical interaction energy (LIE)	• ISO/TS 15066 [30]: PFL (§5.5.5), (HG, §5.5.3). • IEC-80601-2-78 [36]: overtravel end stops (§201.9.2.3.2), movement beyond pre-set limits for individual. Patient movement (§201.9.2.3.101), movement or force/torque of RACA robot (§201.12.101), mechanical hazards associated with misalignment (§201.9.101). • ISO/TR 23482-1 [37]: physical hazard characteristics (for mobile robots), (§7)
	Limit range of movement (LRM)	• ISO 10218-1 [28]: speed control (§5.6), axis and space limiting (§5.12). • IEC-80601-2-78 [36]: movement beyond pre-set limits for individual patient movement (§201.9.2.3.101), overtravel end stops (§201.9.2.3.2) • (ISO/TS 15066 [30]: HG, §5.5.3).
	Maintain proper alignment (MPA)	• IEC-80601-2-78 [36]: mechanical hazards associated with misalignment (§201.9.101)
	Limit restraining energy (LRE)	• ISO/TR 23482-1 [37]: test of physical hazard characteristics (for restraint-type physical assistance robots), (§8). • IEC 80601-2-78 [36]: accuracy of controls and instruments and protection against hazardous outputs (§201.12), mechanical hazards associated with misalignment (§201.9.101), movement beyond pre-set limits for individual patient movement (§201.9.2.3.101).

Figure 2.2: Figure by Valori et al. [3]. The figure illustrates identified standard compliant safety-skills for robots. Standards utilized are: ISO/TS 15066- [4], ISO/TR 23482-1 [5], ISO 3691 [6], IEC-80601-2-78 [7], ISO 10218-1 [8]

In the article, they summarize thirteen standards applicable to autonomous systems and list out the essential information on the safety-test protocols, recovery methods, and required safety-skills for the autonomous system in question and provide existing use-cases for them as well. Moreover, they point out the conflicts between standards; one of particular interest is the one in which a manipulator platform has a person's speed taken into account according to ISO 15066 [4], whereas ISO 3691 [6] does not. They mention that although these standards serve as a soft law and are not legally binding, it is imperative for the standards to be consistent and coherent. Another interesting part of their work was the complied safety-skills that a robot must have extracted from the standards. This is depicted in Fig. 2.2.

In terms of standard safety framework enhancement, Chemweno et al. [9] proposed a new framework ISO 31000 for designing safeguards for collaborative robots to cover up the gaps of hazards and risks left by ISO 15066 (Normative standard for robots and robotic devices - collaborative robots) which is an extension of ISO 10218 (Standard for robots and robotic devices - safety requirements for industrial robots). The drive for his work was to increase safety in collaborative robots applications specifically when the robot shares a space with a human or there is a shared task between the human and the robot. Based on his works analysis, ISO 15066 and ISO 10218 introduces and emphasize hazard analysis and safety but does not provide a concrete guideline for its realization. In order to facilitate its realization, Chemweno et al. presented a systematic framework for designing and implementing the safety standards as presented in Fig.2.3.

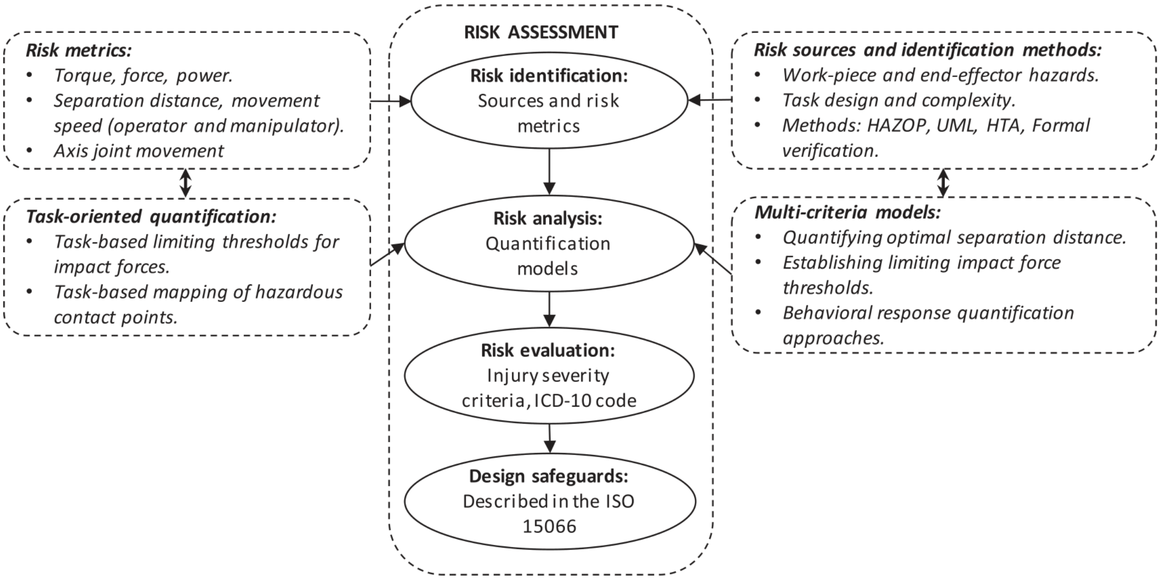

RISK ASSESSMENT
Risk identification: Sources and risk metrics
Risk analysis: Quantification models
Risk evaluation: Injury severity criteria, ICD-10 code
Design safeguards: Described in the ISO 15066
Risk metrics:
• Torque, force, power.
• Separation distance, movement speed (operator and manipulator).
• Axis joint movement
Task-oriented quantification:
• Task-based limiting thresholds for impact forces.
• Task-based mapping of hazardous contact points.
Risk sources and identification methods:
• Work-piece and end-effector hazards.
• Task design and complexity.
• Methods: HAZOP, UML, HTA, Formal verification.
Multi-criteria models:
• Quantifying optimal separation distance.
• Establishing limiting impact force thresholds.
• Behavioral response quantification approaches.

Figure 2.3: ISO 15066 proposed safety extension framework by Chemweno et al. [9]

The result of his work was a framework that covers the gaps left in standards and provides insights on what should be considered in the development of this thesis project.

An article published by Harper et al. [19] provides an overview on the enhancements performed in the ISO 10218 and the development of ISO 13482 standards for robots. The key-points discussed for ISO 10218 was the addition of human-robot collaboration mode which included the stopping function, speed and position control, power and force control, and finally operation work-spaces. As for ISO 13482 the key-features were designating safety-requirements for the different types of robots such as task definition, environment designation, hazard identification, and validation tests as well as design safety cases. The work provides a keen insight on the development and decision making by the ISO committee for the formulation and consideration required for standard definitions.

Following up on ISO 13482, Weng et al. [20] discusses the safety-features of ISO 13482 [21]. In their work they talk about the safety results needed by three types of robots: the mobile servant robot, the person carrier robot, and the physical assistant robot. A highlight of their work is that the standard ISO 13842 is agnostic to how the results of a robot are achieved. This means that it does not take into consideration how the software planner plans an action or sequence of actions but only the final course of action/s performed by the robot. The works also discusses ethics and legal liability of the robots and how it lies with the integrator of the robotic system. This article provides vital information that should be considered while developing a standard based safety-framework and the formulation process is similar to that which we wish to achieve i.e. focus on robot safety violations and not in the software planners.

2.3 Robotic Standard Deficiencies

The current standards that are available are lacking in several areas of which the most prominent ones being in law and the estimation of the capabilities of different autonomous machines. A good work that discusses the legal implications of robotic laws is presented by Villaronga et al. [22]. In the work they illustrate the intricate difficulties between governmental and non-governmental regulators for robotic laws (including service robots, self-driving cars, delivery robots, and health-care robots). One of the emphasized points in their articles is that the current robotic standards are heavily politically driven and there are no clear distinction between hard laws and soft laws for the autonomous technologies (Hard laws are those laws that are reprimandable by law whereas soft laws serve more as a guideline). Moreover, there appears to be no clarity in the robotics laws themselves in terms of area of applicability (e.g. cyber-threats, psychological damage, responsibility, evolving cognition, societal seclusion, etc.). A few potential solution suggested were the assimilation of private and public impact assessments into new policies, an evidence based-evaluation for the implemented technologies (Smart regulations), and finally a stringent policy for ethics on robotic integrators and designers. Relating to this thesis, this work highlights that existing and new impending robotic laws may not serve as a sufficient source of reliability and hence the ability to test robots has to be expansive and flexible enough to integrate different scenarios which in turn may serve as a positive legal argument.

2.4 Standards Review

The following section comprises of the standards identified to be relevant when considering safety in robotics.

ID	**Standard**	**Relevance**
ISO 51:2014 [23]	Safety aspects - Guidelines for their inclusion in standards	
ISO 10218-1:2011 [8]	Robots and robotic devices-safety requirements for industrial robots-part 1: robots	

ISO 10218-2:2011 [16]	Robots and robotic devices-safety requirements for industrial robots-part 2: robot systems and integration	
ISO 12100:2010 [24]	Safety of machinery-general principles for design-risk assessment and risk reduction	
ISO 13482:2014 [21]	Robots and robotic devices-safety requirements for personal care robots	
ISO 13849-1		
ISO 13849-2		
ISO 13851:2019 [25]	Safety of machinery-two-hand control devices-principles for design and selection	
ISO 13855:2010 [26]	Safety of machinery-positioning of safeguards with respect to the approach speeds of parts of the human body	
ISO 14971		
ISO/TS 15066:2016 [4]	Robots and robotic devices-collaborative robot	
ISO 18373		
ISO 18497:2018 [27]	Agricultural machinery and tractors-safety of highly automated machinery	
ISO 18646-1:2016	Robotics - Performance criteria and related test methods for service robots - Part 1: Locomotion for wheeled robots	
ISO/TR 20218-1:2018 [28]	Robotics-safety design for industrial robot systems-part 1: end-effectors	
ISO/TR 20218-2:2017 [29]	Robotics-safety requirements for industrial robots-part 2: manual load/unload stations	
ISO/TR 23482-1:2020 [5]	Application of ISO 13482-part 1: safety-related test methods	
ISO 31000:2018	Risk management - Guidelines	
ISO 3691-4:2020 [6]	Industrial trucks-safety requirements and verification-part 4: Driverless industrial trucks and their systems	

Table 2.1: International standards used for design and safety aspects for autonomous robots.

2.5 Limitations of previous work

At present, there is a lack of research that is available on automated testing for robotic standard compliance. Although academic papers [15, 17, 18, 30] are available in which researchers and industrialist demonstrate standard compliance for certain robots using customized safety-frameworks, planners, auxiliary

Table 2.2: European standards used for design and safety aspects for autonomous robots. Information from [2]

ID	**Standard**	**Relevance**
2006/42/EC	Machinery Directive	
2009/104/EC	Use of Work equipment Directive	
89/654/EC	Workplace Directive	
2001/95/EC	Product Safety Directive	
2006/95/EC	Low Voltage Directive (LVD)	
2004/108/EC	Electromagnetic compatibility Directive (EMC)	

sensors, risk-assessments, etc., none of these works (to the authors knowledge) have a generalized framework which can extensively and expansively verify robotic actions and behaviours in accordance with the standards.

3

Property-Based Testing

This chapter serves as a brief recap for the property-based testing framework which is a core part of this thesis. All information within this chapter has been extracted from [1]. It should be noted that most of the presented framework has been changed and the changes will be explained in the upcoming chapters.

3.1 Overview

Property-based testing is a software validation framework that verifies whether certain designated specifications are up to par. This is conducted by feeding the property-based testing framework with random synthesized input within a certain parameter range and then validating the output in simulation by comparison with the expected output in a given tolerance range.

3.2 Framework

The property-based-testing framework was developed to answer the question of ”How to test robot behaviors in varied executions scenarios?”

The proposed framework for solving the research question was by means of property-based testing in which we generate a set of simulated scenarios in which a robot performs a task and assesses its actions by using pre-defined property-based tests as seen in Fig. 3.1. The cycle starts from a test configuration in which scenarios are defined and generated which are then fed into a property-based test suite. The property-based test suite then utilizes both an action ontology and a test configuration to set up a variety of tests which it verifies via the robot architecture and robot middle ware. This cycle may be repeated indefinitely. At the end of the cycle, a report is generated that contains the results for all the tests.

3.3 Property-Based Test Suite

In the property-based test suite, tests are designed around a system’s properties and it is ascertained whether those properties are satisfied or not. It has four steps:

1. Modeling the system and its expected behavior
2. Determining the allowed range of input parameters for the tested components
3. Generating input parameters

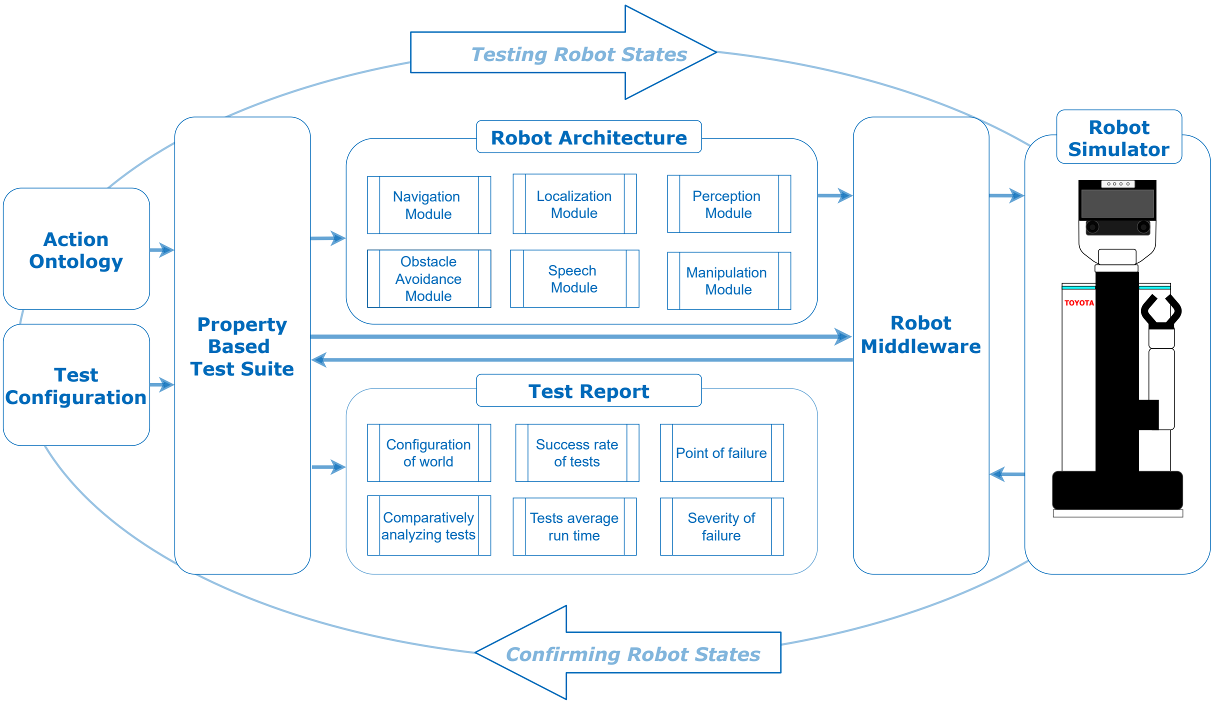


Figure 3.1: Property-based testing framework. Figure by Sohail [1]

4. Validating the expected behavior of the system

The implementation is based on the hypothesis testing framework [31].

3.4 Scenario generation

In test scenario generation, a variety of scenarios are generated by spawning objects in different locations using the Monte Carlo method.

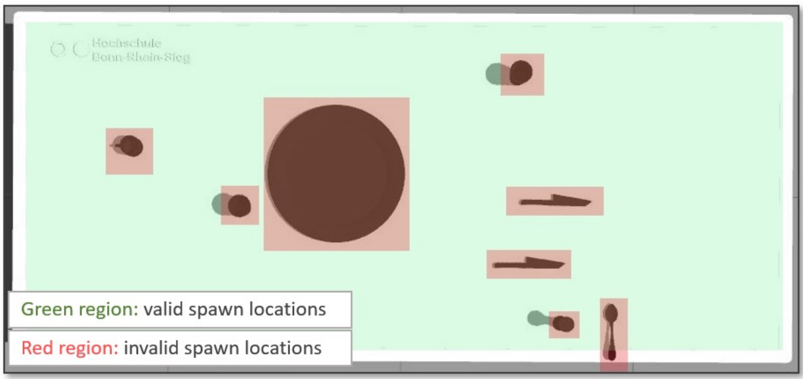

Hochschule Bonn-Rhein-Sieg
Green region: valid spawn locations
Red region: invalid spawn locations

Figure 3.2: Axis-aligned bounding box method for placement of objects. Figure by Sohail [1]

Each object has been manually designed allowing to introduce varied aspects to the objects, such as changed size, shape, mass, inertia, etc. To ensure that items do not spawn on top of each other due to randomness, the axis aligned bounding box approach is utilized which is a fast and efficient method of determining valid spawn locations. An example of the scenario generation using the axis aligned bounding box method is given in the presented figure.

3.5 Parameterization

To add more control and flexibility to the scenario generation, parameterization was enabled for scenarios via a test scenario configuration. The presented table 3.1 shows a subset of the parameters that can be configured for scenario generation. Through these, we are able to determine the number of objects that could be spawned, the type of objects, the world environment, the type of scenario, constraints, so on and so forth. The test scenario configuration narrows down our test domain to what is required, making it feasible enough to perform exhaustive flexible qualitative testing. More parameters may be added depending on the type of robot or environment to be used.

Parameter	**Description**
tests	List of tests to execute; each test is specified as a list of actions
test_count	Number of times to run each test
test_launcher	Path to a launch file that starts all components required by the tests
model_dir	Path to a directory of 3D models used in the tests
worlds	List of possible environments in which a test scenario can take place
model_list	Object models that can be used for manipulation
nav_obstacle_list	Object models that can be placed as navigation obstacles
nav_obstacle_count	Number of navigation obstacles to place in the environment
location_list	Names of locations to which the robot can navigate
object_surfaces	List of surfaces on which objects can be placed for manipulation
place_object_surfaces	List of possible surfaces to which objects can be brought

Table 3.1: Scenario configuration parameters. Table from [1]

3.6 Action Ontology

In the action ontology, an association is defined between a robot's action and the properties that define the action's success. Each of these associations are modelled using the web ontology language OWL.

We define various properties in the ontology such as:

- **performedIn** relates an action of a robot to the coordinate frame in which the action is performed
- **successProperty** describes the properties to be tested within a given action
- **hasParameter** and **needsParameter** are the requirements which have been pre-defined for an action and a property respectively both of which are based on the context of an action.

3.7 Test Report Generation

In the test report generation, we store all the important data including the test scenario configuration in an HTML format. Additional details that are stored are the description of the tests, consistency, world properties, and various other parameters. The framework used for this is the Allure framework [32].

4

Methodology

The proposed project will develop a formalization process using a domain specific language (DSL) (i.e. property-based language generator in this case) for translating existing standards into properties that will be verified by the property-based testing framework [1]. The objective is to bridge the gap of standard compliance testing and automation. The following sections will enumerate the objectives and their descriptions.

We begin by re-visiting and modifying some of the existing modules of the property-based testing framework, namely, the scenario-generation as well as the action-tests and the complex-scenario tests. In scenario generation, a re-evaluation is performed on the simulators to assess the feasibility of new upcoming simulators such as Omniverse's NVIDIA Issac Sim. The reason for this is to address on of the deficits of simulation and reality gap. More on this will be discussed in the upcoming sections. For action-tests and complex-scenarios tests, a shift is made from user oriented property-definition to innate modularized property-definitions that serve as a building block. This building block is configured via a DSL into a concrete verifiable test.

Several selected standards will then have their evaluated regulations formalized into implementable test-cases after which these test-cases will then be verified in a use-case.

4.1 Scenario Generation

Scenario generation is a core part of enabling automation in testing robotic actions and robotic behaviors in the property-based testing framework. In order to achieve more accurate and optimized simulations, a new simulator is investigated which has been developed by NVIDIA known as Issac Sim. Other simulators such as Webots, V-Rep, USARSim, and Morse will not be discussed as they have already been covered in [1].

4.1.0 Gazebo

As a refresher for the currently used scenario generation tool previously discussed in [1], Gazebo [10] is an open-source simulation tool for three-dimensional multi-agent robots (2004). The sim-tool offers good control of information flow of robots and objects that are placed within it along with information visualizations, environment design, simplicity, and physical property assignments.

It can import 3D objects in a variety of file formats, including .dae and .stl, the ability to alter a scene in an XML file, and it has a strong community support. A few of the capabilities that are required but Gazebo lacks include the ability to manipulate a 3D object meshes. In terms of dis-advantages for Gazebo, it is necessary for considerable model preprocessing prior to import, it has frequent crashes, and is quite slow in performance [33].

Figure 4.1: Gazebo robotic simulator [10]

4.1.0 Nvidia Issac Sim

4.2 Re-formalization: Property-Based Testing

The reason for the need of re-formalization of the property-definition in the property-based testing framework is to provide more clarity, structure and grounding to the properties as well as to enhance the framework itself.

Properties in the old property-based testing framework [1] are defined as a core attribute of an action (e.g.). These properties are re-defined and categorized into two groups:

1. Primitive properties

2. Composite properties

4.2.1 Primitive properties

- These properties are the base building block of all the testing behaviours.

- They serves as a blueprint that can be implemented by the composite property.

- They contains base information of an entity in the simulation.

- Example of primitive properties are:

  - Spatial-property: Entity’s position in the simulation in a given time step.
  - Physical-property: Entity’s dimensions and physical information such as mass and inertia.

4.2.2 Composite properties

- These properties use primitive properties to build a relation for a modelled behaviour.

- They are instantiated by the DSL which provides the expected modelled behaviour.

- An example of a composite property is the *collision-property.* The collision-property compares the *primitive: spatial-property* and *primitive: physical-property* of two objects i.e. **object A** and **object B** throughout a given scenario. To implement the collision-property, we use the DSL.

  robot **must not collide** table

- In this example collision-property compares the *primitive: spatial-property* and *primitive: physical-property* of **object A:robot** to **object B:table** to decipher whether the two objects have been in contact through out the test scenario using the Axis-aligned bounding box method and returns a result.

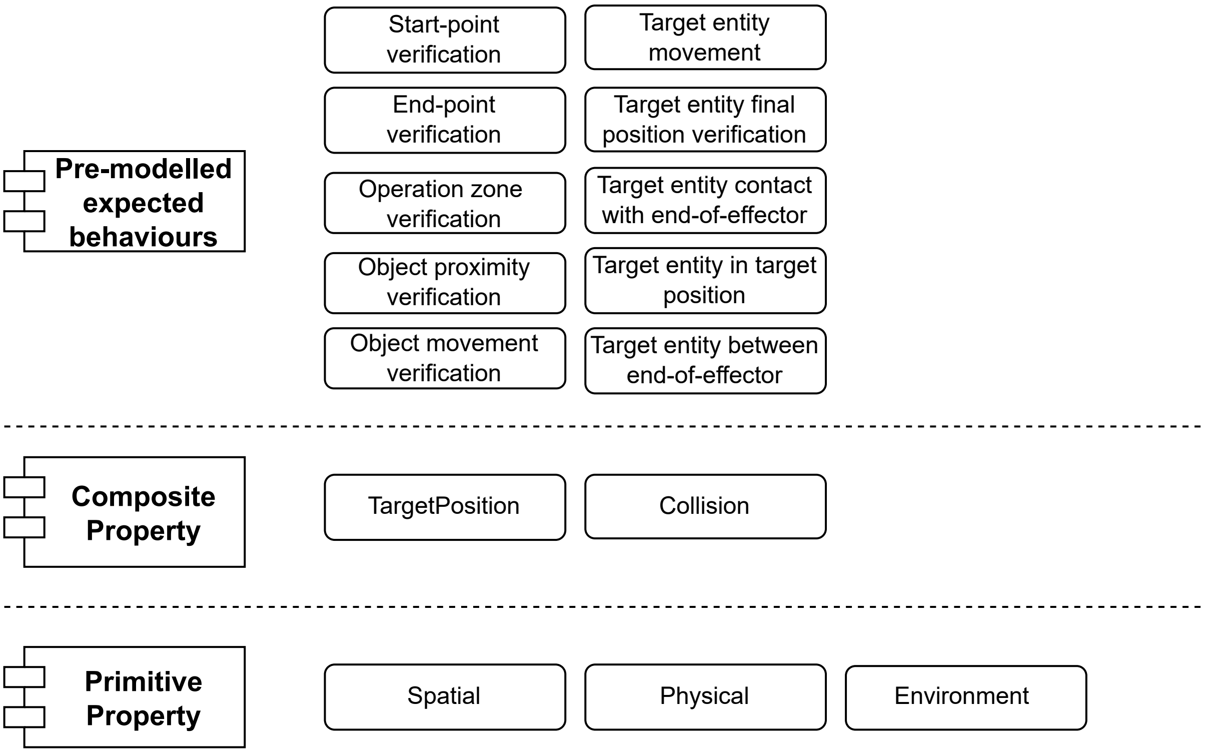

Pre-modelled expected behaviours
Start-point verification
Target entity movement
End-point verification
Target entity final position verification
Operation zone verification
Target entity contact with end-of-effector
Object proximity verification
Target entity in target position
Object movement verification
Target entity between end-of-effector
Composite Property
TargetPosition
Collision
Primitive Property
Spatial
Physical
Environment

Figure 4.2: Current pool of available primitive and composite properties as well as a set of pre-modelled behaviours.

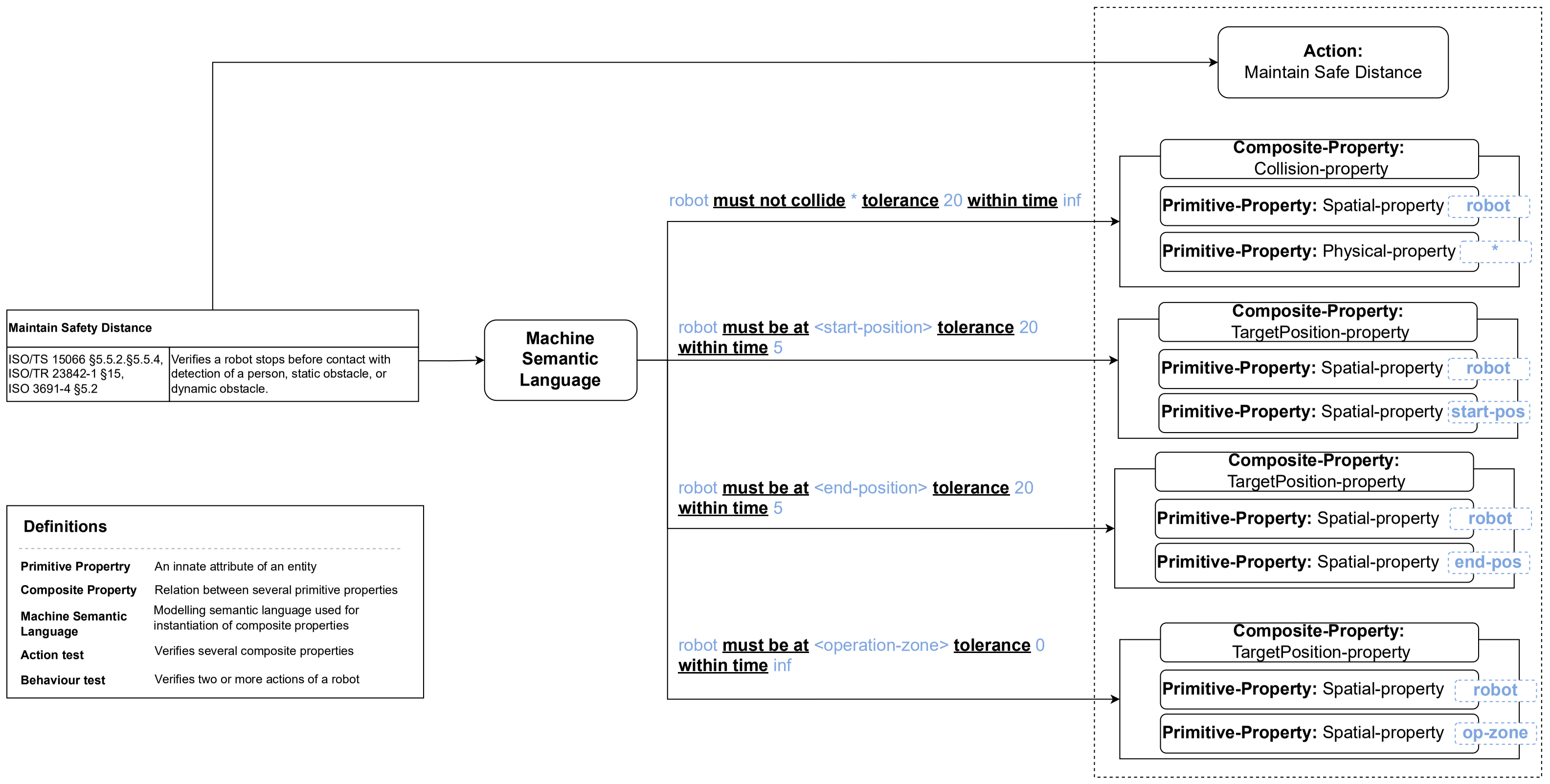

Maintain Safety Distance
ISO/TS 15066 §5.5.2.§5.5.4, ISO/TR 23842-1 §15, ISO 3691-4 §5.2
Verifies a robot stops before contact with detection of a person, static obstacle, or dynamic obstacle.
Machine Semantic Language
robot must not collide * tolerance 20 within time inf
robot must be at <start-position> tolerance 20 within time 5
robot must be at <end-position> tolerance 20 within time 5
robot must be at <operation-zone> tolerance 0 within time inf
Action: Maintain Safe Distance
Composite-Property: Collision-property
Primitive-Property: Spatial-property
robot
Primitive-Property: Physical-property
*
Composite-Property: TargetPosition-property
Primitive-Property: Spatial-property
robot
Primitive-Property: Spatial-property
start-pos
Composite-Property: TargetPosition-property
Primitive-Property: Spatial-property
robot
Primitive-Property: Spatial-property
end-pos
Composite-Property: TargetPosition-property
Primitive-Property: Spatial-property
robot
Primitive-Property: Spatial-property
op-zone
Definitions
Primitive Proeprty
An innate attribute of an entity
Composite Property
Relation between several primitive properties
Machine Semantic Language
Modelling semantic language used for instantiation of composite properties
Action test
Verifies several composite properties
Behaviour test
Verifies two or more actions of a robot

Figure 4.3: This diagram illustrates a preliminary framework for the formulation of property-based tests from a given test. The flow-chart represents the framework and the flow-chart on the right is an example of applying the framework. The tools required for implementing the properties.

4.3 Domain Specific Language: Property-Based Language Generator (PBLG)

We will define and develop a DSL in which we restructure the standard definitions into a form that retains the original information conveyed by the standard as well as being able to be interpreted by the property-based testing framework. This DSL will act as an intermediary step in the automation of standard compliance testing and will be used to model the expected behaviour in property-based testing. The working of DSL will be similar to behavior driven development [34] and will utilize a specialized syntax textX [35]. An example of the DSL is provided in the overview Fig. 4.3

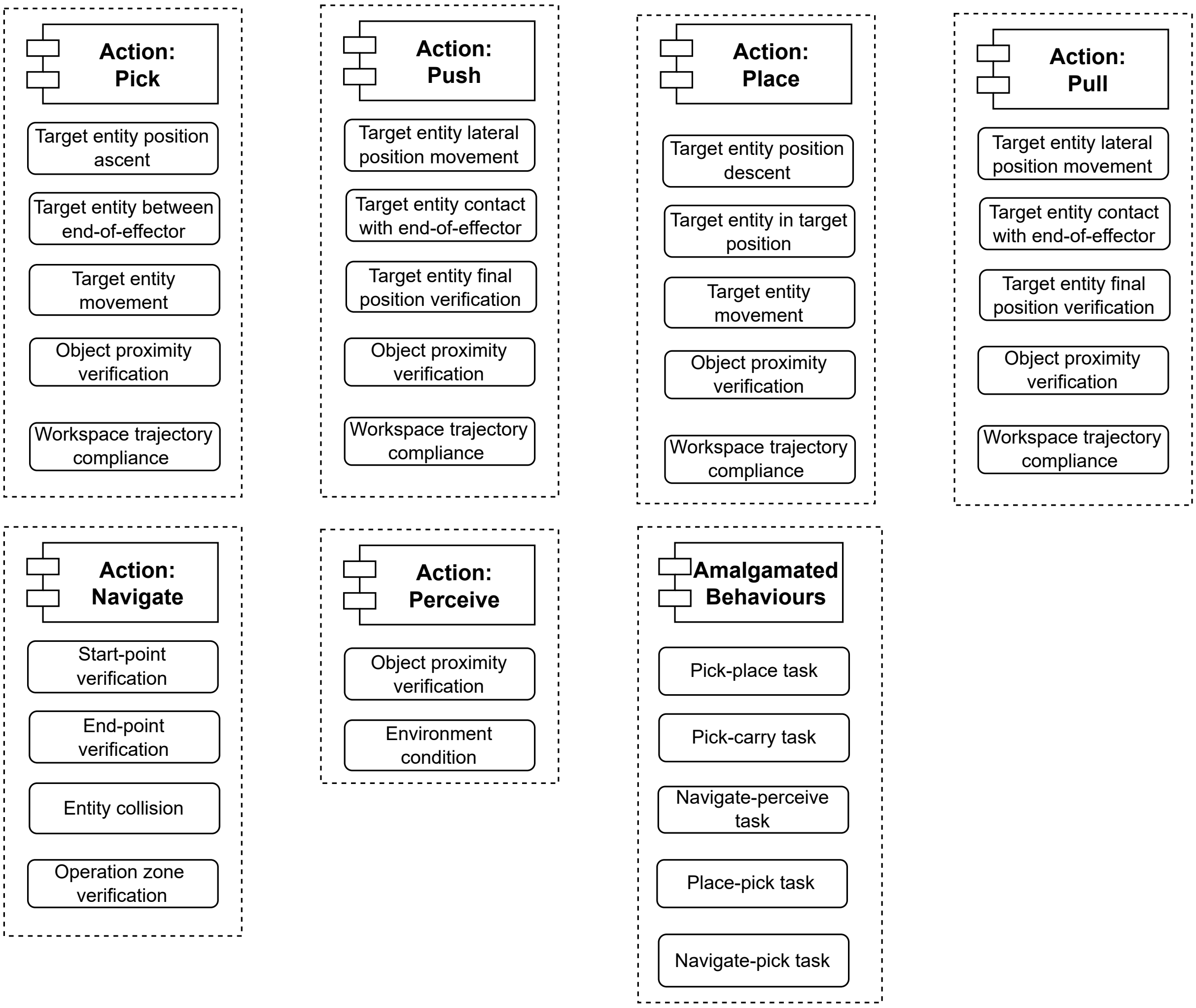

Action: Pick
Target entity position ascent
Target entity between end-of-effector
Target entity movement
Object proximity verification
Workspace trajectory compliance
Action: Push
Target entity lateral position movement
Target entity contact with end-of-effector
Target entity final position verification
Object proximity verification
Workspace trajectory compliance
Action: Place
Target entity position descent
Target entity in target position
Target entity movement
Object proximity verification
Workspace trajectory compliance
Action: Pull
Target entity lateral position movement
Target entity contact with end-of-effector
Target entity final position verification
Object proximity verification
Workspace trajectory compliance
Action: Navigate
Start-point verification
End-point verification
Entity collision
Operation zone verification
Action: Perceive
Object proximity verification
Environment condition
Amalgamated Behaviours
Pick-place task
Pick-carry task
Navigate-perceive task
Place-pick task
Navigate-pick task

Figure 4.4: This diagram illustrates a set of pre-defined modelled behaviours for several actions from the property-based testing frame work which will be used as the groundwork for the development of the DSL.

4.4 Assets Configuration

4.5 Standard Tests

4.5.1 ISO/TR 23482-1

ISO/TR 23482-1 [5] is an informative report that elaborates on the testing of safety methods in ISO 13482 [21] for personal and service robots. It provides guidelines on the design and utilization of safety test-cases. All the tests provided in this technical report cater to real-life testing.

As our proposed testing framework is property-based testing which leverages simulation for verification and validation, we shall be selecting those standards that can be verified via simulation. The table 4.2 shows which safety-tests have been selected. As the ISO 23482-1 is designed for real-life use, several of its safety-related tests criteria cannot be replicated in simulation due to the selection of our simulation platform (i.e. Gazebo) as well as the current simulation technology limitations. An example of this §14 which states the testing of the safety-control in the event off electric discharge when integrating the electro-sensitive protective equipment (ESPE). Currently, the only feasible way to perform these safety-checks is to create dedicated simulation software for these niche cases.

Test Type	**Robot**	**Selected**	**Reason**
Physical Hazard (Voltage) §6.1	Universal	X	Simulation limitation. The current robotic simulators are unable to measure the effect of electrical disturbances on the robot.
Physical Hazard (Acoustics) §6.2	Universal	X	Simulation limitation
Physical Hazard (Surface Temp.) §6.3	Universal	X	Simulation limitation
Physical Hazard (Collision-Injury Parameters) §7.1	Mobile Robot	X	Simulation limitation
Physical Hazard (Collision-Force Control) §7.2	Mobile Robot	X	Simulation limitation
Physical Hazard (Restraint Type) §8	Physical Assistant Robot	X	Out of scope
Endurance (Temp. and Humid. Fluctuations) §9.1	Universal	X	Out of scope
Endurance (Locomotion) §9.2	Mobile Robot	✓	.
Endurance (Collision) §10	Mobile Robot	✓	.
Static Stability §11	.	✓	.
Dynamic Stability (Moving parts) §12	Mobile Robot	✓	.
Dynamic Stability (Travel) §13.1	Mobile Robot	✓	.

Dynamic Stability (Travel-flat surface) §13.2	Mobile Robot	✓	.
Dynamic Stability (Travel-inclined surface) §13.3	Mobile Robot	✓	.
Dynamic Stability (Travel- steps and gaps) §13.4	Mobile Robot	✓	.
Safety-Control (Electro-Sensitive Protective Equipment) §14.1	Universal	*X*	Simulation limitation
Safety-Control (Slippery Environment) §14.2	Universal	✓	.
Safety-Control (Electromagnetic Immunity) §14.3	Universal	*X*	Simulation limitation
Response to ground obstacles (Protective stop) §15.1	Mobile Robot	✓	.
Response to ground obstacles (Distance and speed) §15.2	Mobile Robot	✓	.
Response to ground obstacles (Stopping before convex terrain) §15.3	Mobile Robot	✓	.
Response to ground obstacles (Stopping before concave terrain) §15.4	Mobile Robot	✓	.
Safety-related (Localization and Navigation) §16	Mobile Robot	✓	.
Reliability (Autonomous decision making) §17	Universal	✓	.
Safety-operation (Connection/Disconnection/Reconnection) §18.1	Universal	✓	.
Response (Multi-command/Unintended command) §18.2	Universal	✓	.
Safety-operation (Communication loss) §18.3	Universal	✓	.

Table 4.2: Selected safety-tests from ISO/TR 23482-1 [5].

4.5.1 ISO/TR 23482-1 Test Parameters

The following parameters are to be assumed when applying the safety-case defined by the standard unless stated otherwise.

- Temperature: $10^\circ - 30^\circ$
- Humidity: $0\% - 80\%$

Table 4.1: Description of failure detecting mechanisms using identified standards. Information is extracted from *Validating Safety in Human–Robot Collaboration: Standards and New Perspectives* [3].

#	**Standard**	**Description**
1.	Maintain Safety Distance	
	ISO/TS 15066 §5.5.2, §5.5.4, ISO/TR 23842-1 §15, ISO 3691-4 §5.2	Verifies a robot stops before contact with detection of a person, static obstacles, or dynamic obstacle.
2.	Static Stability	
	ISO 3691-4:2020 §5.3, ISO/TR 23482-1:2020 §11, §13	Verifies a robot's stability when carrying an object on an inclined surface in all types of positions and orientations.
3.	Dynamic Stability	
	ISO/TR 23482-1:2020 §12	Verifies a robot's stability when carrying an object that has variable load on an inclined surface.
4.	Limit Range of Movement	
	ISO 10218-1 §5.6, §5.12	Verifies that a robot does not move faster than rated speed nor does it enter a restricted space.
5.	Person Detection	
	ISO 3691-4:2020 §5.2	Verifies a robot stops before contact with a cylindrical test piece at different locations with different orientations. Requires at least three iterations.
6.	Response to Ground Obstacles	
	ISO/TR 23482-1:2020 §15	Verifies a robot's stopping distance and time given a obstacles in different positions and orientations.
7.	Safety of Localization and Navigation Stack	
	ISO/TR 23482-1:2020 §16	Verifies a robot's movement in a given point-to-point navigation task, e.g. sporadic movement, irregular stops, or potentially hazardous path planning.
8.	Robustness of Autonomous Actions	
	ISO/TR 23482-1:2020 §17	Verifies robustness of a robot's decision making, specifically the course of action during object detection.
9.	Speed Rating	
	ISO 18646-1:2016 §5	Verifies if a robot moves at a *rated speed* with acceleration, de-acceleration, and constant speed.
10.	Stop Rating	
	ISO 18646-1:2016 §6	Verifies a robot's stopping distance and time when moving in a given *rated speed.*
11.	Obstacle Detection	
	ISO 18646-2:2019 §6	Verifies if a robot can detect six static objects with different orientations at the minimum and maximum range detection range declared by the manufacturer with respect to its line of sight.
12.	Obstacle Avoidance	
	ISO 18646-2:2019 §7	Verifies if a robot can avoid dynamic obstacles with varying trajectories while navigating towards a target position.

- Surface Friction Coefficient: 0.75 – 1.00

4.6 Tools & Environments

▷ Communication software

  – ROS Noetic

▷ Simulation

  – Gazebo

  – Nvidia ISSAC sim (To be investigated)

▷ Testing software

  – Hypothesis

  – TextX

  – Allure

▷ Test platform (Dependent on available resources)

  – Clearpath Husky

  – uFactory xARM6

4.7 ...

How you are planning to test/compare/evaluate your research. Criteria used.

4.8 Setup

4.9 Experimental Design

5

Solution

Your main contributions go here

5.1 Proposed algorithm

5.2 Implementation details

5.2.1 Property-Based Testing: Augments

We will re-design and implement property-based test generator to allow compatibility with different robots as well as update the interfaces and data extraction to work with DSL. This task involves the process of integrating, developing, as well testing software planners and algorithms for the robots to have a minimum viable working use-case. Some planners may have to be developed from scratch in case of complications in integration. Expected software packages to be utilized are:

- Navigation and localization (Planners are available)
- Person detection (Custom planner will be written)
- Dynamic object detection (Custom planner will be written)
- New 3D assets to be configured for ISO tests

6

Evaluation

6.1 Evaluation

Evaluation will be performed by applying the newly designed framework on initially standard definitions for mobile robots in which we verify the completeness of the tests to capture failed scenarios. If time permits we will evaluate the frame work on some manipulators as well. A human will then take this set of test use-case and evaluate whether results from the framework adheres to the standard or not.

7

Results

.

7.1 Usecase 1: Autonomous Mobile Robots

- The **Clearpath Husky** will be used as our initial test platform and we will use it for testing AMR related standards present in the literature with the new developed property-based testing framework.
- For task execution, we will provide a simple action of point-to-point navigation and verify the standard definitions on it.

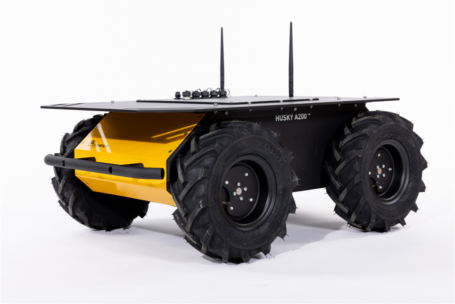


Figure 7.1: Clearpath Husky

.

7.2 Usecase 2: Robotic Manipulators

- Provided time permits we will use the **Ufactory xARM6** as our second use-case in which we will attempt to verify Cobot standards that are present in the literature.
- For task execution, we will provide two simple actions of pick-action and place-action.

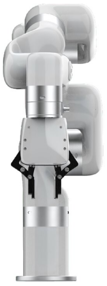


Figure 7.2: Ufactory xARM6

8
Conclusions

8.1 Contributions

8.2 Lessons learned

8.3 Future work

A

Definitions

Machinery Directive: Robots [**3**] European Machinery Directive 2006/24/EC [36] defines a robot as a multi-linked component system being actuated by a drive system. This type of robot is considered to be partly completed machinery and is deemed as complete machinery only when it is fully integrated into its designated application.

Technical Committee (TC) [**3**] Technical committees are in-charge of development of the soft standards that do not require mandatory compliance. An example is the ISO TC299 which defines terms pertaining to Robotics.

Technical Specifications (TS) [**3**] Technical specifications inform on the current state-of-the-art work that may be included into the international standard.

Technical Reports (TR) [**3**] Technical reports are informative reports on what is perceived to be state-of-the-art in a specific topic.

B

Design Details

Your first appendix

C

Parameters

Your second chapter appendix

D

Project Setup

D.1 Software requirements

The following items are required for our project setup.

D.1.1 Operating system (OS)

- Ubuntu 20.04

D.1.2 Applications

We will be using the following applications for our software package:

- Simulation Environment - Gazebo
- Middle-ware - Robot Operating System (ROS Noetic)

D.1.3 Libraries

Libraries that have been utilized and are required for running the package.

- numpy
- pandas
- Hypothesis library
- pytest
- allure
- torch

D.1.4 Robotic Stacks

D.2 Hardware Requirements

These constitute the hardware that was used for the developed software package.

- 16 Gb RAM
- AMD Ryzen 5 5600H CPU @ 3.30GHz
- Nvidia GeForce RTX 306
- 250 Gb SSD

D.3 Parameters

References

[1] S. O. Sohail, A. Mitrevski, P. G. Plöger, and N. Hochgeschwender, “Automated Test Generation for Robot Self-Examination,” *Lecture Notes in Autonomous System*, vol. 1001, pp. 900–921, 2020.

[2] G. Michalos, S. Makris, P. Tsarouchi, T. Guasch, D. Kontovrakis, and G. Chryssolouris, “Design Considerations for Safe Human-robot Collaborative Workplace,” *Procedia CIRP*, vol. 37, pp. 248–253, 12 2015.

[3] M. Valori, A. Scibilia, I. Fassi, J. Saenz, R. Behrens, S. Herbster, C. Bidard, E. Lucet, A. Magisson, L. Schaake, J. Bessler, G. B. Prange-Lasonder, M. Kühnrich, A. B. Lassen, and K. Nielsen, “Validating safety in human–robot collaboration: Standards and new perspectives,” *Robotics*, vol. 10, no. 2, 2021.

[4] “Robots and robotic devices – Collaborative robots,” International Organization for Standardization, Geneva, CH, Standard, Mar. 2016.

[5] “Application of ISO 13482—part 1: safety-related test methods,” International Organization for Standardization, Geneva, CH, Standard, Mar. 2020.

[6] “Industrial trucks—safety requirements and verification—part 4: driverless industrial trucks and their systems,” International Organization for Standardization, Geneva, CH, Standard, Mar. 2020.

[7] “Medical Electrical Equipment—Part 2-78: Particular Requirements for Basic Safety and Essential Performance of Medical Robots for Rehabilitation, Assessment, Compensation or Alleviation,” International Electrotechnical Commission, Geneva, CH, Standard, Mar. 2020.

[8] “Robots and robotic devices – Safety requirements for industrial robots - Part 1: Robots,” International Organization for Standardization, Geneva, CH, Standard, Mar. 2011.

[9] P. Chemweno, L. Pintelon, and W. Decré, “Orienting safety assurance with outcomes of hazard analysis and risk assessment: A review of the ISO 15066 standard for collaborative robot systems,” *Safety Science*, vol. 129, p. 104832, 09 2020.

[10] N. Koenig and A. Howard, “Design and use paradigms for Gazebo, an open-source multi-robot simulator,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.

[11] R. Siegwart, I. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. The MIT Press, 2011.

[12] C. Heer, *Industrial Robots: Robot Investment Reaches Record 16.5 billion USD*, ser. The new World Robotics, 2019. [Online]. Available: https://ifr.org/ifr-press-releases/news/robot-investment-reaches-record-16.5-billion-usd

[13] N. Kalra and S. M. Paddock, “Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?” *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0965856416302129

[14] W. Wachenfeld and H. Winner, *The Release of Autonomous Vehicles.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 425–449. [Online]. Available: https://doi.org/10.1007/978-3-662-48847-8_21

[15] V. D. Cosmo, A. Giusti, R. Vidoni, M. Riedl, and D. T. Matt, “Collaborative Robotics Safety Control Application Using Dynamic Safety Zones Based on the ISO/TS 15066:2016,” in *Advances in Service and Industrial Robotics.* Cham: Springer International Publishing, 2020, pp. 430–437.

[16] “Robots and robotic devices – Safety requirements for industrial robots - Part 2: Robot systems and integration,” International Organization for Standardization, Geneva, CH, Standard, Mar. 2011.

[17] J. Guiochet, Q. A. Do Hoang, M. Kaâniche, and D. Powell, “Applying Existing Standards to a Medical Rehabilitation Robot: Limits and Challenges,” in *Workshop FW5: Safety in Human-Robot Coexistence & Interaction: How can Standardization and Research benefit from each other?, IEEE/RSJ Intern. Conference Intelligent Robots and Systems (IROS2012)*, Vilamoura, Portugal, Oct. 2012. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01282195

[18] G. Herrmann and C. Melhuish, “Towards Safety in Human Robot Interaction,” *I. J. Social Robotics*, vol. 2, pp. 217–219, 09 2010.

[19] C. Harper and G. Virk, “Towards the Development of International Safety Standards for Human Robot Interaction,” *I. J. Social Robotics*, vol. 2, pp. 229–234, 09 2010.

[20] Y. H. Weng, G. Virk, and S. Yang, “The Safety for Human-Robot Co-Existing: On New ISO 13482 Safety Standard for Service Robotn,” *Internet Law Review*, vol. 17, 2015.

[21] “Robots and robotic devices — Safety requirements for personal care robots,” International Organization for Standardization, Geneva, CH, Standard, Mar. 2014.

[22] V. E. F. and J. G. A., “Robots, standards and the law: Rivalries between private standards and public policymaking for robot governance,” *Computer Law & Security Review*, vol. 35, no. 2, pp. 129–144, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0267364918302863

[23] “Safety aspects — Guidelines for their inclusion in standards,” International Organization for Standardization, Geneva, CH, Standard, Mar. 2014.

[24] “Safety of machinery — General principles for design — Risk assessment and risk reduction,” International Organization for Standardization, Geneva, CH, Standard, Mar. 2010.

[25] “Safety of machinery—two-hand control devices—principles for design and selection,” International Organization for Standardization, Geneva, CH, Standard, Mar. 2019.

[26] "Safety of machinery—positioning of safeguards with respect to the approach speeds of parts of the human body," International Organization for Standardization, Geneva, CH, Standard, Mar. 2010.

[27] "Agricultural machinery and tractors—safety of highly automated machinery," International Organization for Standardization, Geneva, CH, Standard, Mar. 2018.

[28] "Robotics—safety design for industrial robot systems—part 1: end-effectors," International Organization for Standardization, Geneva, CH, Standard, Mar. 2018.

[29] "Robotics—safety requirements for industrial robots—part 2: manual load/unload stations," International Organization for Standardization, Geneva, CH, Standard, Mar. 2017.

[30] V. Estivill-Castro, R. Hexel, and C. Lusty, "Continuous integration for testing full robotic behaviours in a GUI-stripped simulation," in *CEUR Workshop Proceedings*, vol. 2245, 2018, pp. 453–464.

[31] D. MacIver, Z. Hatfield-Dodds, and M. Contributors, "Hypothesis: A new approach to property-based testing," *Journal of Open Source Software*, vol. 4, p. 1891, November 2019.

[32] D. Baev, "Allure Framework," 2014. [Online]. Available: https://docs.qameta.io/allure/

[33] L. . Pitonakova, M. Giuliani, A. Pipe, and A. Winfield, "Feature and Performance Comparison of the V-REP, Gazebo and ARGoS Robot Simulators," February 2018.

[34] D. North. (2006, Mar.) Introducing BDD. [Online]. Available: http://dannorth.net/introducing-bdd/

[35] I. Dejanović, R. Vaderna, G. Milosavljević, and v. Vuković, "TextX: A Python tool for Domain-Specific Languages implementation," *Knowledge-Based Systems*, vol. 115, pp. 1–4, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705116304178

[36] E. Parliament, "Directive 2006/42/EC on Machinery; European Parliament: Brussels, Belgium," European Parliment, Brussels, Belgium, Standard, Mar. 2006.