



MIFTAHUR RAHMAN

A NOVEL RAILWAY MAINTENANCE ROBOT FOR INSPECTION
AND REPAIR

SCHOOL OF AEROSPACE, TRANSPORT AND
MANUFACTURING
Phd In Manufacturing

PhD
Academic Year: 2020 - 2023

Supervisor: Dr. Isidro Durazo Cardenas
Associate Supervisor: Prof. Andrew Starr
March 2023



SCHOOL OF AEROSPACE, TRANSPORT AND
MANUFACTURING
Phd In Manufacturing

PhD

Academic Year 2020 - 2023

MIFTAHUR RAHMAN

A novel railway maintenance robot for inspection and repair

Supervisor: Dr. Isidro Durazo Cardenas
Associate Supervisor: Prof. Andrew Starr
March 2023

© Cranfield University 2023. All rights reserved. No part of this
publication may be reproduced without the written permission of the
copyright owner.

ABSTRACT

Robotics and automation are widely used in various sectors for their economic benefits, accuracy, and efficiency. However, the railway industry has been slow to adopt these technologies for track inspection and repair, despite the increasing demands of asset management. Advances in robotics, computing, sensors, and Industry 5.0 are driving companies to explore innovative inspection and repair methods to optimize resource usage. Mobile manipulators, which combine robot mobility with industrial precision, have the potential to replace humans in risky and tedious tasks.

In this research, a Robotic Inspection and Repair System (RIRS) has been developed to establish an improved track inspection method and a robotic repair technique. With a 0.27% error rate in calculating defect positions and 1mm precision in actuation, RIRS demonstrates strong feasibility for railway track maintenance.

Later, an improved inspection method has been proposed fusing 3D reconstructed model of the target object from the monocular camera with the large-scale Global Positioning System (GPS) data, medium-scale environment perception from Light Detection And Ranging (LiDAR) and small-scale Colour and Depth (RGB-D) model. Texture of surface and less than 5%-dimensional error of the 3D model compared to the physical model ensures the credibility of the improved inspection technique which provides more information of the target object and sets the foundation for a data-rich digital twin in the future.

Finally, the framework of both human-in-the-loop repair task and autonomous simulated repair task have been proposed. Successful delivery of the correct tool based on the detected defect upon receiving command from the human operator and automatic circular trajectory generation demonstrate the prospects of RIRS for assisting human in track repair tasks.

Keywords: Mobile manipulator, ROS, simulation, track inspection, 3D reconstruction, track repair

ACKNOWLEDGEMENTS

First, I like to thank God, the Almighty, once again for the given opportunity, strength, wisdom, and courage to progress throughout my research.

My sincere gratitude and thanks towards my PhD advisors Dr. Isidro Durazo Cardenas and Prof. Andrew Starr, from School of Aerospace, Transport and Manufacturing at Cranfield University, for their guidance, mentorship, and suggestions along the entire path towards my degree. Their door was always open whenever I ran into a trouble spot. They were always available to listen to my problems and promptly advise a way to solve them. Without their guidance, strong support and extraordinary patience, this thesis would not have been as it is now.

I also like to convey my gratitude towards Ms. Amanda Hall and Mr. Robert Anderson of Network Rail. They consistently guided me in the right direction by sharing their professional knowledge whenever I needed it.

Also, I would like to acknowledge the experts who were involved in this research project. I am gratefully indebted for their valuable comments on this work. I would also like to thank everyone at the Centre for Life-cycle Engineering and Management (CLEM), for their assistance, their contributions to my research, the time and efforts that they devoted to my growth both academically and personally.

Finally, I must express my very profound gratitude to my beloved family and friends, for providing me with encouragement and unfailing support throughout my study and the process of researching and writing this thesis.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGEMENTS.....	v
LIST OF FIGURES.....	xi
LIST OF TABLES	xvi
LIST OF EQUATIONS.....	xvii
LIST OF ABBREVIATIONS	xviii
1 INTRODUCTION.....	1
1.1 Importance of Railway	1
1.2 Definition of Maintenance, Inspection and Repair.....	3
1.3 Maintenance of Railway	5
1.3.1 Maintenance for ensuring passenger safety.....	5
1.3.2 Current railway maintenance systems.....	6
1.3.3 Future demands in railway maintenance systems.....	7
1.4 Mobile manipulator.....	8
1.5 Railway track environment from a robot's perspective.....	10
1.5.1 Track	10
1.5.2 Switch and Crossings.....	12
1.5.3 Tunnel and Bridges	13
1.5.4 Trackside Furniture	14
1.6 Use of mobile manipulator for railway track maintenance.....	14
1.6.1 Shortcomings of current railway track maintenance technology.....	14
1.6.2 Scope of mobile manipulator for track maintenance	17
1.7 AIM AND OBJECTIVES.....	21
1.7.1 Aim	21
1.7.2 Objectives	21
1.8 Overview of the thesis.....	22
2 LITERATURE REVIEW	25
2.1 Conventional railway track maintenance machines	25
2.1.1 Push-Trolley	26
2.1.2 Road-Rail Vehicle	27
2.1.3 Specialist train.....	29
2.1.4 Train-borne system	30
2.2 Robots used in railway maintenance	31
2.2.1 Robotics inspection technology in railway	31
2.2.1.1 Track inspection.....	32
2.2.1.2 Tunnel and bridge inspection.....	37
2.2.1.3 Vision based inspection method in railway	39
2.2.2 Robotic repair technology in railway.....	41
2.3 Robotic maintenance technology in other industries.....	44

2.3.1 Overview of mobile manipulator based on mobility and applicability	44
2.3.2 Mobile manipulators for inspection and repair tasks	46
2.3.2.1 Maintenance of high-voltage transmission line and substation ..	46
2.3.2.2 Maintenance in nuclear power plants & hazards	49
2.3.2.3 Maintenance in aerospace	52
2.3.2.4 Maintenance in civil and construction sites	53
2.3.2.5 Application of mobile manipulator in agriculture	54
2.3.2.6 Other applications of the mobile manipulator	55
2.4 Research Gaps	57
2.4.1 Lack of multipurpose robot	58
2.4.2 Shortcomings of current vision-based inspection method	58
2.4.3 Necessity of using robot for railway track repair tasks	59
3 METHODOLOGY	61
3.1 Research Methodology	61
3.2 RIRS: Overview	62
3.2.1 Sensors	64
3.2.2 Softwares and tools	66
3.2.3 Simulation	66
3.2.4 Coding Strategy of RIRS	67
3.3 Kinematics, repeatability, and precision	69
3.3.1 Kinematics	69
3.3.1.1 Configuration space and workspace	69
3.3.1.2 Robot structures	70
3.3.1.3 Kinematics of RIRS	71
3.3.2 Repeatability and precision	71
3.4 Inspection (Vision based sensor fusion)	76
3.4.1 Workflow	76
3.4.2 Structural model generation (SfM)	77
3.4.3 Model fusion by perception enhancement and registration	81
3.5 Repair task	82
3.5.1 Repair task with human-in-the-loop	83
3.5.2 Automatic waypoint generation framework	84
3.6 Summary	87
4 RESULTS	88
4.1 Simulation of RIRS	88
4.1.1 Simulation environment for on-track	89
4.1.2 Simulation environment for off-track	90
4.2 Kinematics, repeatability, and precision analysis of RIRS	90
4.2.1 Kinematics of RIRS	90
4.2.1.1 On-track operation	91
4.2.1.2 Off-track operation	92

4.2.2 Repeatability and precision of RIRS.....	93
4.2.2.1 Experiment and data collection procedures.....	93
4.2.2.2 Results.....	95
4.2.2.2.1 Repeatability for inspection task.....	95
4.2.2.2.2 Precision for repair motion	96
4.3 Improved inspection technique	98
4.3.1 Experiment and data collection procedures	98
4.3.2 Results	100
4.3.2.1 Object detection by depth camera	100
4.3.2.2 3D object modelling	102
4.3.2.3 Model fusion for final perception	104
4.4 Repair	106
4.4.1 Human-in-the-loop repair task.....	106
4.4.1.1 Experiment and data collection procedures.....	106
4.4.1.2 Results.....	108
4.4.2 Automatic trajectory generation.....	110
4.4.2.1 Experiment and data collection procedures.....	110
4.4.2.2 Results.....	111
5 DISCUSSION	114
5.1 General Discussion.....	114
5.2 Kinematics, repeatability, and precision analysis of RIRS	115
5.2.1 Kinematic analysis of RIRS	115
5.2.1.1 Changes in kinematics for on-track vs. off-track configuration.	115
5.2.1.2 Benefits of Road-rail Conversion	116
5.2.2 Repeatability and Precision of RIRS	117
5.2.2.1 Inspection	117
5.2.2.2 Repair	117
5.3 Inspection (3D modelling)	118
5.3.1 Image Capturing.....	118
5.3.2 Dimensional Accuracy.....	119
5.3.3 Realistic Scenario Testing.....	120
5.3.4 Flexibility of the Proposed Method	120
5.4 Repair	123
5.4.1 Human-in-the-loop repair task.....	123
5.4.1.1 Communication and hardware architecture	123
5.4.1.2 Latency	124
5.4.1.3 GPS denied area or low accuracy of GPS.....	126
5.4.1.4 Predefined object library and manipulator poses.....	127
5.4.2 Automatic trajectory generation.....	128
6 CONCLUSION	130
6.1 Novelty: a multipurpose railway track maintenance robot.....	130
6.2 Conclusions on the specific objectives.....	131

6.2.1 Gap identification and Simulation of a RIRS	131
6.2.2 Kinematics, repeatability, and precision of RIRS.....	132
6.2.3 Improved Inspection technique	133
6.2.4 Repair tasks framework.....	134
6.3 Limitations of this research	135
6.3.1 Limitations of the demonstrator	135
6.3.2 Limitations of the inspection method	136
6.3.3 Limitations of the repair framework	136
6.4 Future recommendations	137
6.4.1 Introducing machine learning for better image processing	137
6.4.2 Using teleoperation and mixed reality	137
6.4.3 Establishing digital twin using robotic technology.....	138
REFERENCES.....	139
APPENDICES	159
Appendix A Softwares and Tools.....	159
A.1 ROS.....	159
A.2 MoveIt!	160
A.3 Gazebo	163
Appendix B System engineering approach for coding	164
B.1 Verification and Validation (V&V) model	165
Appendix C List of Publications	167
Appendix D Important Python Scripts	167
D.1 Manipulator poses	167
D.2 Code for Human-in-the-loop-repair task framework.....	178
D.3 Code for automatic trajectory generation framework	180
D.4 Code for tf transformation of object	185
Appendix E Repeatability and precision.....	186
E.1 Data table for repeatability from 10 experiments	186

LIST OF FIGURES

Figure 1.1 Typical activities included in maintenance	4
Figure 1.2 Upgraded Hilare 2bis from LAAS with 6-wheel robot and a manipulator [35]	9
Figure 1.3 Challenges for a robot based on railway infrastructure	10
Figure 1.4 Major components of a railway track	11
Figure 1.5 Components of switches and crossings	13
Figure 1.6 Railway tunnels [37, 38]	14
Figure 1.7 Railway trackside furniture, from left to right is LOC's [41], Marker post [40], Signal post [39]	15
Figure 1.8 Necessity of mobile manipulator for railway track maintenance	18
Figure 1.9 Overview of the thesis	23
Figure 2.1 Trolley for carrying material [64]	27
Figure 2.2 Track geometry inspection trolley, (a) and (b) are reproduced from [65] and [66], respectively	28
Figure 2.3 Road-rail vehicles for track maintenance, (a) Unimog road rail vehicle for overhead line maintenance [69], (b) mobile asset management RRV [70]	28
Figure 2.4 Specially designed train in the UK for railway maintenance; (a) New Measurement train (NMT) [19], (b) Mobile Maintenance Train (MMT) [23]	30
Figure 2.5 Track inspection with multiple cameras; (a) inspection trolley [81], (b) modified RRV [82]	34
Figure 2.6 Autonomous track inspection robot; (a) Railpod [89], (b) robot with ACFM sensor [88], (c) AutoScan [90], (d) RII1005 [56]	36
Figure 2.7 Felix in operation to inspect switch and crossing [91]	37
Figure 2.8 Bridge inspection robot; (a) UAV with multiple sensors [92], (b) UAV for bridge bottom inspection [93], (c) UGV for bridge bottom inspection [94]	38
Figure 2.9 Tunnel inspection robot; (a) ROBO-SPECT [74], (b) MIT tunnel inspection robot [99]	39
Figure 2.10 Robot used for repairing rail car in mining operation [116]	42
Figure 2.11 DDR technology [118]; (a) the machine, (b) cavity created after removing defected rail head, (c) rail head after arc welding, (d) final repaired rail head	43

Figure 2.12 Percentage of mobile manipulator based on mobility features	45
Figure 2.13 Classification of mobile manipulator based on area of application	46
Figure 2.14 Specially designed mobile manipulator for underground power transmission line, reproduced from [117, 118].....	47
Figure 2.15 Mobile manipulator for maintenance of power transmission line; (a) for vibration damper maintenance [122], (b) Autonomous live-line maintenance robot (ALMR) [123], (c) POSITRON [125, 126].....	48
Figure 2.16 Mobile manipulator in nuclear industry; (a) RPAR-I (Radiation Protection Assistant Robot) [63], (b) CERNbot with single and dual arm configuration [131], (c) MIRA SPS inspecting and navigating through a narrow passage [134]	50
Figure 2.17 Aerial mobile manipulator extruding PU foam to block a crack in surface [136].....	51
Figure 2.18 Mobile manipulator in space for maintenance application; (a) DAFSAR space mobile manipulator [138, 139], (b) AMODS [140]	52
Figure 2.19 Proposed mobile manipulator system for with multiple sensors [150]	54
Figure 2.20 Mobile manipulator for agricultural application; (a) strawberry picking robot [155], (b) pest detection and spraying robot for a greenhouse [160]	55
Figure 3.1 Overall methodology of this thesis.....	62
Figure 3.2 Sensors and systems of RIRS	63
Figure 3.3 RIRS performing on track navigation at RITA, Cranfield University and in simulation.....	64
Figure 3.4 Simulation tools for robotics	67
Figure 3.5 Coding strategy of RIRS using system engineering approach	68
Figure 3.6 workspace of a manipulator [204].....	72
Figure 3.7 Defect position calculation for RIRS	72
Figure 3.8 D-H parameters of manipulator links	73
Figure 3.9 Frames and joints of UR10e.....	74
Figure 3.10 Transformation between 2 coordinates frame in ROS.....	76
Figure 3.11 Workflow of vision-based sensor fusion for inspection	77
Figure 3.12 Workflow of Structure from Motion (SfM) technique	78
Figure 3.13 Projection of a point in different coordinates frame for SfM	79
Figure 3.14 Reconstruction of 3D points using triangulation [214]	80

Figure 3.15 Repair task framework with human-in-the-loop	83
Figure 3.16 Repair task framework for automatic waypoint generations	85
Figure 3.17 Repair task framework for automatic waypoint generations	86
Figure 3.18 Relation between cartesian and cylindrical coordinate system	86
Figure 4.1 Simulation in Gazebo for on-track scenario.....	89
Figure 4.2 Simulation in Gazebo for off-track scenario.....	90
Figure 4.3 Construction of Warthog and specially trolley for RIRS; (a) 6 joints of warthog from cross-section, (b) design and power transmission mechanism of trolley	92
Figure 4.4 Joints of RIRS during on-track navigation; (a) all joints of RIRS, (b) joint configuration of RIRS with 2 simplified joints	93
Figure 4.5 Joints of RIRS during off-track navigation	94
Figure 4.6 QRcode for inspection task, (a) QRcode on the track, (b) output after detection	95
Figure 4.7 Error investigation for inspection job, (a) calculated and actual defect position from initial RIRS position, (b) error in every test	96
Figure 4.8 Maximum and minimum position of UR10e manipulator in 3D coordinate system	97
Figure 4.9 Pose error of UR10e from 10 repetitive repair motions (a) without payload, (b) with 1.75kg payload, (c) with 4.1kg payload, (d) with 5.9kg payload	98
Figure 4.10 Experimental setup (a) inside lab, (b) RITA railway track test scenario	99
Figure 4.11 Dimensions of objects (a) indoor target object (bottle), (b) wooden pallet, (c) outdoor target object (rail section), (d) sleepers in standard railway track.....	100
Figure 4.12 Perception of the environment from RGB-D camera in (a) indoor lab	101
Figure 4.13 Front and side view of 3D model of the bottle (a) front and (b) side view and rail section (a) front and (b) side view	102
Figure 4.14 Relationship between reconstruction time and number of images	103
Figure 4.15 Dense 3D model of the bottle (a) front and (b) side view and (c) front and (d) side view generated by SfM technique using RGB camera and manipulator.....	103

Figure 4.16 Wrong scaling and alignment of 3D reconstructed model generated from SfM	104
Figure 4.17 Wrong scaling and alignment of 3D reconstructed model generated from SfM	105
Figure 4.18 Fusion of rail model after aligning and scaling (a) front view (b) side view	105
Figure 4.19 GUI of HMI	106
Figure 4.20 Experiment environment for human-in-the-loop repair task.....	107
Figure 4.21 RIRS carrying 3 different toolbox.....	108
Figure 4.22 Cycle time for human-in-the-loop repair task.....	109
Figure 4.23 Cycle time for only payload delivery	110
Figure 4.24 Automatic waypoint generation based on detected object	111
Figure 4.25 Different object for waypoint generation	112
Figure 4.26 Automatic waypoint generation based on detected object	113
Figure 5.1 Manipulator poses for capturing images for data-rich 3D modelling technique	118
Figure 5.2 Improved RIRS perception with the fused model of maintenance object; (a) RIRS position in global map acquired from GNSS, (b) RIRS perception from RGB-D camera with the surrounding environment from LiDAR, (c) fused 3D model of object in robot perception	121
Figure 5.3 3D reconstruction of railway track using SfM with RIRS.....	122
Figure 5.4 Communication and hardware architecture of HMI with computational devices	123
Figure 5.5 Different object for waypoint generation	128
Figure A.1 Computation level architecture of ROS [190].....	159
Figure A.1 Computation level architecture of ROS [190].....	159
Figure A.2 MoveIt! architecture [192]	161
Figure A.2 MoveIt! architecture [192]	161
Figure A.3 Difference between pose goal and joint goal	162
Figure A.3 Difference between pose goal and joint goal	162
Figure A.4 Gazebo and ROS integration [201]	164
Figure A.4 Gazebo and ROS integration [201]	164
Figure B.1 Generic V-model of system development [202]	165

Figure B.1 Generic V-model of system development [202]	165
Figure B.2 Detailed verification and validation model for software development [203].....	166

LIST OF TABLES

Table 1.1 Structure gauges (clearance between railway trackside infrastructure)	12
Table 2.1 Type of track maintenance platforms used in railway	26
Table 2.2 Track inspection robots	32
Table 2.3 Different areas of application for mobile manipulators.....	56
Table 3.1 Key specifications of RIRS components.....	63
Table 3.2 Summary of sensors in RIRS	65
Table 3.3 Summary of joints in RIRS	71
Table 3.4 Values of D-H parameters for UR10e.....	74
Table 4.1 Components of simulation in different scenarios	88
Table 5.1 Dimensional comparison between actual and 3D reconstructed model	119
Table 5.2 Average network latency and standard deviation for HMI GUI components interacting with the Warthog robot.....	125
Table 5.3 Toolbox position and type based on the simulated defect type	127
Table E 1 Experimental data from repeatability tests	188

LIST OF EQUATIONS

(3-1)..... 72

(3-2)..... 73

(3-3)..... 75

(3-4)..... 75

(3-5)..... 75

(3-6)..... 79

(3-7)..... 80

(3-8)..... 81

(3-9)..... 81

(3-10)..... 81

(3-11)..... 81

(3-12)..... 82

(3-13)..... 82

(3-14)..... 82

(3-15)..... 86

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
BA	Bundle Adjustment
BCS	Ballast Cleaner System
CP6	Control Plan 6
CTIV	Comprehensive Track Inspection Vehicle
DART	Diagnostic Analysis for Railways and Trams
DIC	Digital Image Correlation
DLT	Direct Linear Transformation
DoF	Degree of Freedom
EKF	Extended Kalman Filter
EMAT	ElectroMagnetic Acoustic Transducer
FK	Forward Kinematics
GPR	Ground Penetrating Radar
GPS	Global Positioning System
GPS	Global Positioning System
GUI	Graphical User Interface
HITL	Human-in-the-loop
HMI	Human Machine Interface
HSE	Health and Safety Executive
IK	Inverse Kinematics
IMU	Inertial Measurement Unit

InSAR	Interferometric Synthetic Aperture Radar
LiDAR	Light Detection And Ranging
LOC	LOcation Cases
LVDT	Linear Variable Differential Transformer
MMT	Mobile Maintenance Train
MPV	Multipurpose Vehicles
NDT	Non-Destructive Testing
NMT	New Measurement Train
ODE	Open Dynamics Engine
OEM	Original Equipment Manufacturer
OLE	Overhead Line Equipment
OSRF	Open-Source Robotics Foundation
PLPR	Plain Line Pattern Recognition
PnP	Perspective-n-Point
PTZ	Pan-Tilt-Zoom
RAS	Robotic and Autonomous System
RGB-D	Colour and Depth
RIRS	Robotic Inspection and Repair System
RISSB	Rail Industry Safety and Standards Board
RITA	Railway Innovation and Testing Area
ROS	Robot Operating System
RRV	Road-Rail Vehicle

RSSB	Rail Safety and Standard Board
S&C	Switches and Crossings
SDF	Simulation Description Format
SDG	Sustainable Development Goals
SfM	Structure-from-Motion
SITT	Snow and Ice Treatment Trains
SLAM	Simultaneous Localization and Mapping
TRS	Track Renewal System
URDF	Universal Robotic Description Format
ViSP	Visual Servoing Platform
VISyR	Visual Inspection System for Railway
XML	Extensible Markup Language

1 INTRODUCTION

The railway is a standard mode of mass transportation around the world. It is serving civilization to grow and prosper starting from the 15th century. It is safer, cheaper, and, most importantly, greener than other modes of the mass transportation. Maintenance of railway infrastructure is vital to ensure the sustenance of railway network. Among all the railway infrastructure, the track is one of the most important assets. Its structural integrity is being essential the safety of passenger and freight trains. Robots are used in many industries for more than multiple decades to gain efficiency and financial benefits. Robots can be useful for upgrading the current railway maintenance technique to increase asset utilization and improve human safety.

This chapter will shed some light on the importance and advantage of using robots for railway maintenance. The importance of the railway in the sustainability of society and the progress of civilization will also be discussed. Afterward, the definitions of 3 important terms, maintenance, inspection, and repair, will be provided followed by the current scenario of railway maintenance technology. Subsequently, mobile manipulator will be introduced briefly. In the next section, a brief overview of the railway infrastructure will be provided along with the unique challenges for the mobility of a robot imposed by the railway infrastructure. Afterward, deficiencies in current railway track inspection and repair technologies will be investigated. The aims and objectives of this research will be mentioned. Finally, the chapter will be finished by providing a brief overview of this whole thesis.

1.1 Importance of Railway

As one of the largest transportation assets, the railway infrastructure networks have served industry and society to rise globally for hundreds of years, acting as a cornerstone for building a resource-conserving and environment-friendly mode of transportation. The early history of the British railway goes back to 1610 when coal was transported from pithead to waterways using wagons [1]. This small invention eased the burden of humans to transport goods to nearby places.

However, with the advancement of steam engine and steel production, George Stephenson played a crucial role to introduce steam locomotives in the early 1800s [2]. The use of rails increased the production speed of the mining industry. The railway has been one of the safest and most convenient mass transportation systems for transporting people and goods. Even after 300 years, the railway accounts for the least number of fatalities from accidents. The number of fatalities for a ten-year average from 2011 to 2022 was almost zero for rail compared to other transportation systems [3].

Compared with other mass transportation systems on land and air, rail is much greener as it emits less CO₂ per traveller in the environment. Some modern trains, such as Eurostar, emit only 6g of CO₂ per traveller compared to 172g from a medium-sized petrol car. Traveling by train instead of car or airplane can reduce emissions by 80% and 84%, respectively [4]. Reducing carbon footprints using railways will help many countries achieve the climate action of Sustainable Development Goals (SDG), as transportation accounts for a large portion of carbon emissions. Apart from the environmental benefits, the railway transfers people, and goods faster than road vehicles and ships. Therefore, railway use is expected to increase in the upcoming years.

For instance, in the UK, the number of train passengers will be doubled, and freights will rise by 90% in 2041 compared to 2014 [5]. In 2018-2019, more than 1.7 billion people used the railway as their transportation mode, and 17 billion tonnes of freights were transported between places using the railway in the UK [6]. The benefited passengers can be divided into 3 main categories: commuters, leisure travellers, and business travellers. Commuters are the biggest source of passenger demand and account for almost half of the passengers [7]. At the same time, leisure and business purpose travellers accumulate the rest half. On the other hand, businesses and consumers across the UK use the freight service to fulfil their demand for moving goods from one place to another, improving the economic sector's sustainability and productivity.

All this significant contribution of the railway sector adds up to UK's economic growth. Though the construction and maintenance costs are higher than the road

transport system [8], low operating costs make the railway more profitable in the long run than other public transport system [9]. Also, the return on investment is higher than other modes of transportation. This highly demanding transportation sector employs more than 240,000 people in the UK [7]. Different franchised operators, known as Train Operating Companies (TOCs) run train services in this vast railway network in the UK, while most of the fixed assets are owned, operated, and maintained by Network Rail [10].

1.2 Definition of Maintenance, Inspection and Repair

Inspection, repair, and maintenance are 3 important concepts of asset management which will be used repeatedly in this thesis. Asset management is a concept that has quickly acquired traction in the rail sector. It is a combination of coordinated operations that optimise asset performance to accomplish a company's objectives is the core defining concept of an often-complicated system . It includes all systems, methods, and instruments for maximising asset availability while minimising total cost and risk. This frequently entails employing sophisticated software to gather and analyse consumer data to perform predictive and preventive maintenance rather than reactive fixes. Asset management encompasses an asset's lifespan, from design, building, and operation through renewal and disposal, as well as the implications of each activity [11]. Network Rail, one of the largest asset management organizations in the UK, states in its asset management strategy that the system is critical to delivering "excellent" value for taxpayers' and customers' money and supporting sustainable development [12]. When aligned with company objectives, asset management also helps to enhance safety, boost economic growth and social possibilities, and reduce environmental impact.

In this research, maintenance, inspection, and repair will be mentioned multiple times. Hence, the definition of these 3 important topics has been discussed below.

Maintenance

Maintenance refers to activities performed on assets to avoid damage and extend life expectancy. Maintenance not only include technical activities but also all administrative and managerial activities during the whole life cycle of the asset [13]. Condition monitoring, inspection, testing, diagnosis, repair, restoration, repair, upgrade and all other observation and analysis are part of technical maintenance [13]. It is frequently used as a blanket phrase for any maintenance work performed on our assets. Activities in maintenance includes many activities such as schedule inspection or functional checks, replacement of spare parts, upgrading current asset, performing repairs etc. From the definition, all the inspection and repair tasks are placed under the umbrella of maintenance, as shown in Figure 1.1.

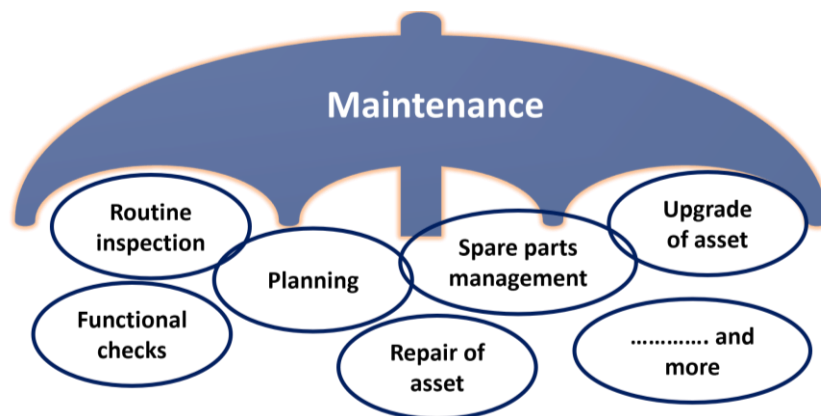


Figure 1.1 Typical activities included in maintenance

Inspection

Inspection in asset management might signify many things depending on the situation. Generally, it can refer to monitoring or inspecting assets and evaluating the data gathered to assess their status. As per the definition from British Standards, inspection is the examination of an asset for conformity by observation, measurement or testing the appropriate characteristics [13]. It may also include correlating inspections with assets to view their history and performance. Inspection is a component of asset management that balances costs, opportunities, and risks against intended asset performance to meet an organization's goals.

Repair

Repair is the process of restoring an item to its operational state. The amount of repair required relates to the extent of failure producing asset malfunction. As per maintenance terminology from British standard repair is the physical activity performed to restore the functionality of a faulty item [13]. Sometimes repair is also mentioned fault correction. There are 2 types of equipment failure which determines the urgency of repair tasks. Partial failure of an asset means the asset is still working but with lower efficiency or capacity. Repair tasks can be carried out instantly or later. On the contrary, complete failure of the asset demands instant and quick repair within the shortest time possible to reduce the loss.

1.3 Maintenance of Railway

Any equipment, infrastructure, or system deteriorates with time and usage. Maintenance is a set of tasks that are carried out to keep the equipment or system usable by sustaining the original state and protecting it from failure. With proper maintenance, the lifetime of any equipment or machine can be prolonged before it exhausts. On the other hand, the lifetime or condition of any equipment degrades faster when usage increases. Maintenance includes not only technical tasks but also includes administrative and managerial tasks [14]. As the demand for the railway is increasing in the UK, the usage of railway infrastructure is escalating, challenging the limit of the railway infrastructure asset management system. An enormous increase in passenger and freight railways demand can only be fulfilled by ensuring less track downtime and optimum usage of railway infrastructure. A railway infrastructure asset management system is concerned with planning, controlling, and maintaining all asset-related hardware, activities, and their relationships.

1.3.1 Maintenance for ensuring passenger safety

Railway track maintenance management is one of the keys to ensuring the basic demands of safe transportation and sustainable development. It is essential to provide a safe and comfortable travel experience to the passenger to gain their

trust for the next journey. Also, maintaining the announced schedule of train arrival, departure, and travel time significantly impacts customer satisfaction.

Accidents in passenger trains may cause the death of people, which is unacceptable, while accidents in freight trains will cause damage to goods which is a loss to the economy. Maintenance-related causes account for the maximum number of accidents. Of more than 600 railway-related accidents and incidents in Sweden from 1988 to 2000, 30% of the accidents happened due to a lack of track maintenance [15]. Moreover, the same study found that lack of communication and information and violations of the rules are the reason for 80% of accidents and incidents during maintenance tasks. The scenario is almost similar in a worldwide context. After analyzing 700 accidents in 23 countries, it was found that 37% of accidents happened due to rolling-stock failure, and 36% happened due to infrastructure failure [16]. Comfortable and safe rides in a designated timetable can only be ensured by sustaining the proper operating condition of railway infrastructure such as railway tracks, level crossings, switches and crossings, bridges, tunnels, etc. Any delays in the train schedule cause financial losses, and 17% of train delays are related to tracking faults such as broken rail, buckled rail for summer heat, overhead line equipment (OLE) failure, signals, point machine failure [17]. That is why almost two-thirds of employees of Network Rail are engaged in conducting the enormous maintenance task. Not only people but also a big slice of the annual budget is also consumed for maintenance.

1.3.2 Current railway maintenance systems

For decades, track maintenance was mostly human involved activities comprising more than 42,000 people engaged in Britain for maintenance-related tasks to solve the signaling malfunction, rail crossings, broken rails, detection failures [18]. Maintenance done by a human with a handheld device is slow and costly. Meanwhile, specially equipped trains such as NMT (New Measurement Train) [19], Doctor Yellow [20], China Railway Comprehensive Inspection Trains [21], etc., have only the capability to identify rail defects but require humans for onboard monitoring and driving the train.

Some maintenance tasks are carried out regularly to keep the track usable and defect-free. For example, some maintenance tasks include redistributing the ballast with the help of a tamper and ballast regulator, smoothening the top surface of the rail with a grinder, and checking the fitting measurement. In addition, many trains are equipped with various sensors which execute track inspection periodically. If any defect is detected in the track, a maintenance action is initiated in the database. Moreover, some seasonal maintenance tasks, such as painting white color to reduce expansion and buckling effect in summer, the addition of a heating device to impede ice formation in winter, or clearing leaves from the track, signals, and switches, etc., are also carried out to ensure a safe and pleasant journey [22]. For conducting inspection and maintenance task, Network Rail deploy many machines ranging from state-of-the-art Unmanned Aerial Vehicle (UAV) to heavy machines such as NMT [19], Mobile Maintenance Train (MMT) [23], Ballast Cleaner System (BCS) [24], Multipurpose Vehicles (MPV) [25] and Snow and Ice Treatment Trains (SITT) [22], Track Renewal System (TRS) [26] and many more. These machines are equipped with various ultrasonic sensors, high-definition cameras, Global Positioning System (GPS), Linear Variable Differential Transformer (LVDT), laser, gyroscope, accelerometers.

1.3.3 Future demands in railway maintenance systems

In the coming years, the passenger and freight trains demand will increase in the UK. Control Plan 7 or CP7 (April 2024 to March 2029) has been designed to accommodate the increased demand. As per the CP7 delivery plan, Network Rail will spend £44 billion for operations, maintenance, and renewal of railway network [27]. In 2019-20, Network rail spent £11 billion for renewals, maintenance, enhancements, operations and support [28]. From the £1.7 billion of maintenance related cost, around 40% of the was spent on track maintenance [28]. National and international organizations have taken initiative to reduce the financial burden and ensure optimum usage of railway network. Shift2Rail [29] is one of such program from European Union (EU). Robotics and Automation have an extensive scope in these plans, especially in track maintenance and inspection.

Many projects have already been implemented, such as improving and installing new signal systems, and more projects are under development.

It is important to find the usage of modern robotics technology for rail inspection and repair to keep pace with the increasing demand and ensure a safe and comfortable journey. The economic benefits of using a robotic system are already proven, such as robotics can provide financial advantages in rolling stock maintenance [30]. Though robotics and autonomous systems have been used in the railway sector for more than a decade, most of them are restrictively used in rolling stock maintenance. Most robotic systems are without manipulation capability [31]. Intelligent digitalization in railway maintenance is one of the objectives of the IN2SMART2 [32] projects in the European with an estimated cost of €23 million which aims to bring the AI-based robotic strategy to improve task decision support and scalable asset information management framework. To achieve this, the inspection and repair of the rail track and related jobs are crucial to the life-cycle engineering and management of track assets from the concept design to disposal. However, as the rail mileage grows promptly, the maintenance criteria are demanded more restrictively with technologies and tools improvement. Therefore, robotic maintenance systems are highly demanded by flexibly combining autonomous sensing, control, decision-making, remote communication, and work execution for inspection and repair.

1.4 Mobile manipulator

A manipulator or robotic arm is a multipurpose mechanical device programmable for various tasks. In the earlier stage, manipulators were handy tools for human workers in industries to assist humans in lifting or moving heavy tools by a predetermined sequence of motions. Manipulators are only articulated kinematic chains linked with one another with flexible joints [33] and programmed to perform some motions in a designated workspace. Based on the requirements and design, manipulators have multiple joints, also known as "Degrees-of-Freedom" or DoFs. The use of industrial manipulators can be dated back to 1954 when General Motors used Unimate, a hydraulic arm, to lift heavy loads [34]. With the advancement in technology and the industrial revolution, manipulators gained

popularity on the shop floor, especially in the automotive and aerospace industry, as manipulators are precise and faster than humans in performing few tasks.

As industrial manipulators and robotics technology matured, researchers and industries have appreciated the immense potential of a mobile manipulator for various industrial applications. A mobile manipulator combines an industrial or robotic arm attached to a mobile platform. In literatures, mobile manipulators can be dated back to 1992 when the mobile robot lab at LAAS developed "Hilare 2bis" [35], shown in Figure 1.2. A fixed-base industrial manipulator has a limited workspace, while the mobility features extend the workspace to a larger area. A mobile manipulation system has the benefit of both the manipulator's dexterity and the mobility provided by a mobile platform. Because of these advantages, mobile manipulators can be used in industries for various challenging tasks. Increasing mobile manipulators can reduce the risk for the human worker, remove the monotony of the task, increase efficiency, and maximize the financial benefits.

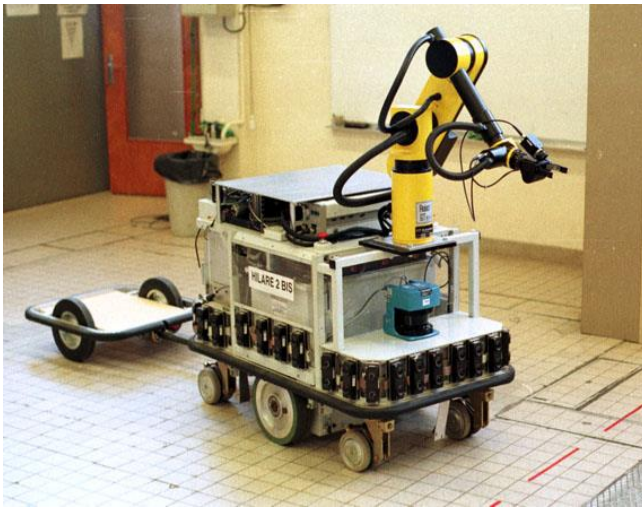


Figure 1.2 Upgraded Hilare 2bis from LAAS with 6-wheel robot and a manipulator [35]

The intelligent mobile manipulator has many advantages over the conventional robotic arm because of its navigation capability. It should have 5 must-have abilities – move, interact, see, think, and communicate [36]. It can be controlled in many ways, such as by a human operator through teleoperation or supervised

autonomy, where humans supervise the autonomous operation, or fully autonomous, where no human supervision is required.

1.5 Railway track environment from a robot's perspective

Railway infrastructures are different from other industries in many ways. Trackside furniture, stations, bridges, and tunnels are examples of railway infrastructure. Some are very big, such as stations, tunnels, or bridges, while some are small, such as trackside furniture. Components in the railway environment vary heavily in shape as well. There are some trackside infrastructures (furniture) such as marker post, signal post etc. which are important to operate and regulate trains along the track. Because of the distinguished nature of the railway infrastructure, a commercially available robot will face some unique challenges while performing manoeuvre tasks in the railway environment from the infrastructure or regulatory obstacles, as shown in Figure 1.3.

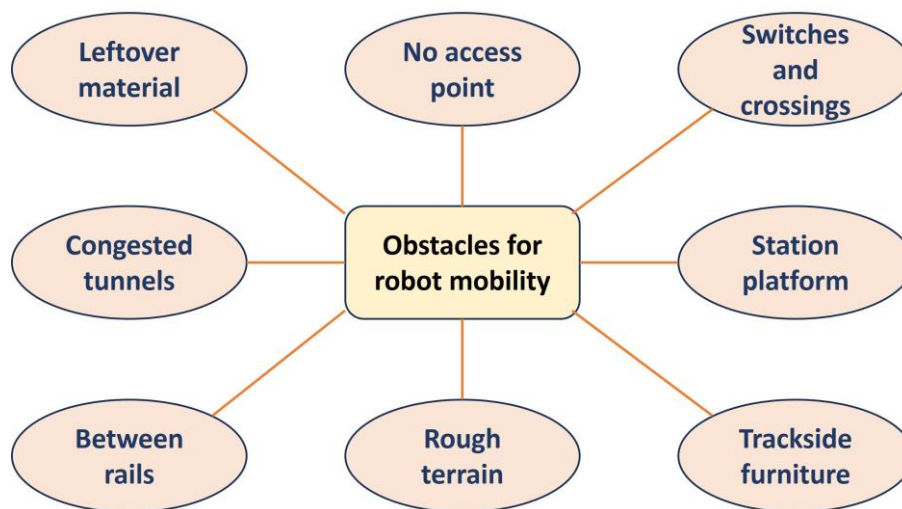


Figure 1.3 Challenges for a robot based on railway infrastructure

1.5.1 Track

The track is the most important infrastructure of the railway asset network. The railway track is the path where the trains run. The main components of a traditional track are:

- Rails
- Sleepers
- Ballasts
- Fittings and fasteners

Rails are parallel and continuous steel structures in a railway track. They guide the train in a lateral path and minimize the risk of derailment. Also, they bear and transmit vertical stress and carry the trains along the designated path. Sleepers are rectangular-shaped support element that holds the rails in the correct gauge and alignments. Sleepers are positioned perpendicular to the rails. Apart from holding the rails, sleepers also provide firm and even support, transferring load from the rail to a broader area of ballast. Sleepers are not evenly spaced; the density of sleepers is higher at the end of the rail or joint area. Sleepers can be made from wood, steel, cast iron, or concrete. Ballast is a layer of granular material to hold the sleeper in the correct position and provides hard and leveled ground. Typically, broken stones are used as ballast material. Transferring and distributing the load from sleepers to a broader area is another functionality of ballast. Apart from these, it also provides necessary drainage for the track. The major components of the railway track are shown in Figure 1.4.

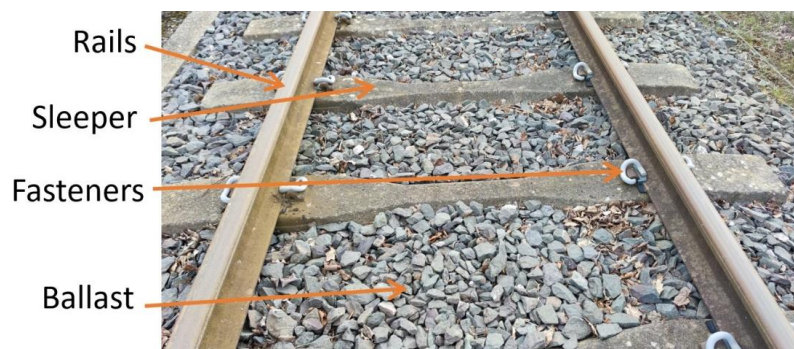


Figure 1.4 Major components of a railway track

As the construction and design of railway track is different than road, industrial establishment, mobile manipulators will face some unique challenges for navigating along the track. Uneven surface of the railway track is not suitable for pneumatic tyres which will induce vibration in the output of the sensors. Moreover, If the mobile manipulator wants to change the side, going from right to left or

navigating to outside of the track from inside, it should cross over the rails. So, the mobile manipulator should be able to cross an obstacle of a minimum rail height. Apart from these, modern railway tracks also have overhead electric lines running along the track on a support system consisting of trackside poles, links, or clamps. These overhead electric cables transfer energy to run the train through an overhead catenary system. As a result, these trackside poles, links, or clamps used for the overhead lines act as obstacles for the mobile manipulator, and the accessible area for the mobile manipulator has been reduced. These distances are known as structure gauges. Table 1.1 shows structure gauges for UK railway infrastructures as per Network Rail standards (NR RT/CE/P/015 – Structure Gauge Policy and NR/L2/TRK/3201–Management of Tight Clearance and Track Position):

Table 1.1 Structure gauges (clearance between railway trackside infrastructure)

Item	Clearance
Fixed wire (from track plane)	6100 mm
Temporary wire (from track plane)	5500 mm
High or big structures (Post, platforms, Ladders, Handrails, screening, etc.) (from running edge)	- minimum 1364 mm - normal 1624 mm
Cable route (from running edge)	- minimum 1250 mm - normal 2000 mm
Walkway (from the track)	- minimum 1300 mm - normal 2000 mm
Signaling equipment (marker post, LOC, etc.) (from running edge)	730 mm

1.5.2 Switch and Crossings

Switch and crossing (S&C) are a mechanism that guides the rolling stock in the desired direction. It is very common near railway stations and railway junctions. In a simple form, it can guide trains between 2 railway tracks. While in a complex form, it can guide multiple trains on multiple tracks. The switching mechanism consists of stock rails, wing rails, check rails and switch rails. Among these, switch rails are movable. Wing rails relate to the switch rails. In ancient times, a manual lever mechanism was used to control the position of the switch rail. In

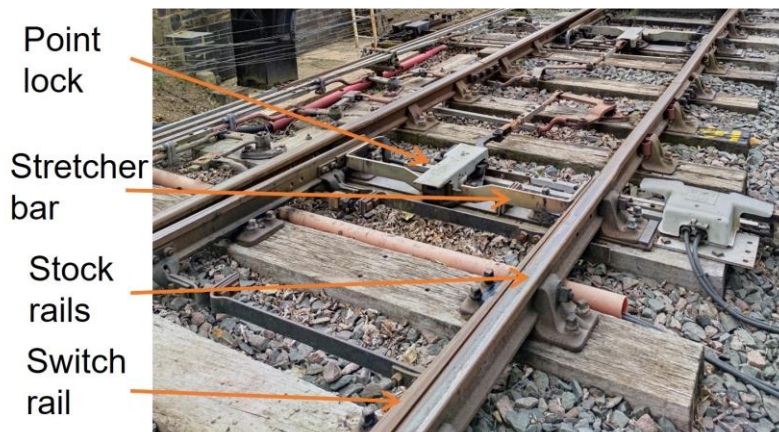


Figure 1.5 Components of switches and crossings

modern days, electric motors are used for automatic control of switching. Check rails, also known as guard rails, ensure that the train is guided to the right track. The components of switches and crossings are shown in Figure 1.5.

Though switches and crossings are a part of the track, they impose more complex problems for a mobile manipulator to navigate along the track. For example, in switches and crossings, the gap between 2 rails is reduced less than the standard gauge (1435mm). Moreover, a point machine, electric motors, or manual lever mechanism is installed beside the track to operate switches and crossings. These items will act as an obstacle for mobile manipulator and reduce the accessible area for off-track navigation along the track.

1.5.3 Tunnel and Bridges

Tunnels and bridges are an essential part of the railway transportation network. A tunnel is a confined space under a mountain or river, or sea to allow continuous running of rails. It is very helpful in mountain or hilly areas as the train struggles and requires much power to pass over a mountain. The tunnel can be very long or short, depending on the geography and requirement. Similar to tunnels, railway bridges allow the rails to cross rivers or obstacles without any stops. Both tunnels and bridges can shorten the distance of traveling between 2 destinations.

Because of their structure, tunnel and bridges can create some inconveniences for an autonomous robot. One big problem in the tunnel environment is that it

lacks a signal. The effect is similar to faraday cage signal blockage. Operating in a confined space is a challenge for mobile manipulator. Some of the tunnels have limited space along the track, as shown in Figure 1.6 [37], [38]. Without a safe workspace, the manipulator may collide with the tunnel wall or other structure, which creates a safety concern for the manipulator.



Figure 1.6 Railway tunnels [37, 38]

1.5.4 Trackside Furniture

Trackside furniture or trackside equipment is positioned on the side of the railway track. They are important to transfer information between the train and the control centre to avoid a collision. Signals posts [39], marker posts [40], Location cases (LOCs) [41], etc., are an example of trackside furniture, Figure 1.7. The signal or marker post size can vary depending on the place and requirement.

Trackside furniture provides an advantage for robot navigation. These can be used as a feature for autonomous navigation. In addition, using this equipment as a reference for the robot will help to locate itself more precisely. But, apart from that, this equipment also acts as an obstacle for the mobile manipulator for off-track navigation and reduces the accessible areas along the track.

1.6 Use of mobile manipulator for railway track maintenance

1.6.1 Shortcomings of current railway track maintenance technology

Railway companies are working continuously to improve the current maintenance techniques to ensure the safety of passengers and freight trains. Maintaining

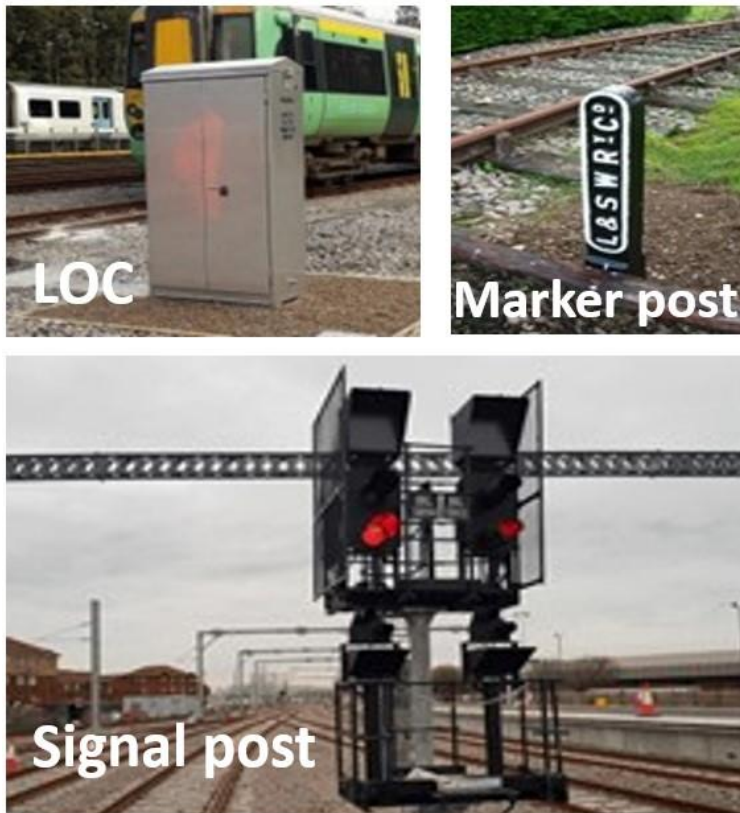


Figure 1.7 Railway trackside furniture, from left to right is LOC's [41], Marker post [40], Signal post [39]

proper track conditions is vital to reduce railway accidents and to provide a comfortable journey. But there are some shortcomings in current maintenance technique which can be improved by using robotics solutions. Few vital shortcomings are listed below:

A. Human dependency and related errors

As per the report published by the Health and Safety Executive (HSE) of the UK, human-related errors are the leading causes of major accidents in railway as technical systems are becoming more reliable day-by-day [42]. These errors can be classified as a lack of skill in using maintenance tools, improper communication, etc. Even after having many years of work experience and expertise, a human can make errors in the execution of maintenance tasks. Railway maintenance tasks are arduous and require the patience and discipline

of the technician towards the task [43]. So, in many cases, human errors in railway maintenance tasks are unpreventable. Common types of human-related errors in railway maintenance are disassembly errors, inspection errors, decision-making errors, installation errors, etc. [44]. Autonomous robots can solve human-related errors as robots are known for accuracy and precision and can work for more extended periods spontaneously.

B. Safety

The safety of the maintenance employees is another concern. Some maintenance employees work along the track in an unfavourable environment. To maintain the train travel schedule, sometimes maintenance work must be done 24/7, even in adverse weather. Sometimes they must work at height in an unergonomic position for a prolonged period. Moreover, train running at high speed on the adjacent line puts the life of maintenance employees in danger. Data on Work-related Health in Railway Operatives published by HSE in March 2018 shows that from 2011 to 2017, there were 350 new cases of long-latency diseases (asbestos-related and pneumoconiosis) and 40 new cases of short-latency diseases (mostly hand-arm vibration) [45]. This report also indicates that the average cost due to work-related illness in 2016/2017 was 18400 GBP per employee [45]. Hence, companies incur direct and indirect losses in workforce and cost because of safety issues.

C. Cost

Current railway maintenance techniques are costly, so Network Rail spends a big slice of the yearly budget to keep the railway infrastructure running properly. For example, the Rail Industry Finance (UK) report for the 2020-21 fiscal year showed that Network Rail spent 1.9 billion GBP for maintenance [46]. However, reducing the maintenance of railway infrastructure is impossible as this will lead to train delays, cancellations, and customer dissatisfaction, decreasing the passenger number [47]. Many specially equipped trains with advanced sensors and non-destructive testing (NDT) are used for track inspection. For example, NMT can inspect 1000 miles of railway track in 1 shift. As it records 10TB of data for 434 miles of track, it requires continuous human involvement to verify the results from

the image processing software [19]. It uses Plain Line Pattern Recognition (PLPR) to match the acquired data from multiple lasers and cameras with a standard image. Though it is faster in inspection, it can only provide 2D images of defects and no perception of the surrounding area. Therefore, a human operator must verify the defect and surrounding area before performing any repair task. In the current maintenance technique, human intervention is required in many manual track inspection tasks, such as checking the track's damage, finding missing fasteners and fittings, etc. For example, the current vision-based automatic track inspection technique requires verification of the defect, and a human captures the surrounding details. Moreover, many vision-based inspection techniques only have a fixed camera, which has a limited view of the defect area and captures only 2D images of the defect, which is inadequate to determine the required repair work.

Besides the technical and economic shortcomings, humans create shortcomings in many cases for railway track maintenance. The recent pandemic, COVID-19, has limited human interaction, which is important in many maintenance tasks. Moreover, the salary of the maintenance labour is increasing day by day. With the aging population, people with expertise in maintenance work are decreasing, creating an imbalance between the demand and supply of maintenance workers.

1.6.2 Scope of mobile manipulator for track maintenance

Demand for passenger trains and freight trains is increasing. This increasing demand can be fulfilled by adding more train journeys between different destinations and increasing the usage of track and other railway infrastructure. However, more usage of the track will accelerate the rate of deterioration and amplify the chances of faults generation which poses a risk to the safety and comfort of the passenger and goods. Therefore, more frequent track maintenance is required to ensure the track's condition and reliability. Nevertheless, the availability of tracks for maintenance tasks will be reduced as the track usage increases. This contradictory phenomenon challenges the maintenance task scheduling capability and questions the efficiency of current maintenance techniques. However, a mobile manipulator with capability of

performing both improved inspection and repair tasks can solve this problem, as shown in Figure 1.8.

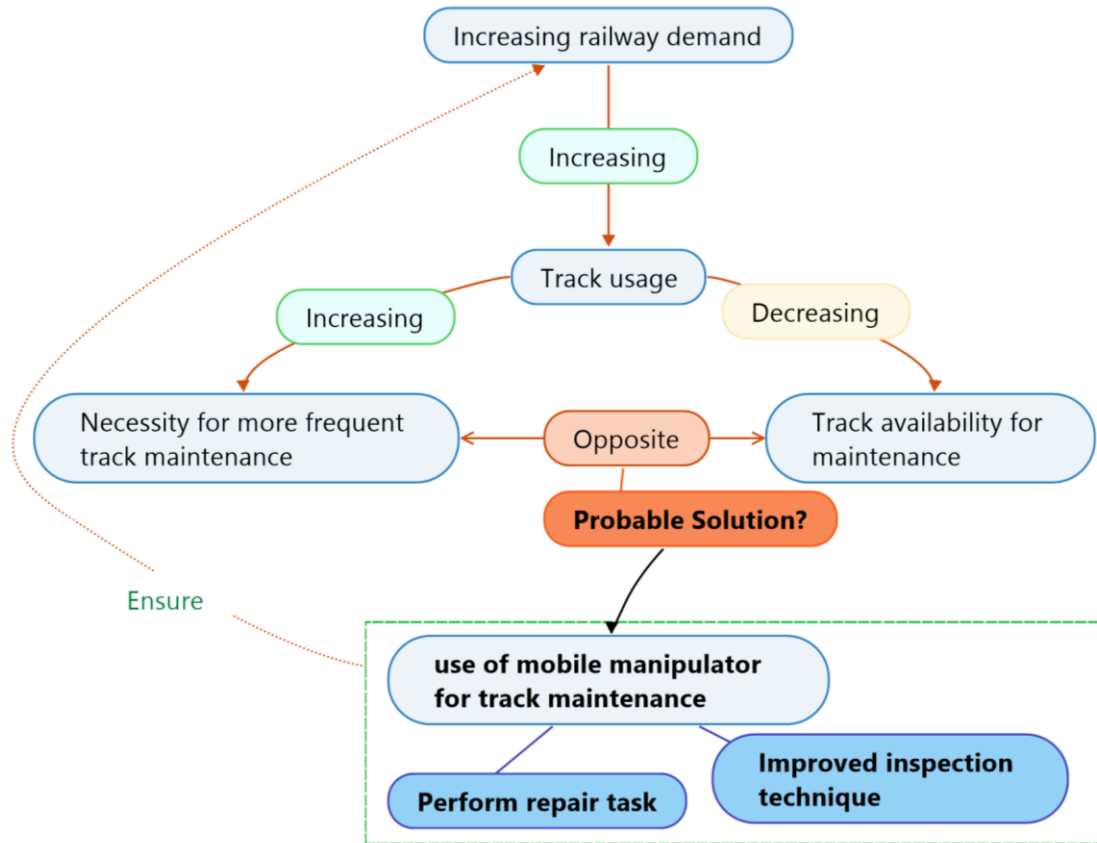


Figure 1.8 Necessity of mobile manipulator for railway track maintenance

Railway infrastructure management companies can be benefited from the use of mobile manipulator in different aspects. Such as:

D. Environmental impact

Rail is a greener means of public transport than the other public transport system. For example, modern Eurostar rail only emits 6g of CO₂ per passenger per kilometre, which is almost zero and insignificant compared to other transport systems [4]. Compared to rail, small and medium-sized petrol cars emit 96g and 192g CO₂ per passenger per kilometre, respectively. Amplifying rail usage for public transport will help achieve the Sustainable Development Goal (SDG) by reducing carbon footprint. But delays and cancellations of the train reduce

customer satisfaction and discourage them from using it for the next trip. In the UK, there was a 3.3% of train cancellation and 10.7% delay by at least 3 minutes during 2021-22 Q4 [46]. Almost 17% of delays were caused by track faults [17]. To maximize the usage of a rail network, adequate track maintenance is mandatory, which can be obtained by the flexible mobile manipulator system with inspection and repair capability.

E. Modernization of the industry

Industry 4.0 is becoming more mature, Industry 5.0 is paving its way to industry, and mobile manipulators are gaining more flexibility and productivity features. Nevertheless, most commercial mobile manipulators are designed for indoor applications, mainly industrial workspaces. These robots are not only working autonomously but also working collaboratively with the human workforce. As the manufacturing cost of robots are reduced gradually [47], and some hardware features such as water and dust resistance are becoming easy to achieve. Hence, the design and fabrication of all-weather outdoor mobile manipulator are becoming more feasible. Moreover, manipulators are now capable of carrying very high payloads. For example, ABB has multiple models of industrial manipulators which can carry more than 200kg payload [48]. As many railway maintenance tasks require a high payload, these manipulators can perform these tasks.

F. Financial benefits

The financial benefits of robots and automation for other industries are already proven theoretically and practically. For example, a cable-driven construction robot can save more than 1.2 million euros in 5 years after considering the cost-benefit analysis with the labour [49]. Furthermore, after considering 10 robots working in 12 different construction sites, researchers concluded that robots save 13% on cost while increasing quality by almost 50% [50]. These examples show that rail companies will be benefited from using mobile manipulators for railway track inspection and repair tasks which will reduce human involvement. With adequately equipped inspection sensors, the mobile manipulator can inspect the railway track autonomously. Moreover, it can perform some repair tasks

autonomously, assist humans in the repair task, and ensure information flow in different steps.

G. Short track possession for quick inspection and repair tasks

There are few state-of-the-art railway track monitoring devices, such as NMT, in operation. But they are very costly to operate and need a vast number of crews per shift. For that reason, these special monitoring and inspection trains are operated only a few times per year. Human involvement is mandatory for an instant or frequent track inspection and repair task. Besides, during the inspection and repair task, track possession is mandatory for longer period to ensure human safety. However, adjacent live track poses a risk of injury. This method reduces track utilization as passengers, or freight trains are not allowed to use the track during track possession by maintenance work. The forecasted increasing demand requires more frequent passenger and freight trains. More frequent track maintenance will be required with this increased track usage. This is a dilemma and challenge for railway track maintenance companies to ensure the optimum utilization of the assets. A fleet of mobile manipulators can be used to perform and assist humans in these frequent and instant inspection and repair tasks.

H. Safety and safety related cost reduction

Though many modern inspection technologies perform track inspection tasks in railways, validation by a human operator is required in many cases. Repair tasks in railways require higher accuracy to ensure track quality for passenger and freight safety. Weather conditions, tiredness, monotony, etc., can affect the accuracy of inspection and repair tasks completed by a human operator. According to rail safety statistics, there were almost thousands of injuries in the mainline and tube from March 2021 to April 2022 [51], which accounted for 0.02 billion GBP cost for the companies [45]. Injuries will fall if a robot can assist or replace humans in those dangerous maintenance activities. Hence, the use of mobile manipulators can save not only money but also human health.

I. Connectivity in process

As the 5G and cellular technologies are becoming stronger day by day, a fleet of connected mobile manipulators is becoming possible. These connected mobile manipulators can communicate with the central command and control centre with proper security measures. Upon receiving the task command, the mobile manipulator can autonomously navigate to the designated area for inspection or repair tasks. The human operator can also navigate the mobile manipulator through teleoperation to help him perform the repair task.

1.7 AIM AND OBJECTIVES

1.7.1 Aim

This thesis aims to develop a multipurpose Robotic Inspection and Repair System (RIRS) capable of performing improved railway track inspection by fusion of multiple sensors and demonstrating multiple simulated track repair tasks with and without human involvement.

1.7.2 Objectives

This research will focus on railway track inspection and repair tasks. To achieve the above aim, the whole thesis will be divided into 4 objectives as follows:

- 1) **Gap identification and Simulation:** To simulate a Robotic Inspection and Repair System (RIRS) by identifying the research gaps in the current railway track maintenance technology after reviewing the conventional and state-of-the-art maintenance technology in railway and other industries,
- 2) **Kinematics, repeatability, and precision:** To analyse the kinematics of the RIRS in different navigation scenarios and then examine the repeatability and precision of an industrial manipulator on a mobile base with varying payload for railway track inspection and repair tasks,
- 3) **Improved inspection technique:** To propose and test an improved inspection method by generating a 3D model of the maintenance target using vision-based sensor fusion to eliminate the shortcomings of RGB-D camera,
- 4) **Repair task framework:** To demonstrate the usability of a mobile manipulator for multiple railway repair tasks with and without human involvement and investigate the performance.

1.8 Overview of the thesis

This research is divided into 6 main chapters: Introduction, literature review, methodology, results, discussion, and conclusion. An overview of the whole thesis is given in Figure 1.9. Each chapter will be divided into multiple sub-chapters to describe the different aspects and objectives completion.

Introduction chapter starts by describing the importance of the railway. Then a brief narrative on conventional railway track maintenance technique has been given. Furthermore, railway track and trackside environment has been discussed to find out the challenges for a robot from railway infrastructure. Next, shortcomings of current track maintenance technique and scope of mobile manipulator has been discussed briefly to show the potential use of mobile manipulator. Finally, aim and objectives of this thesis have been portrayed.

Literature review has been divided into 4 main sub-chapters. Initially, conventional track maintenance machines have been discussed. Furthermore, a thorough investigation has been conducted about the robotic technologies used in railway maintenance. Additionally, robotic maintenance technology in other industries has been reviewed. Finally, research gaps have been discussed based on the literature review.

Third chapter discuss about the proposed robotic solution and different methodologies to achieve the defined objectives. An overall research methodology has been laid out in the beginning. Then overview of the proposed robotic solution has been discussed followed the by software architecture of the system. Next, hardware analyses have been carried out to find the kinematics, repeatability, and precision of the proposed solution. Afterward, an improved inspection method has been designed. Multiple repair task frameworks have been created subsequently.

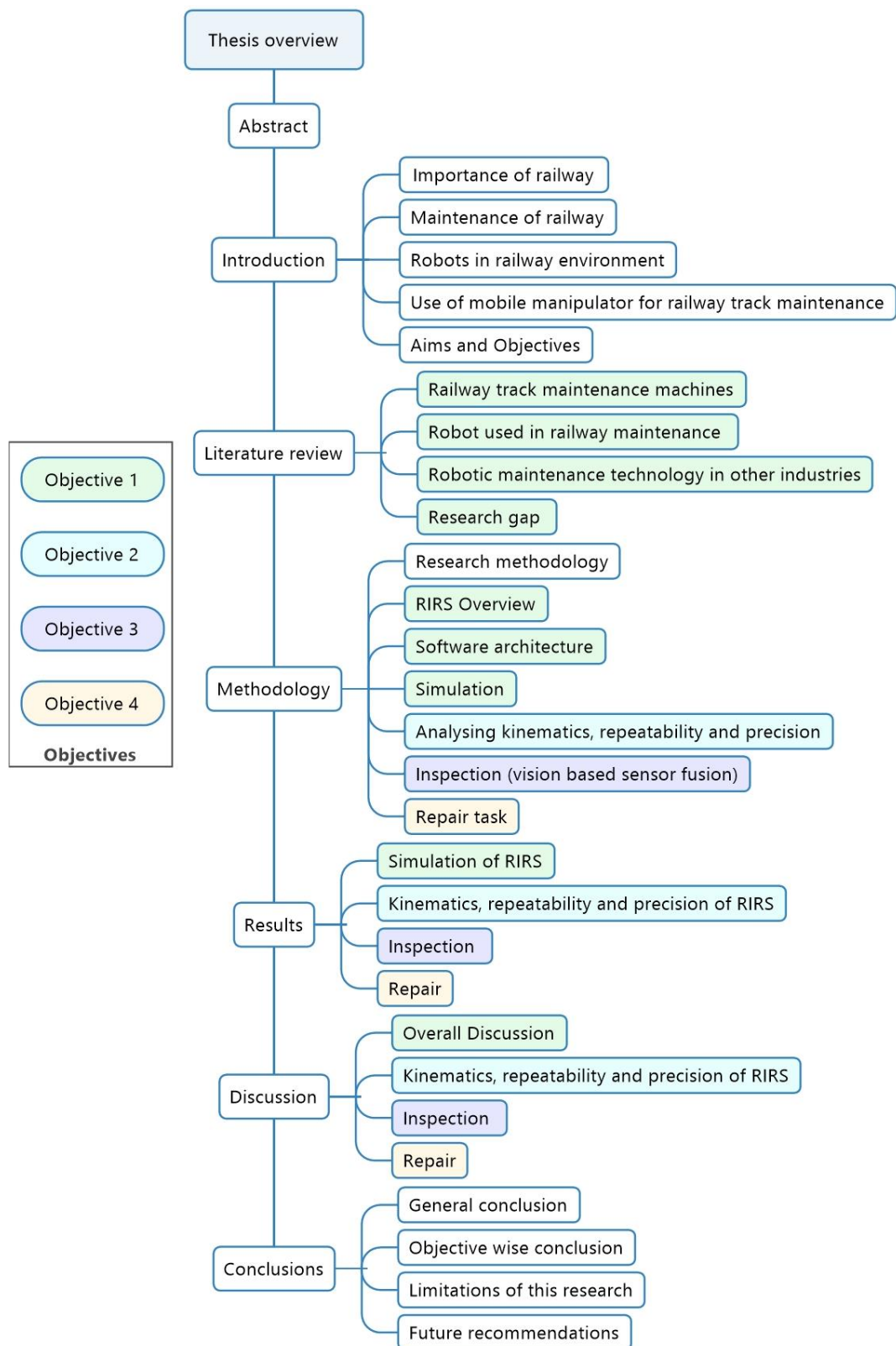


Figure 1.9 Overview of the thesis

Next chapter display the gathered results and describe the different experimental setup. Contents are divided into 4 main sub-chapters based on the objectives. In the beginning, different simulation environments have been showed. Afterwards, kinematics of RIRS will be discussed followed by repeatability of precision of mobile manipulator for track inspection and repair tasks. Then the experimental setup and performance of the proposed improved inspection method has been depicted. Final sub-chapter discuss about the experimental setup and performance of multiple simulated track repair tasks.

Discussion chapter is divided into 4 sub-chapters where each sub-chapter discuss about each objective. First, overall discussion has been provided. Subsequently, kinematics, repeatability and precision has been discussed. Then critical analysis of the results from the inspection method has been analysed. Finally, performance analysis of the repair tasks has been carried out.

Conclusion is the final chapter of this thesis. Initially an overall conclusion has been provided to show the potential impacts of this research. Furthermore, limitations of this research have been discussed followed by the future recommendations.

2 LITERATURE REVIEW

The railway track is a vital infrastructure of the railway network. The proper condition of the railway track will ensure a safe and comfortable journey for the passenger and timely delivery of goods to the business entities by freight trains. Railway maintenance task combines both inspection and repair task. However, few robotic technologies have been researched and introduced in maintenance; conventional maintenance mostly depends on different types of machines and devices.

In the beginning, this chapter will give an idea about the conventional machines for railway track maintenance. After that, use of robotic technology will be discussed in depth to find the shortcomings of current maintenance techniques in section 2.2. In section 2.3, application of mobile manipulator for different types of inspection and repair task in different industries will be discussed to find out the advantage of using mobile manipulator for maintenance tasks. Finally, a gap analysis has been carried out in section 2.4 to find the research gaps and probable solution.

2.1 Conventional railway track maintenance machines

Railway maintenance task combines both inspection and repair task. Track maintenance machines can be massive in size, the same as a train, or very small, which can be a hand-held device. Also, these machines can be stationary or movable, depending on their usage. Usually, fixed platforms are used in crucial positions along the railway infrastructure network to record information such as passing train information, monitoring track components, etc. On the other hand, movable platforms are designed to run along the railway track to perform specific tasks at a certain speed.

In some cases, the movable platform can be airborne such as Unmanned Aerial Vehicle (UAV) or a helicopter. Table 2.1 shows the advantages and disadvantages of several types of railway track maintenance technologies. These platforms are equipped with various types of sensors for navigation and for completing inspection tasks. Common inspection methods so far used include

visual inspection, Digital Image Correlation (DIC) [52] and thermal cameras [53] for rail component inspection, laser-vision sensors [54] for rail wear and track geometry, laser distance meter [55] for rail corrugation, ultrasonic sensors [56] for surface and internal defects, Light Detection and Ranging (LiDAR) [57] for track clearance and vegetation detection, eddy currents [58] for rail internal defects, ElectroMagnetic Acoustic Transducer (EMAT) [59] for surface and internal defects, Ground Penetrating Radar (GPR) [60] for ballast fouling and moisture, Interferometric Synthetic Aperture Radar (InSAR) [61] for track settlement, etc.

Table 2.1 Type of track maintenance platforms used in railway

Type	Main tasks	Advantages	Disadvantages
Push-trolley	Inspection of track, carry material	Smaller in size, easy to transport	Engage track, slow speed, human-operated
Road-Rail Vehicle (RRV)	Repair, transport material, inspection	High speed, easy to transport	Very costly, highly dependent on human operator
Specialist trains	Repair and inspection of track	High speed, multiple measurement devices, extremely high payload	Very costly
Train-borne system	Inspection of track, repair of track	Moderate cost, easy to install	Limited validation

2.1.1 Push-Trolley

Inspection trolleys used in railway maintenance are capable of on-track navigation because of their rail shaped wheel. The size of the trolley depends on the payload requirements. They can be configured for different purposes; for carrying materials for repairs or seats carrying people. Depending on the size and configuration, these simple trolleys can carry up to 60kg of material [62]. Figure 2.1 is an example of a track geometry inspection trolley. Inspection trolleys are typically trimmed down to reduce mass and increase sensor payload capability. They are ideal for ultrasonic inspection systems because of their slower operating speed. Moreover, they are widely used for track geometry validation on short



Figure 2.1 Trolley for carrying material [64]

lengths of track [63]. GPS capability provides the defect location information [64]. As human typically pushes these, their maximum speed is limited and only suitable for short distances. Disadvantages of push-trolley for smart maintenance is that it can only carry limited payload before it become very heavy to push. Moreover, it lacks the localization and navigation modules for payload deployment.

These trolleys are trimmed down for inspection to make them lightweight, and different sensors are added for defect detection. They are ideal for ultrasonic sensors because of their slow operating speed. They are widely used for track geometry validation [63]. An attached GPS will provide the location information of the defect [64]. As human manually pushes these, their speed is limited, usually the same as human walking speed, and only suitable short distance track inspection. Figure 2.2 is an example of track geometry inspection trolleys.

2.1.2 Road-Rail Vehicle

Road-Rail Vehicle (RRV) is a unique type of maintenance equipment with both pneumatic and rail wheels, shown in Figure 2.3. Usually, the rail wheels can be engaged and retracted easily. Pneumatic wheels are powered by the engine, and power is transferred to the rail wheel by friction. These multipurpose maintenance platforms can be used for transporting material or persons or as an inspection device with proper sensor attachment. Rail Industry Safety and Standards Board



(a)



(b)

Figure 2.2 Track geometry inspection trolley, (a) and (b) are reproduced from [65] and [66], respectively

(RISSB) classifies the RRVs into three main categories based on power transmission mechanism: self-powered, friction drive and direct drive [65]. Moreover, braking power, safety procedures during road-rail conversion and seating arrangements are also standardized by Rial Safety and Standard Board (RSSB) to reduce accidents and maintain safety [66].

High-load-carrying RRVs are usually used for repair or construction tasks, while smaller RRVs can be used for inspection. Unimog, shown in Figure 2.3(a), from



(a)



(b)

Figure 2.3 Road-rail vehicles for track maintenance, (a) Unimog road rail vehicle for overhead line maintenance [69], (b) mobile asset management RRV [70]

Mercedes-Benz [67] can tow up to 1000 tons and serving the railway industry for a long time. RRV can be used for towing a trailer or carrying a crane for different maintenance purposes.

Many sensors can be attached to RRVs for detecting rail defects, tracking geometry measurements, etc. For example, vision-based sensors, laser sensors, eddy current sensors, ElectroMagnetic Acoustic Transducers, etc. With multiple LiDAR and cameras, RRVs can provide a dense model of the railway environment, which are essential for railway survey and asset management. Road-rail access points, vegetation, and the condition of track infrastructure can be assessed from the survey and scanning results. Figure 2.3(b) shows a Mobile Asset Management platform with LiDAR, which can provide scanning results with a maximum speed of 20mph. RRVs are much faster than manual inspection trolleys and can carry more weight. But they are slower than trains. Some RRVs, such as Runner Wizard V2R [68] has a maximum speed of 100 km/h while running on the track.

2.1.3 Specialist train

Maintaining a vast railway network is particularly a challenging task. Inspection and monitoring of these lengthy tracks require a faster inspection method. A specially designed train with sensors can provide better results and reduce track possession by quickly finishing the inspection task while carrying the repair crew and materials. New Measurement Train (NMT) and Mobile Maintenance Train (MMT) are 2 state-of-the-art specialist trains in the UK for track inspection and repair, as shown in Figure 2.4. Though these trains provide inspection and repair at a higher speed and accuracy, operation cost per shift is exceedingly high, limiting their usage to a rare frequency and only for a specific purpose.

NMT, shown in Figure 2.4(a), is fitted with multiple cameras, laser sensors, transducers, and accelerometers [19]. Plain Line Pattern Recognition (PLPR) can detect anomalies, and the transducer and accelerometers can record the track geometry. Running at a maximum speed of 125mph, it generates and records almost 10 terabytes of data on an onboard computer for 440 miles. Inspection with specialized trains is done uniformly to determine the track quality and create

a track maintenance plan. On the other hand, MMT, shown in Figure 2.4(b), is a specially designed and equipped train for track maintenance tasks and is considered a "workshop on wheels" because of providing vast choices of repair equipment, electric, hydraulic, and pneumatic power outlets, extendable work area providing safe workplace from sun, rain and running train [23].



(a)



(b)

Figure 2.4 Specially designed train in the UK for railway maintenance; (a) New Measurement train (NMT) [19], (b) Mobile Maintenance Train (MMT) [23]

2.1.4 Train-borne system

Train-borne systems are convenient for finding the rail's geometrical or structural properties or remote condition monitoring of the rail. It is easy to install and movable. As this can be fitted with any train and does not need a separate power

source, it is economically viable and more efficient than the conventional track inspection method [69]. For example, Machines with Vision provide a train-borne track condition monitoring system based on computer vision for preventive maintenance [70]. With easy-to-fit hardware with a millimetre level of accuracy and real-time data communications and processing software, this train-borne system can reduce human involvement in condition monitoring. Another example of a train-borne system is RILA from Furgo [71]. It is a remote train-mounted system that combines laser, computer vision, and GPS for remote track surveys without humans stepping on the track.

2.2 Robots used in railway maintenance

For financial and safety reasons, robotic technologies are slowly becoming popular in the railway industry [72]. But most Robotic and Autonomous Systems (RAS) are developed for rolling stock maintenance. Though railway track is one of the most critical infrastructures, only 28% of the RAS are adopted for track maintenance [31]. Moreover, very few, only 6 of the RAS, have manipulation capability while the rest are without manipulation capability [31]. In this literature review, RAS includes robotic and state-of-the-art sensing and defect identification technology based on modern sensors for railway infrastructure maintenance, including rolling stock, tracks, bridges, tunnels, etc.

2.2.1 Robotics inspection technology in railway

Railway network consists of different types of infrastructures and uses various inspection methods for ensure the safe operating condition. Discussing all the inspection methods used in railway is out-of-scope for this research as this research particularly focus on the track inspection and repair technique. So, the inspection technologies only for track and related infrastructure (such as tunnel and bridges) are discussed. Moreover, this thesis focuses on the vision-based inspection systems for finding railway track. Hence, a separate sub-section has been included to discuss the past and current vision-based inspection methods and to show the scope of improvement.

2.2.1.1 Track inspection

The track is one of the essential assets of the railway network, allowing trains to move between places. They are made of different components. It is subject to enormous stresses of the lateral and longitudinal forces from trains [73]. Apart from the forces, environmental factors such as heavy rain, extreme cold, snow, scorching heat, etc., accelerate the degradation of the track components. Railway track maintenance includes various types of inspection and repair tasks to resist any catastrophic failure by ensuring the strength of the track. The railway asset management team performs track maintenance tasks regularly to maintain the track's geometric shape, extend the life of track components, and identify and rectify any potential failure of the track. Table 2.2 shows summary of different track inspection robots.

Table 2.2 Track inspection robots

Platform or type	Main function	Sensing method	Key technology and contribution
STAR-TRACK [74]	Inspection and repair of fastening bolts	Monocular camera	Multi-robot autonomous track maintenance
Train-borne system [75]	Fastener inspection	TV camera	Neural classifier for fastener detection
Train-borne system [76]	Fastener detection	Video camera	Artificial lighting source to reduce ambience noise
Visual Inspection System for Railway (VISyR) [77]	Fastener detection	camera	Computer vision and machine learning, maximum speed of 200km/h
Train-borne system [78]	Fastener detection	camera	Machine learning, structured lighting condition
Manual trolley [79]	Turn out and tie detection	Multiple camera	Slow speed, machine learning algorithm
RRV [80]	Track component inspection	4 cameras, GPS	Defect location based on GPS

Train-borne system [81]	Fastener detection	CCD camera, GPS	Machine learning for detection and GPS for location information
Comprehensive Track Inspection Vehicle (CTIV) [82]	Fastener detection	camera	Fastener quality based on image and machine learning algorithm
Modified truck with sensor [83]	Rail-road crossing inspection	Camera, LiDAR	Pattern recognition, 3D point cloud using Support Vector Machine (SVM)
Diagnostic Analysis for Railways and Trams (DART) [84]	Track geometry	Track measurement	Computer assisted diagnostic tool
Hand pushed inspection device [85]	Faulty rail profile detection	Multiple laser camera	3D modelling of rail, deep learning model for detecting faulty rail profile
Robot trolley with sensors [86]	Fatigue cracks on track	Alternating Current Field Measurement (ACFM)	Automatic crack detection robot
Railpod [87]	Track inspection	Customizable based on requirements	Easy to transport, both rail and pneumatic wheel
AutoScan [88]	Crack detection on track	ElectroMagnetic Acoustic Transducer (EMAT)	Manipulator for further inspection
RIIS1005 [54]	Multiple track defects	Camera, LiDAR	Easy to assemble and disassemble, pattern recognition, deep learning
Felix [89]	Switches and Crossings (S&C)	laser	artificial vision system based on profilometry, wireless data transfer

First significant research on the use of robotic technology for track maintenance was carried out by Trivedi et. al. The project, SysTem for Autonomous Railway TRACK operations (STAR-TRACK), was a project to find the feasibility of robots in track maintenance and develop a multi-robot based autonomous track

maintenance system for high-speed Shinkansen line to perform 2 specific task: loosening bolts after detection and assembly of new fastener automatically [74]. The designed system included cameras and gripper for detecting and loosening bolts and feeding and strengthening fastener.

Fasteners fix the rails to the sleepers. Missing fasteners reduce the stiffness of the track. A computer vision-based approach was developed by Mazzeo et al. to detect the absence of fasteners [75] using a camera which increased detection reliability more than humans. Another computer vision-based missing clip detection technique was developed by Singh et al. using a video camera and an artificial lighting source to eliminate the inconsistent illumination effect in image processing [76]. Marino et al. presented Visual Inspection System for Railways (VISyR) to determine the presence of fasteners using computer vision and machine learning at a maximum speed of 200 km/h [77]. Another machine learning-based fastener inspection method was proposed by Zhang et al. by combining unstructured and structured lighting conditions to reduce the impact of light, vibration, and obstacles [78]. This proposed method was more reliable in practical conditions as the training dataset was robust.

A conceptual hand pushed cart with computer vision and machine learning algorithm was developed for turnout detection and tie detection [79]. Figure 2.5(a) shows the inspection device with multiple cameras and on-board computer for

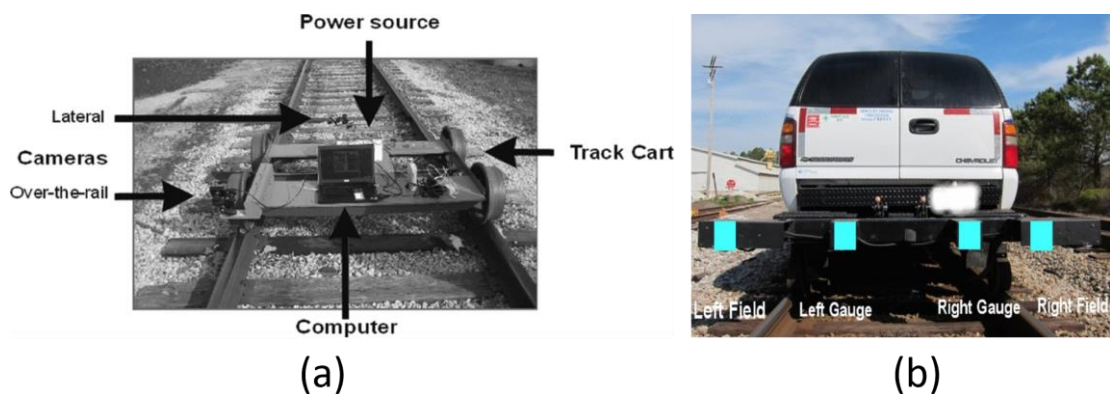


Figure 2.5 Track inspection with multiple cameras; (a) inspection trolley [81], (b) modified RRV [82]

track inspection. Li et. al proposed another vision based track component inspection system [80]. Though this is not the fastest but can detect multiple components using 4 cameras with geo-location from onboard GPS, as shown in Figure 2.5(b). Wang et. al also used machine learning and computer vision for missing fastener detection [81]. Sometimes inferior quality of clip can cause failure in near future. Gibert et. al proposed a machine learning based approach which was capable of detecting multiple types of clips as well as determine their quality [82].

Rail-road crossing is a significant place of accident occurrence in railway networks. For example, in 2009, 248 people died in the grade crossings of USA because of humped elevation profile, excessive vegetation etc. Pradeep Ranganathan and Edwin Olson depicted an automatic inspection method to determine the safety condition of the rail-road crossings by post-processing LiDAR data collected using a modified truck [83].

Track geometry parameters describe the quality of the track and what might need repair interventions. Diagnostic Analysis for Railways and Trams (DART) was able to point out the weak location of railway tracks by calculating and measuring multiple geometrical components using a laser scanner [84]. Later, those degraded or deteriorated track locations for potential future defects can be further inspected by a human. Using multiple laser cameras, Santur et al. created 3D modelling of the rail track to train and identify defects [85].

In the recent years, multiple state-of-the-art robotic solution has been developed, as shown in Figure 2.6, for track inspection to which are easy to transport and reduce human involvement. Railpod, shown in Figure 2.6(a), is such an inspection platform which can perform autonomous navigation along the track or can be controlled by remote operation. Based on customer need, various types of inspection module can be attached with it. Because of having both the pneumatic and rail wheel, it can be easily transported along the road [87].

Cracks are a common type of faults in railway track. Rowshandel et. al. proposed a system consisting of a commercially available Alternating Current Field Measurement (ACFM) sensor attached to trolley and a robot for detecting rolling

contact fatigue cracks in rails [86], shown in Figure 2.6(b). This inspection device run at a maximum speed of 5km/h and store the inspection data for 31km of track.

AutoScan, a project from Horizon 2020, is another example of autonomous robotic inspection platform which also has a manipulator for close inspection, shown in Figure 2.6(c). Initially, Rolling Contact Fatigue (RCF) or Electromagnetic Acoustic Transducer (EMAT) sensors are used to find the defect. Then the operator can use the manipulator to inspect further by standing on the side of the rail [88].

RIIS1005 is another intelligent track inspection robot which is developed by Shenhao Technology, shown in Figure 2.6(d). It has modular design which can be easily disassembled and reassembled for easy transportation [54]. With the combination of deep learning, pattern recognition, feature matching technology,



(a)



(b)



(c)



(d)

Figure 2.6 Autonomous track inspection robot; (a) Railpod [89], (b) robot with ACFM sensor [88], (c) AutoScan [90], (d) RIIS1005 [56]

point cloud processing technology it can detect multiple types of defects on rail surface, fattener, ties etc.

Switches and crossings (S&C) are important part of the track which guides the train in proper direction. Felix, shown in Figure 2.7, an automated inspection kart, developed by Loccioni, is designed to inspect the S&C and track [89] with a maximum operational speed of 5km/h. Later, Loccioni train mounted inspection module has been developed based on the knowledge of Felix.



Figure 2.7 Felix in operation to inspect switch and crossing [91]

2.2.1.2 Tunnel and bridge inspection

Tunnels and bridges are important infrastructures of railway network as they help the train to cross big obstacles such as waterways or mountains. Usually, bridges are situated in a remote place and difficult to access. Inspection of bridges poses fatal risks for the operator, specially inspecting the bottom of the bridge is extremely dangerous. Tunnel is usually dark and enclosed place which lacks usual communication such methods as mobile or GPS. Apart from the track inspection inside the tunnel, special inspection method is required to inspect the structural integrity of tunnel. Many researchers have designed and proposed robotic inspection techniques to reduce the risk of human life involved in bridge and tunnel inspection.

Unmanned Aerial Vehicle (UAV) can navigate in remote places with and without help of human. Jung et. al proposed a framework for autonomous bridge

inspection using a UAV equipped with LiDAR, GPS, IMU, camera [90], shown in Figure 2.8(a). The UAV can localize and navigate between different waypoints on a predefined map and acquire inspection data which are processed later. Wang et. al also demonstrated automatic data acquisition for bridge bottom inspection using a tethered UAV [91], shown in Figure 2.8(b). Phillips et. al took a different approach. They developed an autonomous bridge bottom inspection robot with an Unmanned Ground Vehicle (UGV) [92], shown in Figure 2.8(c). The system was equipped with a LiDAR and GPS for data acquisition.

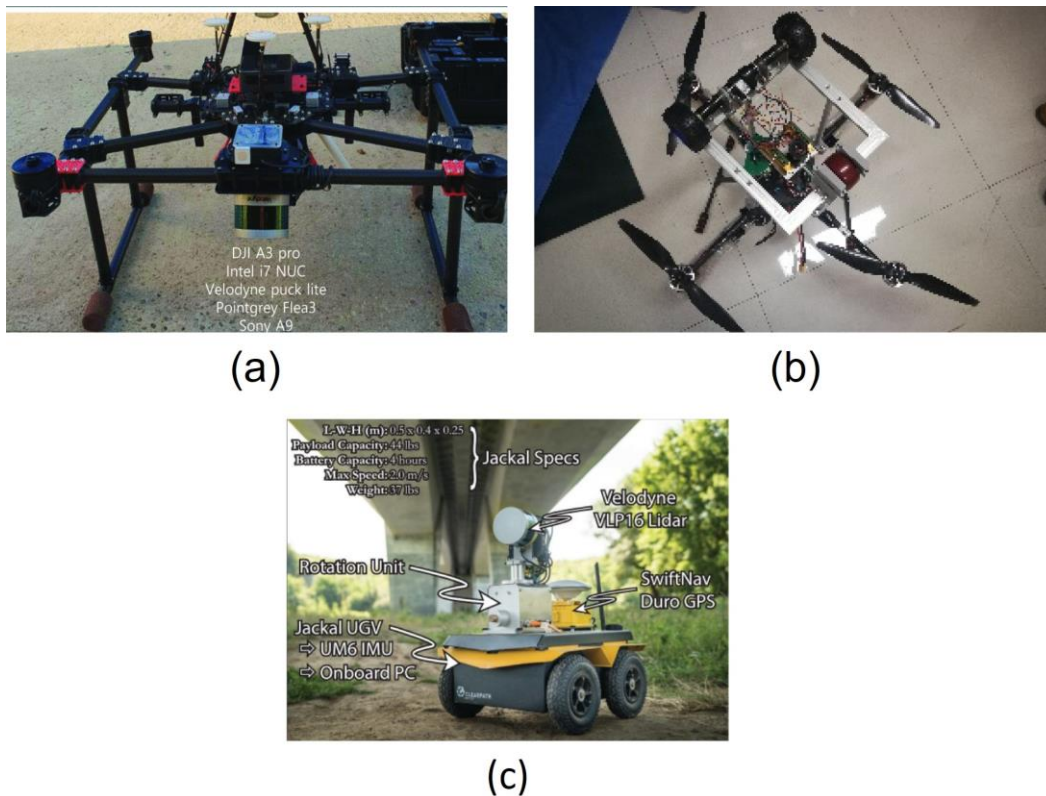


Figure 2.8 Bridge inspection robot; (a) UAV with multiple sensors [92], (b) UAV for bridge bottom inspection [93], (c) UGV for bridge bottom inspection [94]

Liu et. al designed a prototype of steel bridge inspection robot to find out the research challenges. The robot was a combination of rail kart and a manipulator equipped controlled by a joystick [93]. Rui Wang and Youhei Kawamura designed and experimented a robot with a magnetic wheel for steel bridge inspection. Magnetic wheel provided 2 advantages, it can climb vertical steel structures and

can create magnetic field for sensing [94]. Combining a specially designed car, guide rail and robot, a robotic system was proposed with a user interface for controlling it from long distance [95].

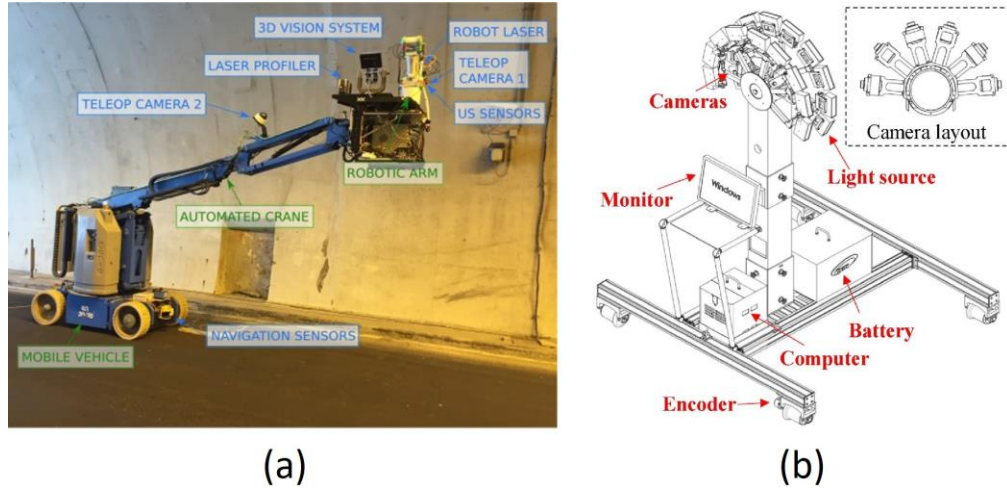


Figure 2.9 Tunnel inspection robot; (a) ROBO-SPECT [74], (b) MIT tunnel inspection robot [99]

Crack in the wall is a common defect for tunnel. Zhang et. al presented tunnel wall surface crack detection and classification techniques based on camera [96]. A semi-supervised computer vision based robotic system, named ROBO-SPECT, with multiple degree-of-freedom was designed by Menendez et. al on a road tunnel [72]. As shown in Figure 2.9(a), with the help of robotic arm, the robot can inspect the crack on the surface wall with an ultrasonic sensor. Huang et. al used deep learning based semantic segmentation of image for detecting crack on surface [97]. Moving Tunnel Inspection (MTI-200a), shown in Figure 2.9(b), was used to acquire data which was later analysed using deep learning algorithm.

2.2.1.3 Vision based inspection method in railway

With the continuous growth of railway mileage and requirements of intelligent digital infrastructure management, the efficient track inspection and survey is crucial and highly demanded. Traditional vision-based track inspection methods use 2D imagery for railway infrastructure inspection or monitor, which provides fast and high-resolution inspection for specific track fastener [82], [98], rail-head

[99] or tunnel [100]. However, these methods were designed for single specific track component or fault by capturing without depth information with fixed angle cameras. The laser scanning or Lidar is another option with depth sensing that can provide structural measuring and reconstruction. The generated point cloud projection can cover different structural surfaces and represents deformation and geometry faults. Stein et al. [101] used template matching and spatial clustering to process the laser scanning data of the railway and obtained the rail detection results through quantitative and qualitative experiments. Furthermore, a retrofitted industrial manipulator can detect and localize small components using a specially designed end of arm tooling using multiple vision sensors and force sensors [31]. On the other hand, [102] reports a railway tunnel inspection system with vehicle-based laser scanning that generates good quality mapping of tunnel internal wall. However, laser scanning cannot capture the detailed data like surface condition for small fault or defect and its point-cloud resolution is always mediocre within sensing distance. And traditional fixed laser scanner can hardly cover the whole 3D surfaces of a complex object. For example, it is difficult to cover the side and foot surfaces of the rail steel with a single scanner mounter on the train bottom. Therefore, a 3D reconstruction technique capturing high-resolution data that can manage complex shape of the track components is limited and drawing high attention.

3D models of the target object can provide more information compared to 2D inspection methods which can be used to find the defect or anomalies on the side of the rail [103]. For example, 3D model of a sub-millimetre pipe can be created using existing miniature videoscope camera and Structure-from-motion (SfM) technique [103]. In many cases, 3D scanning can provide better results, such as, industrial parts inspection in a noisy environment [104]. Similarly, 3D models of multiple railway bridges were created using 3 different optical digital modelling methods and it was concluded that 3D reconstruction method using camera provides more portability and cost benefit than laser scanning and infrared scanning [105]. Additionally, 3D model of the railway track can improve the rail inspection techniques [103], [106].

2.2.2 Robotic repair technology in railway

Inspections are important in the railways asset management to verify the integrity of assets and to prevent catastrophic failures. For example, in the UK, once a defect is detected on the track, details of the defect is recorded in the Track Geometry Reports (TGR). Based on the Network Rail standards (NR-L2-TRK-001-MOD11) Infrastructure Maintenance Engineer (IME) and Track Maintenance Engineer (TME) categorize the defects in Immediate Action Level (IAL), Intervention Level (IL) and Alert Level (AL) faults [107]. Both IAL and IL defects require repair without delays while AL defects can be schedule. Repair tasks in railways heavily vary, such as bolt or clip removing or fastening, grinding rail surface, welding, removing obstacles, demolishing trackside vegetation, fixing overhead electric lines, cleaning snow or leaf from the track, cleaning rolling stock, repairing rolling stock, etc. Railway repair dramatically depends on human involvement and specially designed tools and machines. Most of the repair task is done in outdoors despite of adverse weather condition. Adverse weather and heavy loads create fatigue for the human worker and pose risks to their health and safety. Though robots are used in many inspection tasks around the railway network, it is rarely used in any repair task. This section will conclude all the repair robots-related literature despite focusing only on the track. Even after scrutinizing the literatures rigorously, very few examples of repair robots have been found which are mostly used in cleaning tasks and are related to the repair task of rolling stock.

The First attempt to determine the usability of robots in repair tasks was carried out in 1987. After considering 24 technically viable processes in locomotive rebuilding, the research concluded that most of the processes are neither economically beneficial nor productive [108]. 2 years later, carrying out a separate study in Toronto Subway, Wiercienski and Leek arbitrated that robots are economically and technologically advantageous for rollingstock cleaning [109].

Four types of cleaning robots were introduced by East Japan Railway to tackle the lack of cleaning staff problem [110]. Those robots were equipped with

ultrasonic and laser sensors for navigation and obstacle avoidance. Another separate type of robot was developed to clean the bonnet of the Shinkansen high-speed train. An eight-axis robotic system was developed to replace 6 cleaning workers. Xu et al. designed and demonstrated a self-traction-model for cleaning the vehicle used for urban mass transit in China. The robot was designed with a mobile platform and a manipulator. Apart from the path planning and control algorithm, the researcher also developed an efficient energy and water management algorithm [111]. Tomiyama et al. discussed the requirements and conceptual design of a rolling stock front cleaning robot [112], and a prototype was built to analyse the cleaning time and applied force at the cleaning head [113].

Apart from cleaning, cutting, welding, grinding, or material handling are some of the repair tasks carried out by robots. Vale, a mining company in Brazil, developed a robot to repair the rail cars used for the mining operation. As shown in Figure 2.10, the system had multiple manipulators capable of cutting and welding carbon steel [114]. Wheel set of rolling stock, the combination of 2 wheels with a shaft, are very heavy and require expertise for repair. NSH USA, formerly known as SIMMONS Machine Tool Corporation, utilized robot and overhead cranes to completely remove the human involvement in transporting the wheel set in different workshop locations [115].



Figure 2.10 Robot used for repairing rail car in mining operation [116]

A novel technology has been developed for in situ rail defect repair which is cost effective and semi-automated. Discrete Defect Repair (DDR) process removes the defected part of the rail head and then repair the defected rail using semi-automatic arc welding [116]. Finally, a milling tool has been used to grind the rail profile and quality has been ensured with automatic inspection process. Figure 2.11(a) shows the railhead repair machine while (b), (c) and (d) shows the intermediate process of DDR technology respectively.

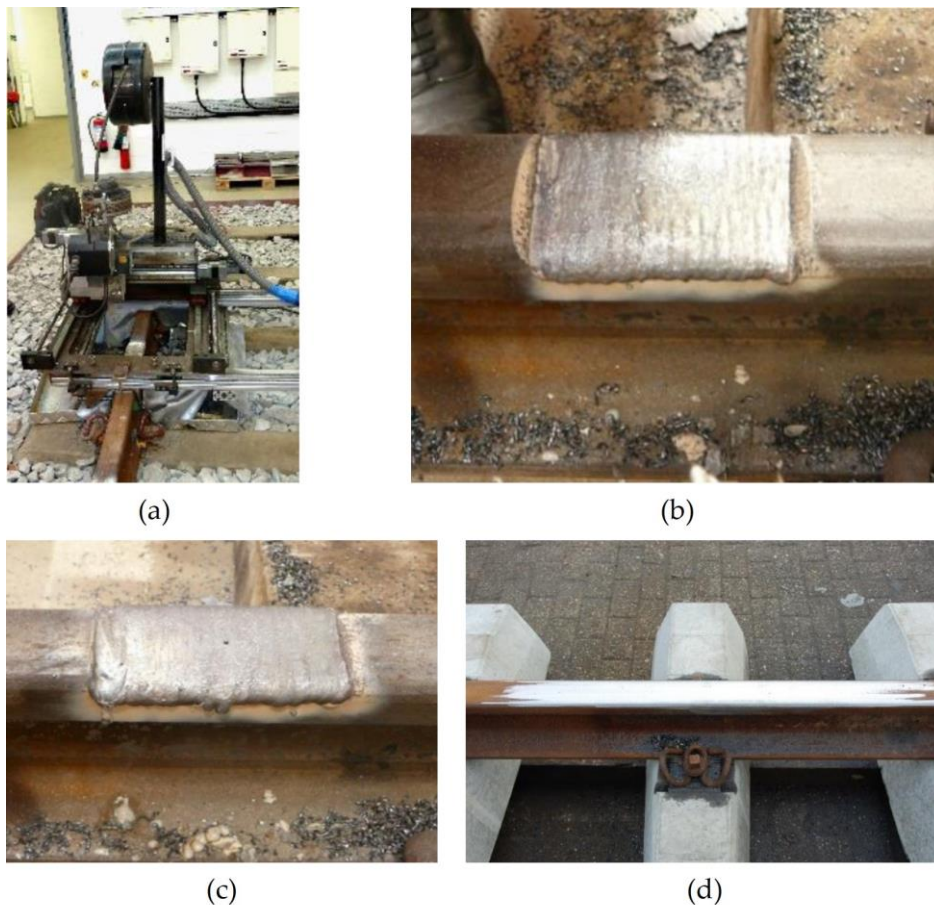


Figure 2.11 DDR technology [118]; (a) the machine, (b) cavity created after removing defected rail head, (c) rail head after arc welding, (d) final repaired rail head

2.3 Robotic maintenance technology in other industries

Mobile manipulator is a remarkable area of research for robotics, control, and automation enthusiasts. Simple search in indexing website will provide thousands of research articles in several types of research area such as controlling strategy, designing, simulation, safety and cost analysis, motion planning, navigation, and perception capability, etc. To match this specific research's objectives, search criteria have been narrowed down to "mobile manipulator" and "maintenance". After scrutinizing the search results, 74 pieces of research literatures have been selected which discuss the application of mobile manipulators specifically for inspection and repair tasks in various fields for the last 20 years, from 2002 to 2022. Furthermore, after reviewing the selected literatures, mobile manipulators have been classified based on the mobility features and application area.

2.3.1 Overview of mobile manipulator based on mobility and applicability

In the beginning, the concept of a mobile manipulator started with attaching a manipulator with a wheeled robot. But in many cases, the wheeled robot's navigation capability is limited because of the design and construction. Also, the ground mobile manipulator can only perform manipulation tasks at a certain height. Instead of using an Unmanned Ground Vehicle (UGV), Unmanned Aerial Vehicle (UAV) can extend the capability of a mobile manipulator at any height and traverse terrain. Moreover, a mobile manipulator can operate underwater with proper design and fabrication. Apart from the ground, air, or water, mobility features are required for mobile manipulators used in space or overhead high voltage power transmission lines. Based on the mobility features, the mobile manipulator can be divided into several categories: ground, aerial, marine, space, crawling, etc. After analysing all the selected mobile manipulator-related literatures for this research, they have been divided into multiple categories based on their mobility features, as shown in Figure 2.12. More than half of the mobile manipulators are ground-based, while 20% are aerial navigation capability. Crawling moving features are beneficial for climbing the wall or navigating along the power transmission line.

Mobile manipulator based on movability features

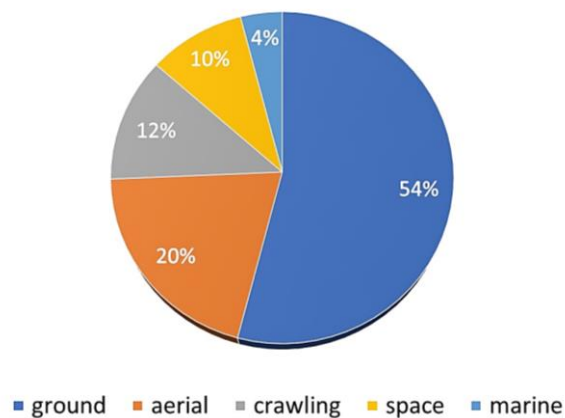


Figure 2.12 Percentage of mobile manipulator based on mobility features

The mobile manipulator has enormous potential to reduce human involvement in many dangerous tasks. As sensor technology is upgrading daily, the mobile manipulator is becoming more capable of performing many monotonous and repetitive tasks, which might pose a risk to human health and causes financial losses for the company. A mobile manipulator can be used for various inspection, surveillance, or repair task. For example, a mobile manipulator can do dangerous tasks such as surveillance or inspection of a nuclear reactor or performing inspection or repair at an overhead high-voltage power transmission line. Moreover, with a proper design and navigation mechanism, the mobile manipulator can perform inspection and repair tasks inside and outside a pipe or clean a window. Based on the area of application, selected literatures are divided into several categories as shown in Figure 2.13.

As the mobile manipulator provides a boundless workspace because of its mobility capabilities, it can be used for various types of applications such as inspection and repair of a wind turbine, industrial and indoor sing inspection, patrolling inside a building, cellular tower inspection, grasping an object, drilling, or screwing, etc. Among the selected literatures, the highest percentage (20%) of the mobile manipulator is used for inspection and repair tasks in the high-voltage power transmission line. Nuclear reactor inspection is another risky task for humans as this poses severe health and life risks. A notable percentage, 12%

Mobile manipulator based on area of application

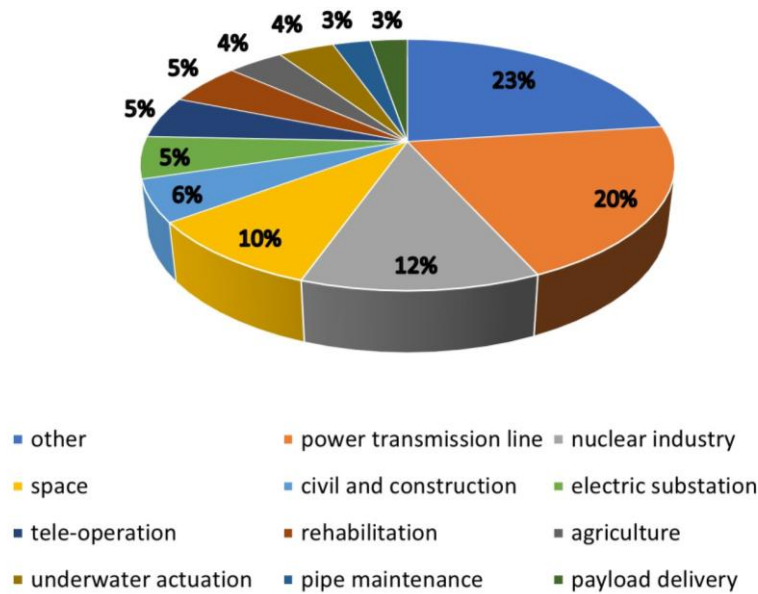


Figure 2.13 Classification of mobile manipulator based on area of application

of the mobile manipulator, is used to find a defect in nuclear reactor vessels or other infrastructure in nuclear power plants. Another significant area of application is the inspection and repair of spaceships. Inspection, surveillance, and repair of civil infrastructure and construction sites, electric substations, teleoperation, and rehabilitation accounts for the same amount.

2.3.2 Mobile manipulators for inspection and repair tasks

2.3.2.1 Maintenance of high-voltage transmission line and substation

High-voltage power transmission is an unsafe workspace for a human worker not only because of height but also because of electricity. Humans had to work in a very unergonomic workspace and suffer from severe health issues. A mobile manipulator can solve this problem by substituting or assisting humans in inspection and repair tasks. These mobile manipulators require a unique moving mechanism to navigate or move along the power transmission line. Also, initially, they are lifted and placed manually on the power transmission line. A semi-autonomous teleoperated mobile manipulator has been designed to inspect the aircraft warning spheres of the power transmission lines [117]. Sensor-Arm is a

mobile manipulator that can move along the transmission line by sliding mechanism and perform the inspection using a high-quality camera [118].

Like the overhead power transmission line, underground power transmission lines provide a risky and inaccessible workspace. Therefore, a robotic system has been designed to inspect and repair the switch in the underground transmission line, as shown in Figure 2.14. The robot consists of a vehicle equipped with a 5 DoFs hydraulic arm which holds a 6 DoFs manipulator [119], [120]. A hydraulic tool changing mechanism also been developed for changing the end-effector to perform various tasks.

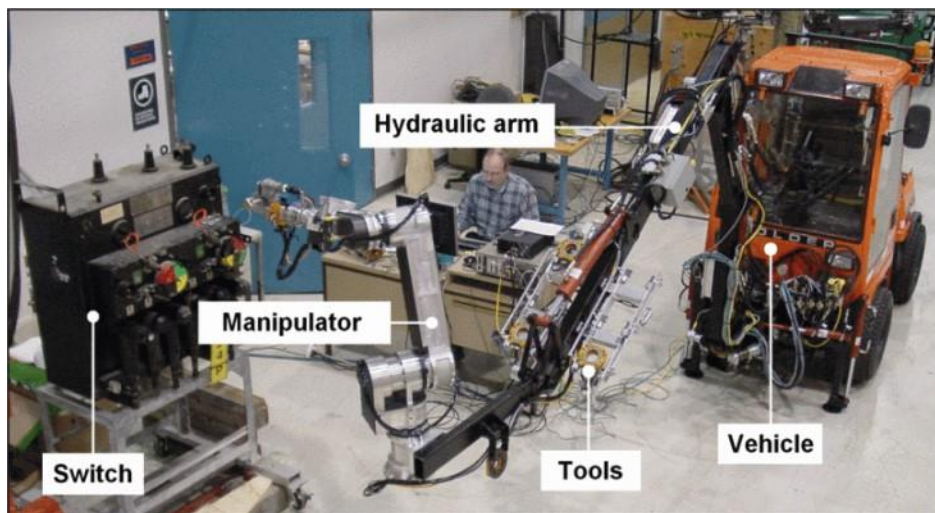


Figure 2.14 Specially designed mobile manipulator for underground power transmission line, reproduced from [117, 118]

LineScout, a tele-operated robot has been proposed for live transmission lines with rolling wheel capable of obstacle passing attached with a programmable pan and tilt camera [121]. Another live transmission line inspection robot has been proposed in [122]. Two locomotion strategies provide obstacle-crossing capability: rolling on the wheel and crawling with the arms. Obstacles in a power transmission line include compression slices, vibration dampers, spacers, suspension clamps, etc., and the capability of crossing these obstacles is a desirable feature [123]. Simulation and experiment of a mobile manipulator for adjusting the vibration damper in the transmission line has been done in [124].

This robot can hang on the line with the arm and move along the line with the rolling wheel, as shown in Figure 2.15(a).

Apart from inspecting the power transmission line, mobile manipulators can perform various actuation and repair tasks. An autonomous live-line maintenance robot (ALMR), shown in Figure 2.15(b), can strip and clip wire clips on a 10kV transmission line [125]. Extra-high-voltage (EHV) uses horizontal and suspended insulators, and Zhang et. al proposed a mobile manipulator with a creeping mechanism that can move along horizontal and suspended insulator strings [126].

Aerial mobile manipulators can be another option for overhead transmission line maintenance as aerial robots can easily reach different heights and navigate in the air without any special kinetic or driving mechanism [127], [128]. POSITRON,

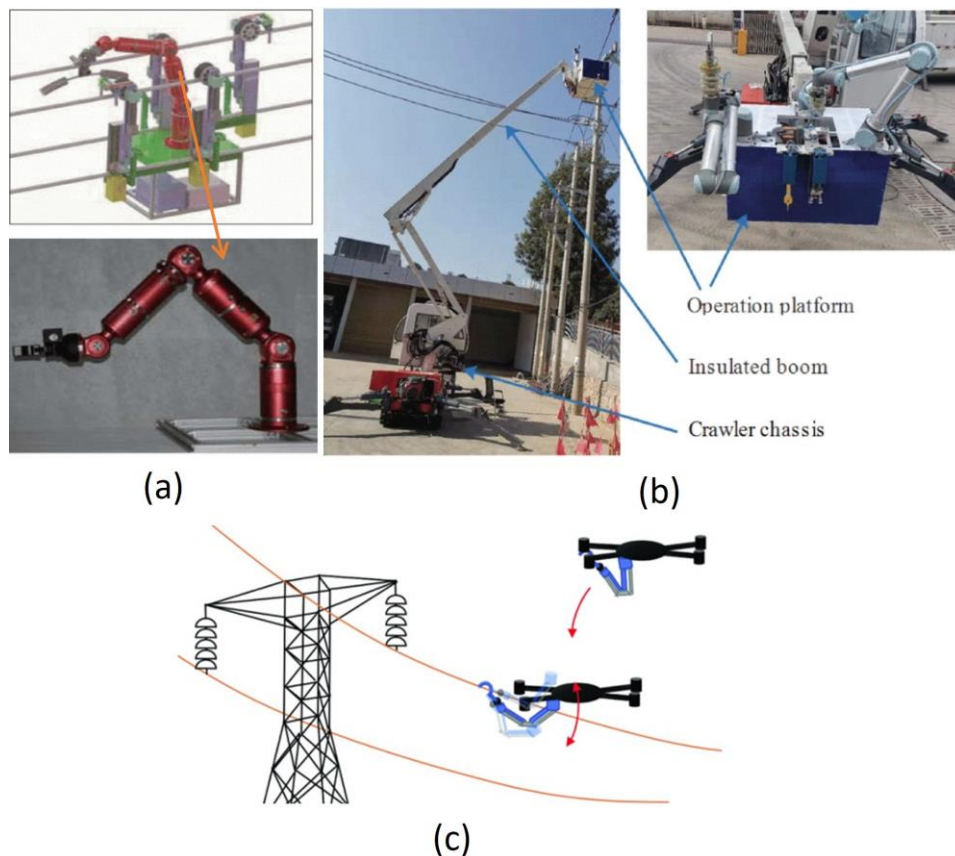


Figure 2.15 Mobile manipulator for maintenance of power transmission line; (a) for vibration damper maintenance [122], (b) Autonomous live-line maintenance robot (ALMR) [123], (c) POSITRON [125, 126]

as shown in Figure 2.15(c), is a conceptual design and simulation for controlling the aerial mobile manipulator in a GPS-denied environment.

The substation is an integral part of electricity transmission. Although most of the equipment in the substation is installed mostly in the outdoor environment, substation components are heavily affected by environments and weather conditions. In contrast, some are installed inside an infrastructure. Therefore, indoor and outdoor components require scheduled maintenance and safety checks. Moreover, most of the maintenance in the substation is done on live wires or equipment, which poses a risk for humans. Therefore, research has been carried out by Want et al. to find out the suitable robotic technology for combined maintenance machines for both indoor and outdoor [129].

Zhang et al. designed, fabricated, and experimented with a mobile manipulator for the maintenance of outdoor equipment of the substation. The robot move by crawling, and the manipulator can clean and remove ice from the insulator and repair broken wires [130], [131].

2.3.2.2 Maintenance in nuclear power plants & hazards

Nuclear materials, the mining industry, or any container with radioactive material are hazardous for humans. Minor exposure to these hazardous, radioactive materials may damage the cell structure inside the human body and create cancerous cells in the long run. An elevated level of exposure will create severe health effects such as skin burns or radiation sickness, leading to death. Though many protective gears are available to minimize the risk of exposure to hazardous materials, the mobile manipulator can replace humans working in a hazardous environment and keep valuable human life safe.

A wheeled mobile manipulator has been designed to monitor radioactive material [63], called Radiation Protection Assistant Robot (RPAR-I), which is equipped with a camera for safe navigation and can be controlled by a remote which protects the worker from exposing themselves to radiation. Figure 2.16(a) shows 4 DoFs manipulator attached to a 2-wheel drive robot base. Instead of teleoperation, many researchers have taken a dual approach to design the

control system, considering both autonomous and remote-controlled modes [132], [133], [134]. A semi-autonomous system, CERNbot [133], has few pre-defined standard operating procedures, shown in Figure 2.16(b). CERNbot uses an autonomous mode for inspection and navigation and an operator-friendly Human-Robot Interface (HMI) for telemanipulation. As all these robots have teleoperation capability, these robots are also equipped with computer vision systems and have data transfer capability.

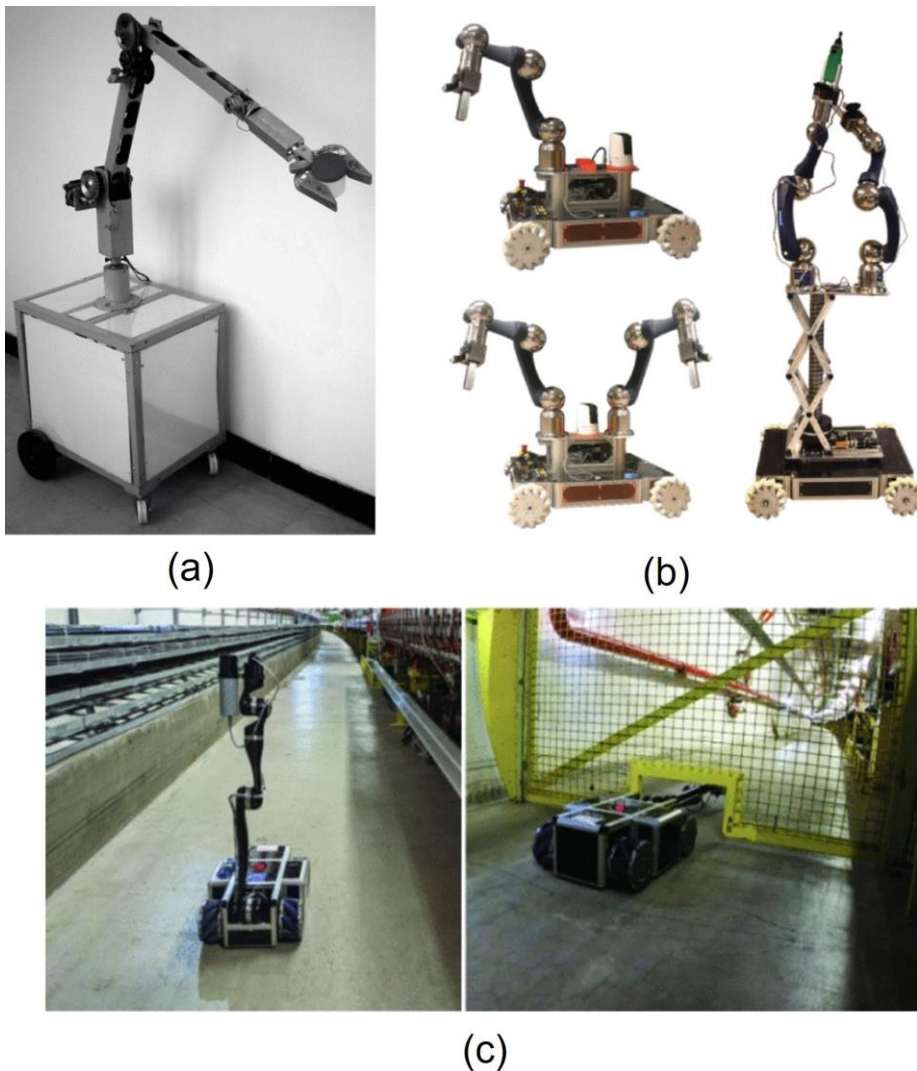


Figure 2.16 Mobile manipulator in nuclear industry; (a) RPAR-I (Radiation Protection Assistant Robot) [63], (b) CERNbot with single and dual arm configuration [131], (c) MIRA SPS inspecting and navigating through a narrow passage [134]

Mobile manipulators can perform multiple tasks as sensor technology and controlling algorithm advances. For example, Multifunctional Inspection and Rescue Robot (MIRR) has been engineered to execute an inspection, maintenance, and rescue operation as the Remote Handling (RH) system in the tokamak fusion facility [135]. Equipped with 2 manipulators: Operation Robot Arm (ORA) and Inspection Robot Arm (IRA) on an Omnidirectional Mobile Vehicle (OMV), MIRR can navigate in a narrow space by avoiding multiple obstacles and performing different tasks. Another omnidirectional mobile manipulator, MIRA SPS, as shown in Figure 2.16(c), can detect and navigate through a narrow passage in a semi-structured environment leveraging computer vision strategy [136].

In many cases, manipulation is required underwater [137] or in the air [138], or inside a steam generator [139]. For example, the underwater mobile manipulator designed in [137] can detect and remove unwanted objects from the bottom of a nuclear reactor vessel. In contrast, the lightweight aerial mobile manipulator proposed in [138] is experimented for accessing hard-to-reach and hazardous locations. The aerial mobile manipulator is equipped with a manipulator and can deposit material through a small-scale extruder to repair a crack, as shown in Figure 2.17.

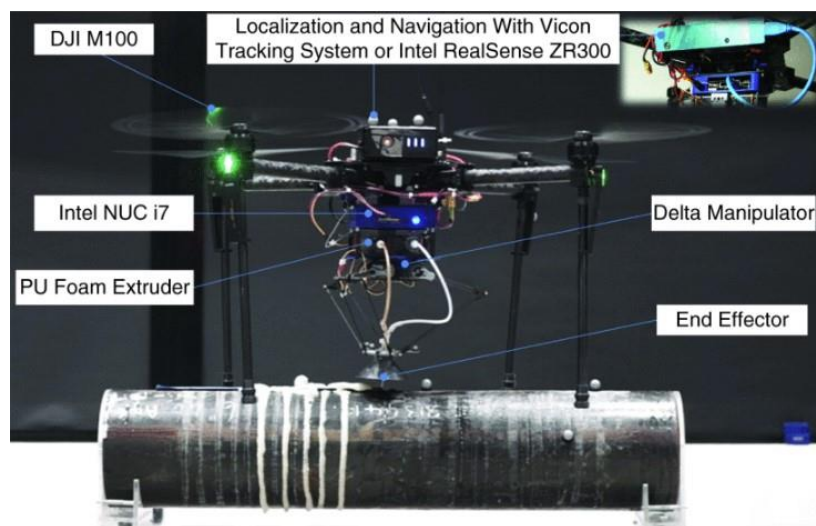


Figure 2.17 Aerial mobile manipulator extruding PU foam to block a crack in surface [136]

2.3.2.3 Maintenance in aerospace

Workspaces and environments are different in space. Ground, aerial or underwater navigation systems are not usable for mobile space manipulators. That's why the researcher has developed free-flying or free-floating mobility features for mobile space manipulators [140], [141]. Dual Arm Free-flying Space Robotic Systems (DAFSAR), shown in Figure 2.18(a), can estimate the target object's pose, self-dock on the host, and perform robotic maintenance.

To increase the repair capability and real-time on-orbit assessment, researchers designed a mobile manipulator called Autonomous Mobile On-orbit Diagnostic System (AMODS), which is a combination of multiple CubeSat (Cube shaped Satellite) with 7 DoFs manipulators [142]. This robotic system consists of 2 satellite systems, 1 for repair (RSat) and 1 for navigation (BRICSat), as shown in Figure 2.18(b).

The Control system of the space mobile manipulator requires special attention as a free-floating mobile base, disturbance from the host spacecraft, and end-effector motion can affect the pose of the whole robot. A coordinated control system considering the attitude of the base robot and the motion of the dual arm manipulator system has been investigated [143]. Alternatively, Dongming et al.

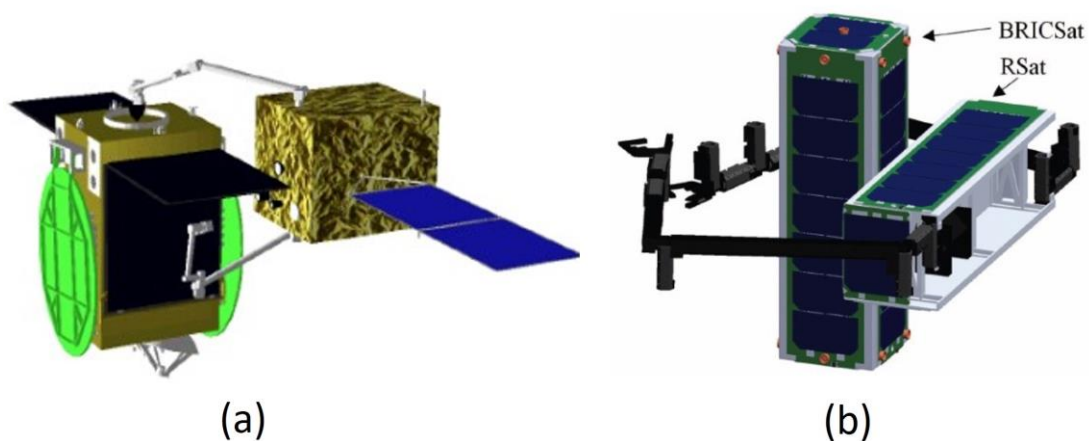


Figure 2.18 Mobile manipulator in space for maintenance application; (a) DAFSAR space mobile manipulator [138, 139], (b) AMODS [140]

discussed an impedance-based control system by designing the end-effector motion as a mass-damper-spring system for grabbing the target [144]. Another impedance-based control system has been investigated for robot docking on spacecraft [145].

Tele-operation is the most commonly used control system for mobile space manipulators [146], [147]. Instead of GUI or HMI-based teleoperation, a VR system controller has been used to control the mobile manipulator ReMoBot.

2.3.2.4 Maintenance in civil and construction sites

Civil infrastructures such as buildings, roads, bridges, etc., require scheduled inspection and repair to ensure the safety and comfort of the users. Inspection and repair technology for civil and construction infrastructure requires human involvement, which in many cases is risky and cost ineffective. Though robots have been used in civil infrastructure since the early 2000s [148], [149], [150], they were mostly used for data collection for inspection but not intended to perform any repair work. The first mobile manipulator came up in research for proper repair work in 2013. Liu et al. demonstrated a mobile manipulator for dry friction-based drilling for bridge deck repair tasks [151].

Nguyen et al. designed a civil infrastructure inspection and evaluation with a mobile manipulator with a combination of computer vision and thermal camera, shown in Figure 2.19. As the inspection system was attached to the 4 DoFs manipulator end, it ensured efficient data collection for crack detection [152].

Apart from the ground-based system, the aerial mobile manipulator can be used for civil infrastructure inspection and repair, as many places are hard-to-reach with a wheeled robot. Collaborative aerial mobile manipulators have been proposed for crack detection and repair using AI and 3D printing [153]. Usually, commercially available drones and UAVs can only carry a lightweight payload of less than 1kg. Repair work in civil infrastructure requires more payload capacity. To solve this capacity issues, research has been carried out to design and fabricate an aerial manipulator capable of carrying a high payload [154].



Figure 2.19 Proposed mobile manipulator system for with multiple sensors [150]

2.3.2.5 Application of mobile manipulator in agriculture

This research mainly focuses on applying mobile manipulators in different areas, especially for inspection and repair technology. There are more mouths to feed as the population increases, but the amount of cultivatable land and workforce is decreasing [155]. Though agriculture does not need conventional inspection and repair, it demands precise and efficient robotic solutions to increase food production with fewer resources and workforce. For example, a dredger robot can perform maintenance tasks such as removing debris, weeds, and sediments [156]. Knowledge of these precise and efficient mobile manipulator technologies can be used for performing maintenance tasks in other sectors.

Mobile manipulators can replace and assist humans in picking fruits or vegetables require much human workforce. For picking fruits and vegetables, mobile manipulators need one or multiple arms. They should have the capability to locate the plant, determine the ripeness of fruits, navigate autonomously and pick the fruits or vegetables without damaging them [157], [158], [159], [160]. As shown in Figure 2.20(a), most of these mobile manipulators in the agricultural application are made of a ground-based mobile base robot with vision system for fruit detection.

Another complex task for the mobile manipulator is detecting weeds and spraying pesticides while navigating traversing terrain. Vineet et al. proposed a pesticide-

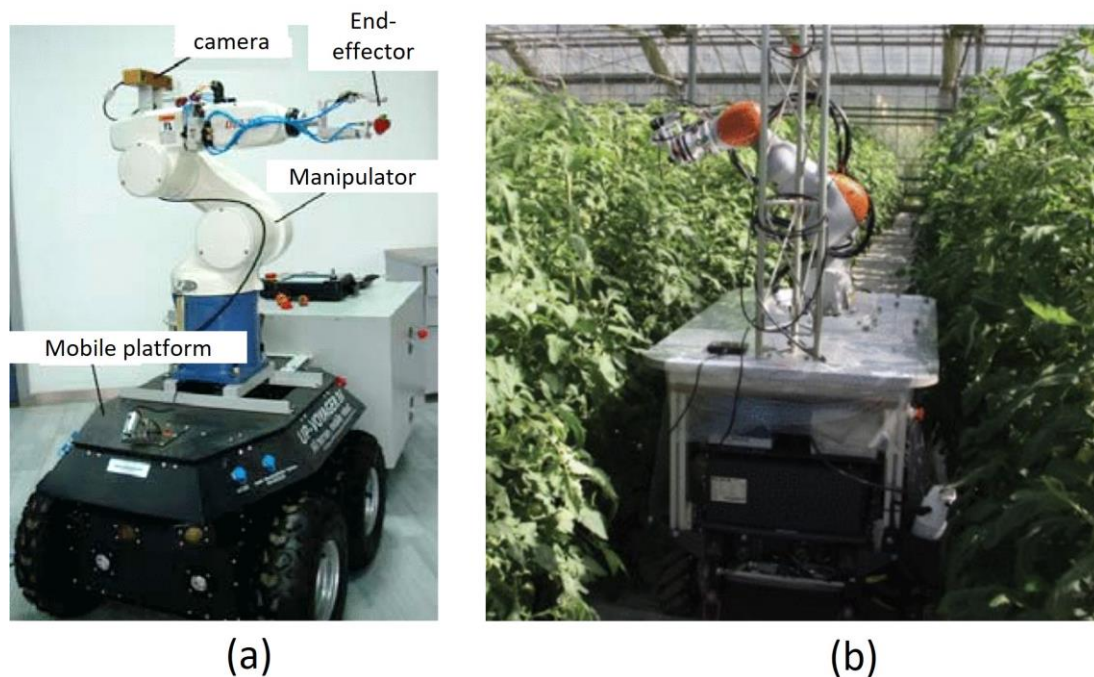


Figure 2.20 Mobile manipulator for agricultural application; (a) strawberry picking robot [155], (b) pest detection and spraying robot for a greenhouse [160]

spraying robot with a skid-steering mobile manipulator [161]. To improve the navigation capability in traversing terrain, tracked wheels are used instead of a pneumatic wheel. Martin et al. demonstrated a mobile manipulator with Robot Operating System (ROS), shown in Figure 2.20(b), fitted with multiple sensors for pest detection and spraying inside a greenhouse [162].

2.3.2.6 Other applications of the mobile manipulator

Mobile manipulation is an issue of special importance for the researchers of robotics. Industry 4.0 has paved the process of multiple automation techniques on the workspace and business model to increase efficiency and productivity by ensuring safety. Researchers are trying to improve the usability of a mobile manipulator to meet the demands of the industry by introducing more functionality in the mobile manipulators. Conceptual designs of mobile manipulators helping humans in the industry surfaced in the research community in early 2000 [163], [164]. Currently, many companies offer commercially available mobile manipulator platforms that can be reprogrammed for various use cases [165],

[166]. As the versatility, accuracy, environment perception from sensors, and collaboration capacity is enhanced, areas of application for mobile manipulators are also widening. Table 2.3 shows the diversity of application areas for a mobile manipulator.

Table 2.3 Different areas of application for mobile manipulators

Application	Type of base robot	Sensing method	Key technology and contribution
5G tower maintenance [167]	aerial	camera	Tower inspection and repair, the study of manipulator stability with wind disturbance
Drilling and screwing [168]	aerial	camera	Adaptive robust control system, real-time object detection
Maintenance in industry [169]	aerial	LiDAR, stereo camera, GNSS	Multidirectional thrust controller
Payload delivery [170]	aerial	RGB-D camera	Heavy load control, learning-based planning for aerial cooperation
Pipe inspection [171]	aerial	external motion capture system	Multidirectional thrust control flying,
Window cleaning of building [172]	Magnetic climbing	Force sensors	Elastic actuators, passive obstacle avoidance
Archive room patrolling [173]	ground	Infrared camera, laser	Storage basket, monitor abnormal temperature
Assembly task [174]	ground	2 cameras	Stretchable manipulator, assembly in space
Indoor sign inspection [175]	ground	camera	Scissor mechanism for manipulator,
Payload delivery [176]	ground	Multiple camera	Extended payload carrying capability
Penstock inspection [177]	ground	Camera, laser	Wall climbing capability

Pipe inspection and repair [178]	ground	4 cameras	Navigation inside pipe, contamination-less rehabilitation (CLR) of pipe
Garbage collection and sorting [179]	ground	RGB-D camera	Deep learning for garbage type identification
Bomb disposal [180]	ground	2 cameras	Tele-operation
Undersea environments [181]	underwater	Multiple cameras	Deep underwater operation
Autonomous underwater vehicle [182]	underwater	camera	Control system similar to spaceflight, user-in-the-loop over low data transmission
Seabed inspection, repair [183]	underwater	camera	Free-floating actuation, a convolutional neural network for object recognition

2.4 Research Gaps

Current maintenance technique in railway is heavily dependent on humans, which poses severe health and life risks to the operator. This dependency on humans also reduces the efficiency of the rail network. Mobile manipulator, a special type of robot, can perform different types of tasks because of flexibility and dexterity. It is obvious from the literature review that various industries are using mobile manipulator to perform many dangerous and dull inspection and repair tasks to improve the safety of human and gain financial benefits (section 2.3). The economic advantage of using robotics and automation in rolling stock maintenance and cleaning-related tasks is verified. Robots have been used in the manufacturing industry for a long time to perform precise and continuous tasks. In recent years, the cost of industrial robots has been declining. According to ARK Invest's research in 2019, there will be a 50-60% drop in robot prices by 2025, and 10 times more robots will be used in industries in 2025 compared to 2017 [184]. Multiple gaps were found after scrutinizing the previous and available research items which worked as the motivation of this thesis.

2.4.1 Lack of multipurpose robot

Though modern robotic technologies provide many benefits, railway industry is slow to adopt these modern technologies. Currently, inspection and repair are performed separately. While there are many modern maintenance technologies available, such as RRV or train-borne systems, they are designed to perform specific tasks, very costly to operate or require human intervention in each step (section 2.1). Furthermore, there are a significant number of robots available for track inspection, but a very few are available for repair (section 2.2). In the case of the current repair technique, separate specialized vehicles or tools are required for the repair task.

Mobile manipulator provides dexterity and mobility which increases the area of workspace and actuation at long distance. Another advantage of a mobile manipulator is that it not only detects the defect but also can capture the details of the surrounding defects. This increases the time for maintenance as inspection and repair are carried out in 2 different stages. As the railway environments are different from other industries, robots will face some unique challenges while navigating along the railway track (section 1.4). Hence, research is required to introduce the usage of mobile manipulator for railway track inspection and repair interventions. Design modifications, proper sensor attachments, improved inspection technique and repair task workflow requires much attention from the research community to adopt the mobile manipulator technology for railway track maintenance tasks. As an autonomous robot equipped with sensors, mobile manipulator can perform repair tasks and help the human to ensure the quality of the repair task.

2.4.2 Shortcomings of current vision-based inspection method

Sensor technologies are becoming very mature, which increases the scope of robots to reduce human involvement in inspection tasks. As the 2D perception of the defect is not sufficient in many cases to perform the repair work, 3D sensors can be used to improve the robotic perception. A cost-benefit analysis of railway tunnel showed that data acquisition is one of the bottlenecks for 3D reconstruction technique [185]. Using robots to gather data for 3D reconstruction can improve

the efficiency of data acquisition. A single fixed camera on a robot can provide imaging information in a fixed field-of-view [186] while an array of camera can provide multiple images from different field-of-view [103]. Though the later one retains better information for final output, array of camera requires separate transportation system.

Mobile manipulator can be used to improve the current vision-based inspection technique. A mobile manipulator equipped with a camera can acquire data autonomously from multiple field-of-view to reduce the human involvement in inspection method using 3D reconstruction. Moreover, active sensors such as LiDAR can provide sensing of the surrounding environment, while GPS can provide a location on a global scale.

2.4.3 Necessity of using robot for railway track repair tasks

It is important to find the usage of modern robotics technology for rail repair to keep pace with the increasing demand and ensure a safe and comfortable journey. Moreover, adverse weather and heavy loads create fatigue for the human worker and pose risks to their health and safety. The economic benefits of using a robotic system are already proven. For example, using robotic technology for rolling stock repair can provide financial advantages to rail companies [30]. Though robotics and autonomous systems have been used in the railway sector for more than a decade, most of them are restrictively used in rolling stock repair and most of them are without manipulation capability [31]. Though robots are used in many inspection tasks around the railway network, it is rarely used in any repair task. Even after scrutinizing the literature rigorously, very few examples of repair robots have been found which are mostly used in cleaning tasks and are related to the repair task of rolling stock (section 2.2).

Intelligent digitalization in railway repair is one of the objectives of the IN2SMART2 [187] projects in the European Shift2Rail program, which aims to bring the robotic strategy to improve task decision support and scalable asset information management framework. To achieve this, inspection and repair of the rail track and related jobs are crucial to the life-cycle engineering and management of track assets from the concept design to disposal. However, as

the rail mileage grows promptly, the repair criteria are demanded more restrictively with technologies and tools improvement. Therefore, robotic repair systems are highly demanded by flexibly combining autonomous sensing, control, decision-making, remote communication, and work execution for railway track repair. Hence, this research will fill the void of railway track repair robot and will provide repair tasks completion framework for a mobile manipulator.

3 METHODOLOGY

The methodology is the procedure for completing a task with a specific intention. This chapter will provide a detailed description of the procedure for different tests, data collection, and analysis. In the beginning, the overall research methodology of this research will be discussed to ensure successful completion. Prior to diving into the main descriptions of the process, a brief description of the key sensors, software, and hardware architecture will be presented in section 3.2. Then in section 3.2.2, software architecture of this research will be discussed. This section will focus on ROS, MoveIt! architecture. Next, section 3.2.3 will introduce simulation software (Gazebo) for RIRS. A coding strategy based on systems engineering approach will be discussed in section 3.2.3. Afterward, 3 methods with necessary equations and techniques will be introduced for the rest 3 objectives of this research. Section 3.3, will discuss about the kinematics, repeatability, and precision of the RIRS which is related to the second objectives. This section will describe how the joints relate to each other in different navigation scenarios, repeatability of RIRS in finding defect location and precision of the UR10e manipulator for reaching the same defect position. An improved inspection technique will be presented for the third objective in the following section 3.4. This technique will consist of creating a 3D model using Structure-from-Motion (SfM) and then fusing it with RGB-D sensor and robot perception. Finally, the last part will illustrate the multiple repair task framework in section 3.5 for the fourth objective. This repair task framework is divided into 2 division; first one will involve humans, while the later task will be without human involvement.

3.1 Research Methodology

Research starts with knowledge gathering about railway terminologies, current railway maintenance techniques, and state-of-the-art robotic maintenance systems in railway and other industries. Purpose of this initial study is to understand the problem and setup the aim and objectives. Subsequently, the research gap of this thesis will be derived from the gathered knowledge during the literature review. The next step of this research is to finalize the methodology. The methodology will start by analysing the hardware, such as the joints of robots,

and how they relate to each other. Then, a vision-based inspection technique has been developed by combining the state-of-the-art data-rich 3D modelling technique and mobile manipulation. Later, multiple repair task frameworks have been proposed and tested. Subsequently, several tests have been conducted for the inspection and repair task, and data has been analysed. Finally, the thesis ends with a discussion of experimental results and future suggestions. Figure 3.1 shows the overall methodology of this research.

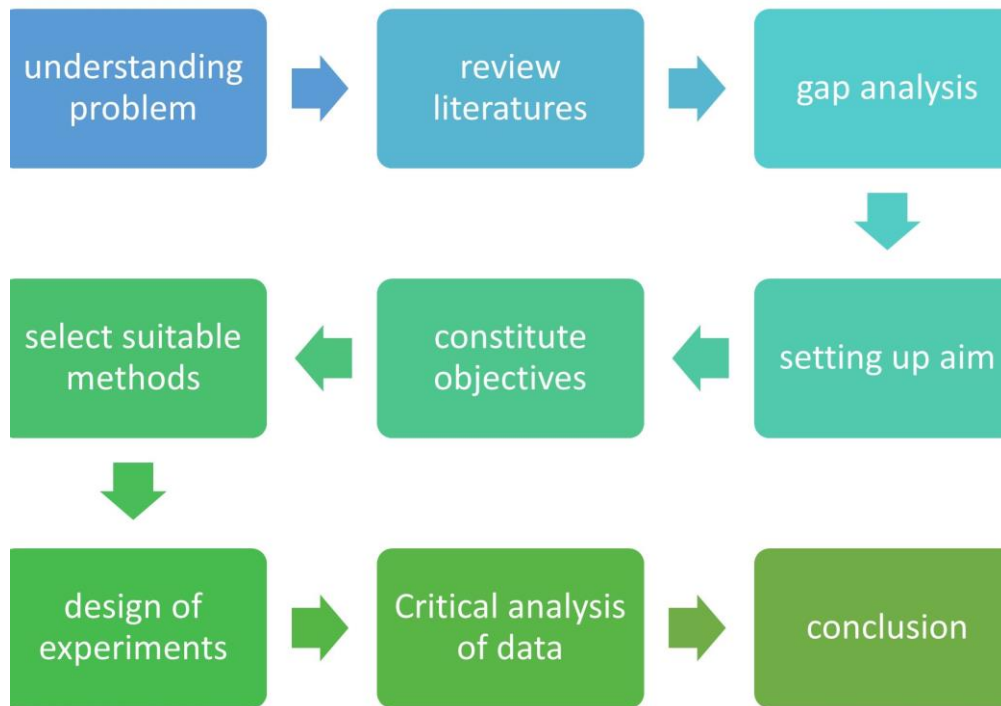


Figure 3.1 Overall methodology of this thesis

3.2 RIRS: Overview

RIRS stands for Robotic Inspection and Repair System, shown in Figure 3.2. It combines a mobile manipulator (warthog UGV and UR10e manipulator) and a specially designed trolley. Key specifications of the components of RIRS are given in Table 3.1. A specially designed trolley enables the RIRS to navigate properly on the track, as shown in Figure 3.3. Warthog UGV comes with an onboard computer with Ubuntu 16.04 operating OS and Robot Operating System (ROS) as middleware. RIRS has been designed to reduce human interaction on

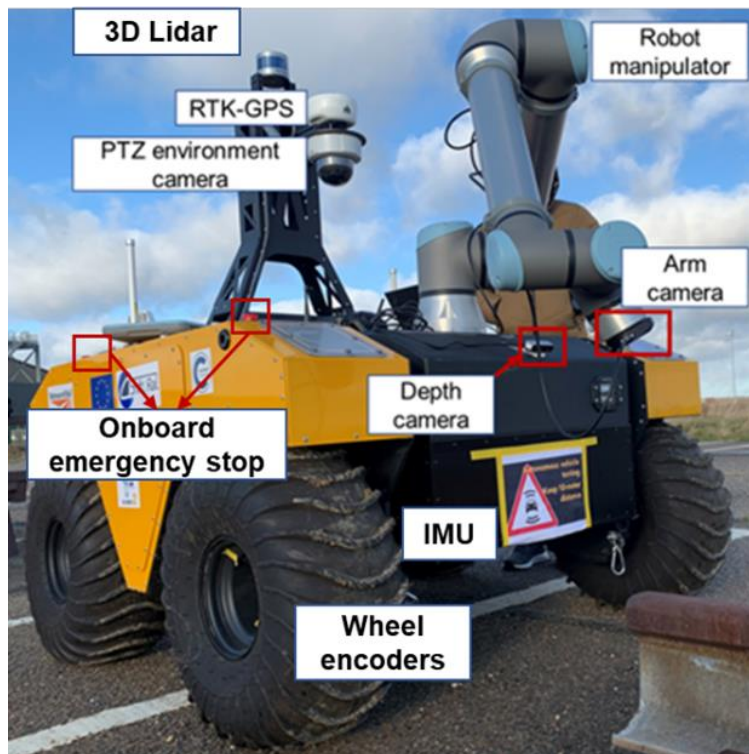


Figure 3.2 Sensors and systems of RIRS

the track during the inspection and repair job, improving the safety of humans working on the rail. Also, RIRS has been designed to improve the reliability of track usage by ensuring proper inspection and maintenance.

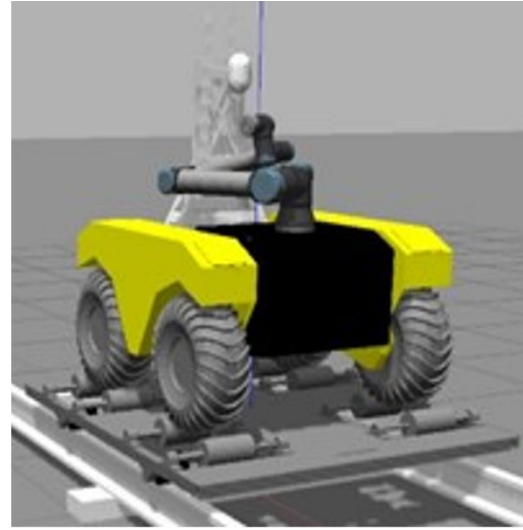
Table 3.1 Key specifications of RIRS components

Warthog	OEM: Clearpath Robotics Wheel number: 4, 20" AGV tyre Vehicle dimension: 1.52 x 1.38 x 0.83 m Drive type: Differential drive Payload capacity: Maximum 272 kg
UR10e	OEM: Clearpath Robotics Number of Joints: 06 Payload capacity: Maximum 10 kg Maximum Reach: 1200mm
Trolley	OEM: custom design and made. Dimension: 2.0 x 1.78 m Wheel Number: 04, 10" rail wheel

	Number of Contact Roller: 08 Power transmission: belt and pulley
--	---



(a)



(b)

Figure 3.3 RIRS performing on track navigation at RITA, Cranfield University and in simulation

3.2.1 Sensors

Various sensors have been integrated into the RIRS to conduct the inspection and repair process. Some sensors, such as LiDAR and camera, are required for robotic perception. In contrast, some sensors, such as IMU, GPS, and wheel encoder, are required for localization and navigation. LiDAR on the RIRS provides the 3D perception of the surrounding environment for safe navigation. At the same time, the RGB-D camera on the front of the RIRS gives the image and depth perception of the track and obstacles. A monocular camera at the arm tip ensures the detection of the defect and close inspection.

Moreover, the PTZ (Pan-Tilt-Zoom) camera, which can rotate 360°, renders the view of the surroundings to the human operator through the Human Machine Interface (HMI) or web services. This camera is not used for robot perception, hence not used in any experiments. But this camera can be used for remote

operation or monitoring. IMU can provide angular velocity and linear acceleration. GPS calculates the position of the RIRS in the global map, while the wheel encoder can provide the travelled distance from the wheel rotation and orientation. Table 3.2 below summarises the available sensors in RIRS.

Table 3.2 Summary of sensors in RIRS

Name	Model / Manufacturer	Key specifications
LiDAR	VLP-16 / Velodyne Lidar	Range: 100m Number of laser/receivers: 16 pairs Field of View (vertical): +15° to -15° Field of View (Horizontal): 360° Rotation rate: 5Hz to 20Hz
RGB-D camera	Intel Realsense d415i / Intel	Range: 0.3m to 3m Depth frame rate: Up to 90 fps Depth Field of View (FOV): 87° × 58° RGB frame rate: 30 fps RGB sensor FOV (H × V): 69° × 42°
RGB camera	Robotiq wrist camera / Robotiq	Maximum resolution: 5Mpx at 2fps Maximum frame rate: 30 fps at 0.3 Mpx Focus range: 70mm to infinity Autofocus: Yes
PTZ camera	M5525 / AXIS Communication	Resolution: 1920x1080 Maximum frame rate: 30fps Field of view (Horizontal): 61.8° to 6.7° Pan capability: 360° Tilt capability: 90° Zoom capability: 10x
IMU	GX5-AHRS / Microstrain	Sampling rate: 1000Hz Pitch-roll static/dynamic accuracy: ±0.25°/0.4°
GPS	SMART2	Compatibility: GPS, GLONASS, BeiDou, Galileo, SBAS, QZSS Maximum data rate: 20Hz Horizontal positional accuracy: DGPS: 0.4m RTK-GPS: 2.5cm

3.2.2 Softwares and tools

RIRS has been used as an experimental rig in this research. Prior to conducting experiments at Railway Innovation and Testing Area (RITA), codes and algorithms have been tested in simulation and indoor environments. RIRS uses Ubuntu 16.04 as the OS and open-source Robot Operating System (ROS) [188] as the middleware to perform a robotic operation. The Open-Source Robotics Foundation (OSRF) has recently begun to handle the upkeep of ROS and partner projects such as Gazebo (default simulation software in ROS) including creating new capabilities [189]. ROS provides a network in which all tasks are connected by a node. Any node in the system may access this network and communicate with other nodes[190]. Details of components of ROS communication architecture is discussed in appendix A.1.

Movelt! [191] provides multiple packages to perform manipulation in ROS. It provides various functions such as motion planning, manipulation, 3D perception, kinematics, collision and obstacle, control, and goal of the manipulator. Moreover, it can be easily integrated with the Rviz [192] visualizer to check the manipulator's trajectory to ensure safe operation before deploying it in the real world. Appendix A.2 will discuss more about the integration between ROS and Movelt!.

3.2.3 Simulation

Simulation is the digital representation of the real world. Robotics simulation software is a digital tool for robotics developer to visualize the impact of the robotics motion or any action. A digital model represents the required features and characteristics, such as dynamics, kinematics etc., of the robot. Simulation software then illustrates the transformation of the robot over time based on the provided input to the model and the available physics. A physics engine is an integral part of the simulation software which allows real world physical behaviour of the digital model, shown in Figure 3.4.

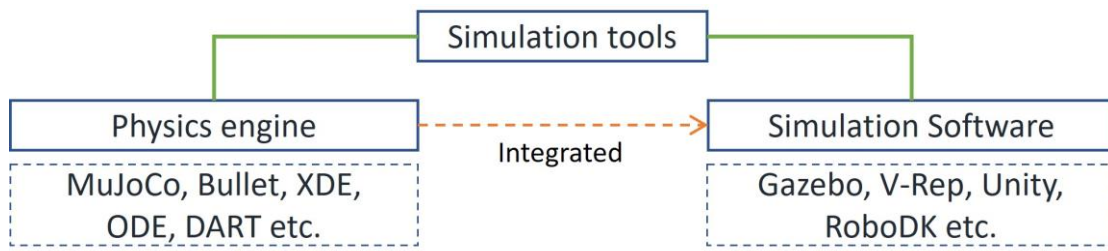


Figure 3.4 Simulation tools for robotics

Simulation software allows developers to test their robot before implementation. This provides financial benefits as well as saves time. Simulation software gives the flexibility in robot development by giving the ability to test the theories and innovation repeatedly. For these reasons simulation softwares are gaining popularity among researchers and developers. There are many simulations software available based on the need. Gazebo [193], RoboDK [194], RobotStudio [195], Webots [196], CoppeliaSim [197], Unity [198] etc. are few examples of widely used simulation software. Because of close integration with ROS and availability of many robotic sensor plugins, Gazebo was selected in this research as the simulation software. Details of the Gazebo simulation software has been discussed in A.3.

3.2.4 Coding Strategy of RIRS

Using the knowledge of system engineering, a coding strategy has been developed for RIRS by combining the generic V-model and V&V model for software development. System engineering is important to understand both the business needs or customer requirements and technical needs [199]. This coding strategy will ensure the successful creation of codes for each objective of this thesis. The coding strategy is shown in Figure 3.5. Moreover, details of the generic system engineering approach and V&V model has been discussed in Appendix B.

There are 4 main steps to create a code for RIRS as followings:

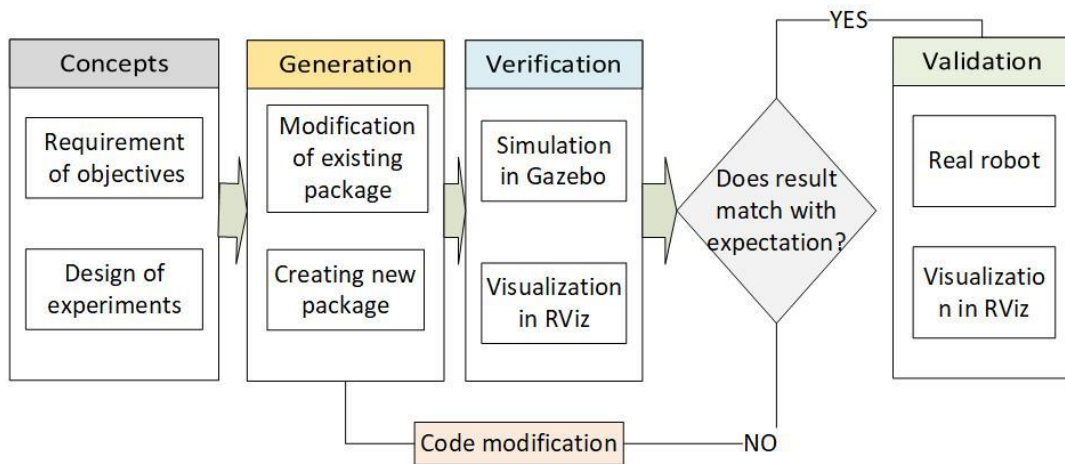


Figure 3.5 Coding strategy of RIRS using system engineering approach

1. Concepts stage: In this stage requirements of the objectives are analyzed to understand the conceptual model of the code. Then all the required parameters for input and output are find out to study the design of experiments. Afterwards, a conceptual model is generated.
2. Generation stage: Based on the conceptual model, a code or ROS package is created in this stage. If any available open-source software package is available to use, then the open-source package is matched with requirements. Based on the requirements of the conceptual model, modifications are carried out in this stage. In case of unavailability of readily available package, a new software package is created.
3. Verification stage: The generated or modified code is then entered in verification stage. For verification, a simulation model in Gazebo is used. This provides flexibility to capture any fault in early stage. Both Gazebo and Rviz can be connected with ROS, which provides an easy visualization of the output.
4. Validation stage: The final stage is the validation. Multiple assessments have been carried out in this stage. This confirms the performance of the code. If the simulation results meet the expected results, then this code is tested in the real robot for validation. In case the code fails to meet the

requirements, they have been sent back to generation stage for modification.

3.3 Kinematics, repeatability, and precision

In robotics, kinematics is the investigation of the links and their relation. It exploits the relation between a robot's joint coordinates and its spatial. Kinematics generates extremely precise and helpful calculations in various situations, such as placing a gripper in a specific location in space, creating a mechanism that can move a tool from one point to another, or determining if a robot will collide with obstacles during motion. The movement of the robot under forces and torques is not considered by kinematics. Kinematics only considers the immediate values of its joint coordinates in each instance.

Kinematics is the fundamental knowledge for various aspects of robotics. As kinematics defines the relations between different links and joints, it is essential to design and analyse the robot. Combined with control algorithms and perception, kinematics can help to plan and control the robot's movement in different configurations. Though kinematics does not consider the forces and torques applied to the robot, it is crucial when calculating the impact of forces and torques. Moreover, knowledge of kinematics is required to optimize path planning, ensuring the robot's efficiency.

3.3.1 Kinematics

3.3.1.1 Configuration space and workspace

Robot kinematics calls for a robot's structure to be defined as a collection of links, most of which are rigid bodies and joints that connect them and limit their relative motion, including rotational or translational joints. The configuration of the robot determines the set of joint coordinates. All fixed-base systems, whether serial or branching, have this property. Configuration space is the space for all the configurations of the robot. Since it is a non-redundant and simplified depiction of the robot's arrangement, configuration space is essential. Workspace is a spatial representation of its connections in the 2D or 3D environment in which it functions. Spatial representations, such as matrices, describe the frame of each

connection to a global coordinate system. In simple words, the workspace is the reachable area of the robot.

3.3.1.2 Robot structures

Various robot systems can be created based on the joints and link types. Robot joints refer to the moveable elements of the robot that result in relative movements between adjacent linkages. Links are the stiff components connecting the joints to ensure proper and precise functioning in this context. Sometimes the robotic joints are referred to as axes. This is because they are necessary and valuable to ensure that the robot's action is unconstrained and unobstructed.

Types of robotic joints:

- Linear joints: these are also known as prismatic joints; input and output links are related linearly and can only provide translation motion,
- Rotational joints: links can rotate on an axis but only on 1-axis, and based on the link configurations, rotational joints will have different joint limits,
- Revolving joints also provide rotational movement, but it does not limit the rotation in 1-axis.

Degrees of freedom are the next fundamental concept to understand the robot structure. Degrees of Freedom (DoFs) define the robot's mobility by calculating the total number of joints and analysing the relations between the links and joints. The offset between the 2 connected links from their laid-out position in a particular reference frame indicates the degrees of freedom for a single joint. The 1-DoF is a joint angle that describes the offset from a joint's zero position along its axis of rotation for revolute joints.

In robotics, joints may have some limitations because of design, physical structure, or collision with other links. These limitations are called joint limits. Reachable workspace, an area of environment that is reachable by the robot in any configuration, depends on joint limits. Both reachable workspace and joint limits are significant for a manipulator. For example, in a mobile manipulator, both

the wheel and manipulator will have a rotational joint. But the difference between the wheel and manipulator joint is the configuration of joint limits.

3.3.1.3 Kinematics of RIRS

RIRS is a mobile manipulator that combines 3 main hardware systems: railway track trolley, warthog UGV, and UR10e manipulator. On-track and off-track navigation are 2 different operational modes of RIRS. There is a total of twenty-four joints in RIRS, as shown in Table 3.3. The number of active joints will change based on the navigation mode.

Table 3.3 Summary of joints in RIRS

	Track trolley	Warthog UGV	UR10e manipulator
Number of joints	12	6	06
Types of joints	Rotation	rotation	rotation
Joint limits	no	no	Yes. Joint 1: 0° to 360° Joint 2: 0° to 360° Joint 3: 0° to 360° Joint 4: 0° to 360° Joint 5: 0° to 360° Joint 6: -360° to 360°

3.3.2 Repeatability and precision

Repeatability and precision are important for a robotic system to measure performance and effectiveness. For positioning measurement, repeatability means the consistency of results when repeated under same conditions, while precision refers to the consistency and accuracy of the results of multiple repetitive motions considering the variations in conditions. For a mobile manipulator, repeatability and precision can be measured from the pose (position and orientation) of the end-effector in a workspace, as in Figure 3.6 [200].

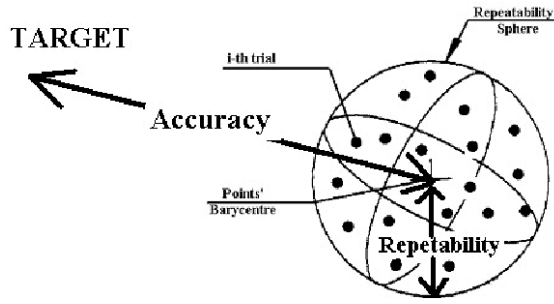


Figure 3.6 workspace of a manipulator [204]

For RIRS performance measurement in this research, the inspection repeatability and precision are measured with the detected defect position referencing the initial RIRS position. The inspection precision can be evaluated by comparing the detected defect position multiple times for inspection execution. Finally, the repair precision is measured by the arm reaching the same end-effector pose multiple times.

The detected defect position from the initial RIRS position is shown in Figure 3.7. The detected defect position from the initial RIRS position can be found by following equation:

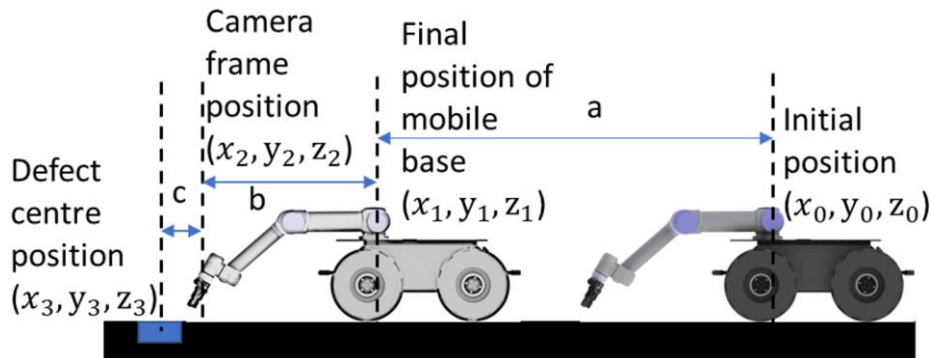


Figure 3.7 Defect position calculation for RIRS

$$d = a + b + c \quad (3-1)$$

Where a, b, c, d denotes the position information in the coordinate system (x, y, z) by the following:

d: defect distance from initial RIRS position

a: distance from initial RIRS position to final RIRS position

b: distance from manipulator base to manipulator tip

c: distance from camera frame to defect

The defect position detected by RIRS depends on 3 measurements: mobile base (Warthog and trolley) movement from the initial RIRS position to the final position, end-effector position from mobile base to camera frame, and defect position from camera to the defect center.

Distance traveled by the RIRS, *a*, can be found from the wheel encoder, from the LiDAR, or using any distance measuring technique. Forward kinematics can be used for finding the end-effector position from the RIRS base, *b* [201], [202]. The end-effector position can be calculated using known joint angles and length values. UR10e manipulator has 6 links, and the final transformation of the end-effector can be found by multiplying the transformation of each link.

$${}^0T_6 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \cdot {}^5T_6 \quad (3-2)$$

Inverse kinematics and D-H parameters are used to find the relationship between the links of a manipulator. Figure 3.8 shows the convention of D-H parameters, while Figure 3.9 shows the frames and joints of a UR10e.

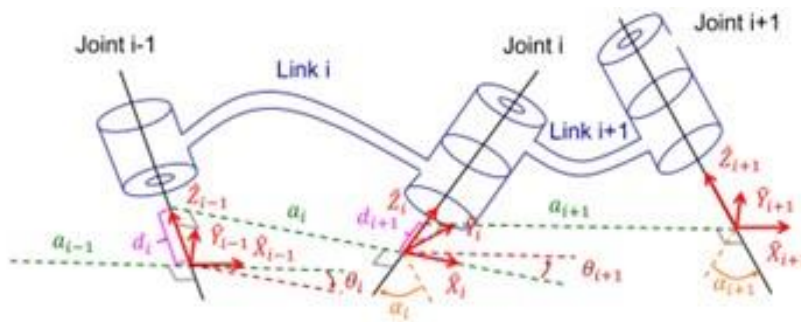


Figure 3.8 D-H parameters of manipulator links

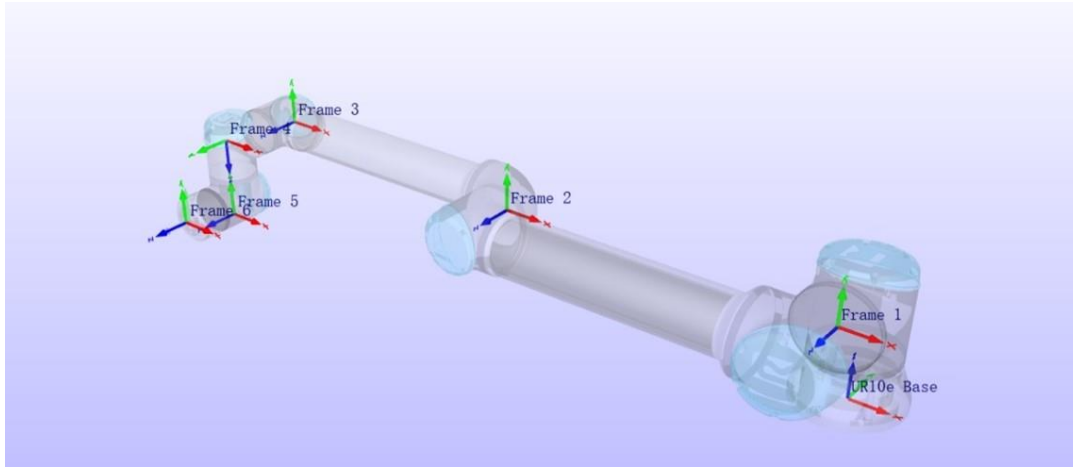


Figure 3.9 Frames and joints of UR10e

The explanation of the 4 parameters is as follows:

θ_i : From the direction of Z_{i-1} , the angle between X_{i-1} and X_i .

d_i : Along the direction of Z_{i-1} , the distance between X_{i-1} and X_i .

a_i : Along the direction of X_i , the distance between Z_{i-1} and Z_i .

α_i : From the direction of X_i , the angle between Z_{i-1} and Z_i .

Table 3.4 Values of D-H parameters for UR10e

Joint i	θ_i [rad]	a_i [m]	d_i [m]	α_i [rad]
1	$\theta_1(0)$	0	0.1807	$\pi/2$
2	$\theta_2(0)$	-0.6127	0	0
3	$\theta_3(0)$	-0.57155	0	0
4	$\theta_4(0)$	0	0.17415	$\pi/2$
5	$\theta_5(0)$	0	0.11985	$-\pi/2$
6	$\theta_6(0)$	0	0.11655	0

The transformation between 2 consecutive links can be found using the following Eq. (3), where one frame can be transformed into another frame using rotation and translation (c denotes $\cos$ and s denotes $\sin$):

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & s\alpha_{i-1} & s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-3)$$

Fortunately, Robot Operating Software (ROS) provides a package named tf for calculating the transformation between the frames of a robot [203]. The advantage of tf is that it keeps tracking dynamic changes between multiple frames. Moreover, ROS extends the transformation frame not only in space but also in time using the following equation of tf.

$$T_{a@t_0}^{c@t_1} = T_{a@t_0}^{b@t_0} * T_{b@t_1}^{b@t_0} * T_{b@t_1}^{c@t_1} \quad (3-4)$$

where $T_{a@t_0}^{c@t_1}$ denotes transformation between frame c to a in 2 different time t_0 and t_1 . This transformation is achieved by 3 transformations: the first one is calculating transformation from frame b to a in time t_0 , then transformation of frame b in 2 different timestamps and finally, finding the transformation of frame c concerning frame b in time t_1 . The transformation between frames using ROS tf is shown in Figure 3.10. Lastly, the position of the defect from the camera frame can be found by using Visual Servoing Platform (ViSP). ViSP auto tracker is an open-source real-time 3D model-based tracker built on the principle of ViSP. For finding the transformation of a feature point in the object to the camera frame, the ViSP auto tracker uses the points transfer method [204]. If a point is denoted in frame 1 as 1P , then the coordinates of the same point in frame 2 can be found using the following equation:

$${}^1p = {}^2tr({}^1p) = \alpha_1^2 H^1 p \quad (3-5)$$

Here, 2_1H is the homography which defines transformation between 2 frames at a scale factor α .

The inspection repeatability and precision of the RIRS system are measured by the whole distance from the defect center position (x_3, y_3, z_3) to the initial RIRS position (x_0, y_0, z_0) . The consistency of multiple repetitions measures the inspection precision.

For the repair performance, the precision is measured by multiple repetitions of the same repair pose motions, executed with and without carrying different payloads.

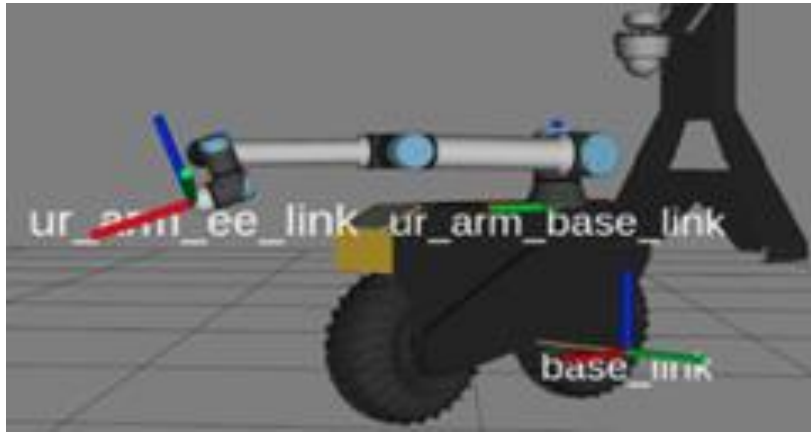


Figure 3.10 Transformation between 2 coordinates frame in ROS

3.4 Inspection (Vision based sensor fusion)

3.4.1 Workflow

RIRS has been designed to perform rail inspection and repair tasks, incorporating many sensors, including vision-based monocular and RGB-D cameras. Taking advantage of the monocular and RGB-D cameras, it is a promising application to autonomously achieve inspection and site-condition based modelling for rail components or foreign objects, supporting further repair manipulating (grind, cut, or remove). In this technology, the RIRS system employs 2 types of cameras and a robot manipulator for proof-of-concept visual inspection and modelling: 1 RGB-D camera fixed in front of the warthog and 1 monocular camera attached at the end of the manipulator.

By fully utilising the sensing and payload system, the methodology of the proposed technique is described in the following Figure 3.11. The technique serves at the maintenance implementation stage after the RIRS has been navigated at the job location or found a new job. Then the RGB-D camera can inspect for maintenance targets on both sides of rails on a big but sparse scale.

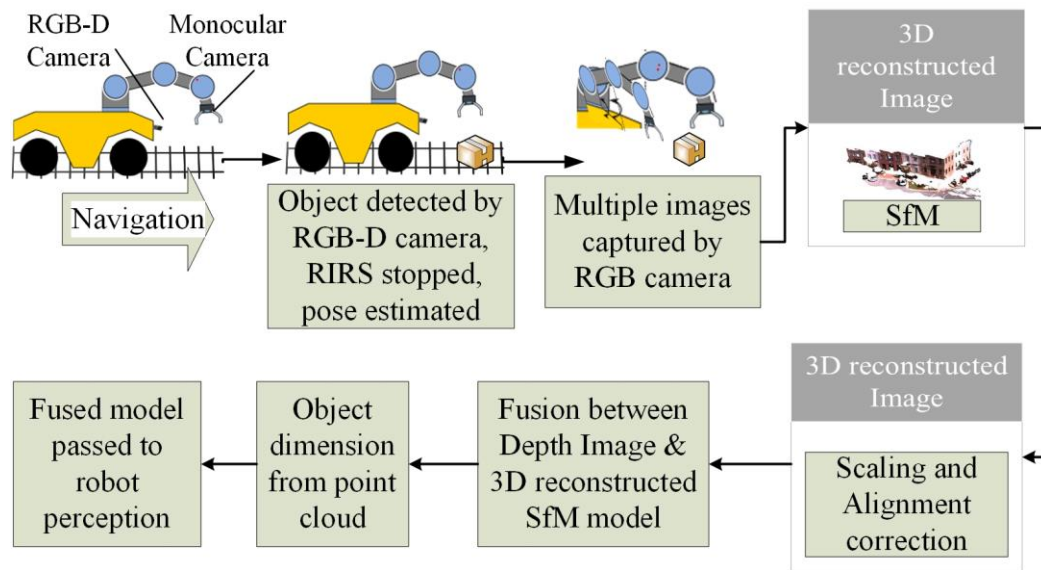


Figure 3.11 Workflow of vision-based sensor fusion for inspection

As the RGB-D camera can generate depth information along with the RGB images, it is possible to get accurate object pose information by associating a 3D coordinate system. ROS tf package can provide pose information of the object from any robot link. Afterward, extracting the relative position from the defect object to the RIRS base, it guides the manipulator for 3D modelling of the target using the 3D scene reconstruction technique. With the help of the arm tip monocular camera, the 3D model with more information (dimension, volume, exterior texture) of the target is acquired. Though a 3D scene reconstruction technique, such as SfM, can generate a dense 3D model of the object, it lacks scaling and alignment information. Fusing the 3D model from SfM with the 3D model from the RGB-D camera can solve the scaling and alignment problem. Finally, the fused 3D model can be transferred back to robot perception.

3.4.2 Structural model generation (SfM)

A 3D structural reconstruction is a tool in computer vision that recovers the shape and appearance of an object [205]–[207]. It can be done in 2 methods: the active method and the passive method. In the active method, sensors such as laser, ultrasound, or microwave interact with the object actively and create the 3D profile. In contrast, the passive method is based on images from a camera.

Structure from Motion (SfM) is a passive, cost-effective, and popular method for 3D reconstruction which uses a single camera to calculate the photogrammetric properties of objects from the movement of the camera [208]–[211]. The workflow of SfM can be divided into 2 main steps: correspondence search and reconstruction, shown in Figure 3.12.

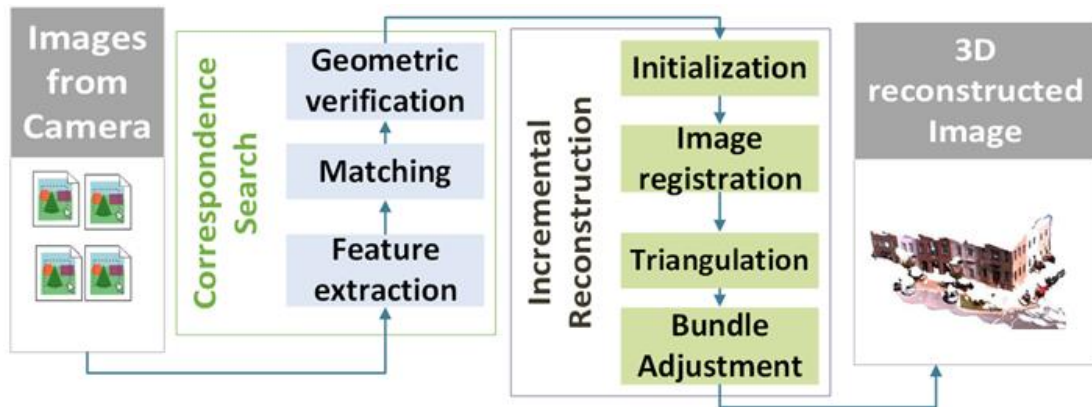


Figure 3.12 Workflow of Structure from Motion (SfM) technique

There are 4 coordinate systems during the SfM process:

1. the coordinate system for pixels in an image
2. the coordinate system for imaging plane
3. the coordinate system for the camera and
4. reference or world coordinate system.

The first 2 coordinate systems, the pixel coordinate system and the imaging plane coordinate system, have the same plane. The difference between these 2 coordinate systems is the principal point and measuring unit. The principal point is located at the upper left corner of an image. Axes are denoted as u in the right direction and v in the downward direction. While in the imaging plane coordinate system, principal point is situated at the centre of the image, and axes are denoted as x in the right direction and y in the down direction shown in Figure 3.13.

The camera coordinate system is located at the camera's centre and parallel to the image plane coordinate system. But the direction of the x -axis of the camera

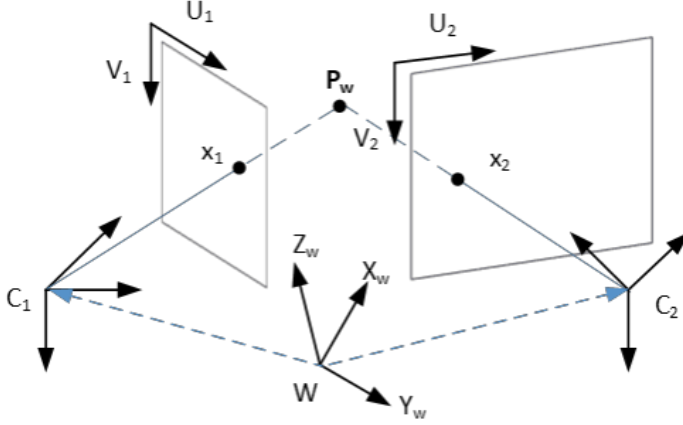


Figure 3.13 Projection of a point in different coordinates frame for SfM

coordinate system is similar to the image coordinate system. In contrast, the direction of the y-axis of the camera coordinate system is opposite to the image coordinate system. Once all the relationships are determined, it is possible to find the conversion between the pixel coordinate system and the world coordinate system using the following equation:

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{d_x} & 0 & u_0 \\ 0 & \frac{1}{d_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (3-6)$$

For solving the 3D reconstruction problem using SfM, the relationship of a point in 2 different images is also required, as shown in Figure 3.13. This relationship can be obtained using epipolar constraint and essential matrix. P is a point in world coordinate system with coordinate value X . Camera captures the image of p as x_1 and x_2 in first and second image, respectively. The optical centre of the first image is C_1 and second image is C_2 . Principle point in imaging plane coordinate system is $O_1(u_1, v_2)$ and $O_2(u_2, v_2)$ respectively. S_1 and S_2 is the vertical distance between point p and image plane. Transformation for the first image plane and second image plane to the world coordinate system can be found by the rotation and translation matrixes $[R_1, T_1], [R_2, T_2]$.

Once all the unknown parameters for transformation are known, SfM can create a relationship between the matching points in 2 images using a different coordinate system and create a point cloud. If there are enough matching points, then camera pose for the images can be found using the perspective-n-point(PnP) [212] or direct linear transformation (DLT) algorithm [213]. Once new image is registered and enough points are matched with previous image, new points in the point cloud will be added using triangulation [214] as shown in Figure 3.14.

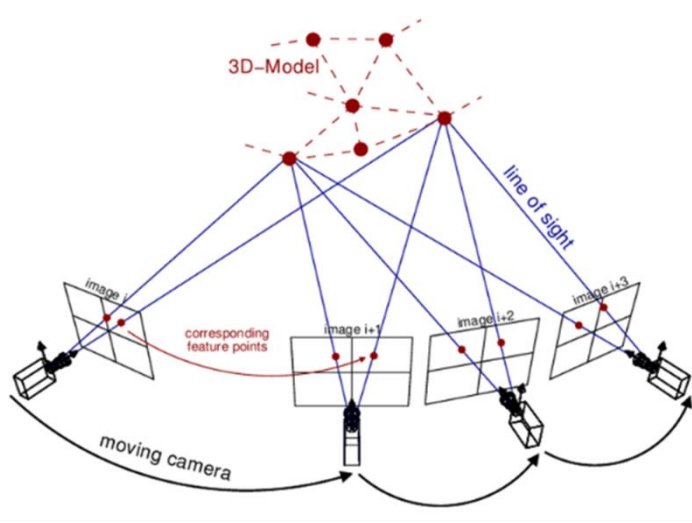


Figure 3.14 Reconstruction of 3D points using triangulation [214]

Uncertainties in the camera poses or error during image registration and triangulation may create reprojection error in point clouds. To reduce the reprojection error, non-linear optimization or Bundle Adjustment (BA) based on the Levenberg-Marquardt algorithm [215] is carried out as the last step for 3D construction using SfM.

$$E = \sum_j \rho_j(\| \pi(P_c, X_k) - x_j \|_2^2) \quad (3-7)$$

Where E is the reprojection error, P_c is the camera parameter, X_k is the point parameters and ρ_j is the loss function to remove the outliers.

3.4.3 Model fusion by perception enhancement and registration

After the 3D modelling of the target object, the final step is to fuse it with the site surrounding captured by an RGB-D camera. Fusing both the 3D model possess 2 challenges one is scaling, and another is alignment. To solve both problems, affine transformation can be used to transform the translation, rotation, and scaling between 2 coordinate systems.

An affine transformation matrix can store translation, rotation, scaling, and shear information in a 4x4 matrix. For example, if there are 2 points or vectors x and y , the affine transformation matrix M , contains the transformation information between 2 points or vector $u(u_x, u_y, u_z)$ and $v(v_x, v_y, v_z)$.

$$u = Mv \quad (3-8)$$

Where:

$$M = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-9)$$

The Upper left 3 by 3 sub-matrix represents rotation, while the right 3 by 1 matrix represents the translation between 2 points. In case there is only translation (points will only move along one or multiple axes), then the translation matrix can be simplified with a translation vector, $t(t_x, t_y, t_z)$.

$$u = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ 1 \end{bmatrix} = \begin{bmatrix} v_x + t_x \\ v_y + t_y \\ v_z + t_z \\ 1 \end{bmatrix} \quad (3-10)$$

In the case of rotation, 3 separate 3 by 3 matrix represents 3 rotation matrices (R_x, R_y, R_z) along 3 axes.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-11)$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-12)$$

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-13)$$

Scaling transformation changes the size of the 3D model either by expanding or contracting by a scaling vector, $s(s_x, s_y, s_z)$. So, if there is only a scaling transformation between 2 points or vectors, the following equation represents the transformation between 2 points or vectors.

$$u = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ 1 \end{bmatrix} = \begin{bmatrix} s_x v_x \\ s_y v_y \\ s_z v_z \\ 1 \end{bmatrix} \quad (3-14)$$

In this experiment, scaling information can be found by measuring the dimensions of a fixed object from the 3D environment created by the RGB-D camera, as the point cloud from the RGB-D camera preserves the original dimension and alignment. A wooden platform can be used as the reference object for the indoor experiment. On the other hand, sleepers and railway track gauges can be used as reference objects in the outdoor railway environment. A common object, available in both the point cloud, can be considered a reference for aligning 2 point-cloud. Coordinates of 4 points from the reference object will be selected. That information will be inserted in the corresponding 4 points from the unaligned object to align the 3D model created from SfM.

3.5 Repair task

RIRS is equipped with a 6 DoFs manipulator with a payload capacity of 10kg. A framework for 2 different repair tasks has been proposed to test the usability of robots in railway track repair tasks. The railway track is significant for the safety of the passenger and freight trains. Therefore, the repair task of railway tracks needs to be very precise. That's why most of the current repair tasks need human

verification. Therefore, a repair task framework has been proposed with human-in-the-loop (HITL). Furthermore, many railway repair tasks, such as welding and grinding, require following a specific trajectory. Hence, a waypoints-based automatic trajectory generation framework has been demonstrated.

3.5.1 Repair task with human-in-the-loop

Railway repair task requires human intervention to complete the task by ensuring precision. Also, a human operator is involved in verifying the task completion, as a minor fault during the repair task will lead to a catastrophic failure. A repair task framework has been designed and tested for the RIRS, where RIRS carries different payloads to the defect position and drops the appropriate payload based on the defect type, as shown in Figure 3.15.

A repair task in a railway is not always straightforward, like industrial task completion, and requires decision-making based on the defect type, shape, position, and severity. Moreover, railway repair task requires many heavy tools.

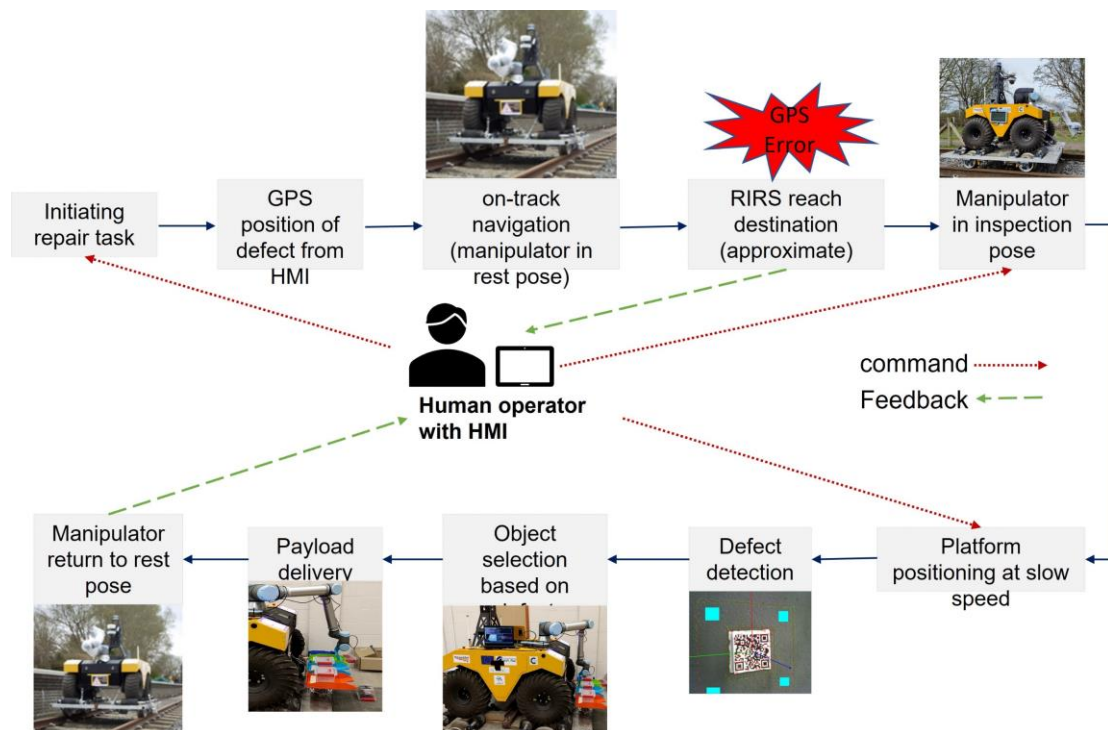


Figure 3.15 Repair task framework with human-in-the-loop

Carrying these heavy tools may create fatigue for the human operator. RIRS can carry all these repair tools, which will ensure an ergonomic repair workspace for a human, reducing fatigue and increasing safety. Furthermore, repair task requires human supervision to ensure the use of correct tools and proper machining work. An experienced human operator can take all these decisions, while the robot struggles to determine the task type, tools, and action.

This task repair starts with a command from the human operator. A human operator can select the job location (latitude and longitude from recoded GPS values for defects). Once RIRS receives the GPS position, it will navigate to the defect position faster, and the manipulator will be set in a safe pose. RIRS will carry multiple payloads for this task. As the GPS output along the railway track is not stable and reliable sometimes, the platform positioning method will be used to reach the defect position. For this, once the RIRS reaches the GPS goal, the human operator will set the manipulator again in the inspection pose with the help of HMI. RIRS will crawl at a slower speed and keep searching for a defect in both forward and backward directions. Once it finds the defects, it will stop. Once the defect type has been identified, RIRS will push the required payloads based on the detected defect type.

3.5.2 Automatic waypoint generation framework

Many repair tasks, such as grinding the rail head, cutting the defected rail section or welding 2 rail sections, will require the manipulator to follow a particular trajectory. Path planning of a manipulator deals with following a specific trajectory. Path planning of a manipulator can be done in 2 ways; in joint space or cartesian space. In joint space path planning, the angles for each joint of the manipulator are well-defined. Then, using the forward kinematics of the manipulator, the pose of the end-effector will be calculated. It is safer and faster as there is no need to solve mathematical equations, and the manipulator's movements are known based on the given angle. But for joint space planning. The Angles of each joint need to be pre-determined, and angles cannot be adjusted automatically.

On the other hand, in cartesian space path planning, the manipulator can find the required angles in each joint based on the known pose of the end-effector. Then, inverse kinematics is used to find the solution of each joint angle. As the workspace of a robot is a 3D cartesian space, it is convenient to use cartesian space planning to generate waypoints.

For example, as shown in Figure 3.16, if the robot detects a defect or object in front of it while navigating, it can perform further inspection by taking multiple pictures of the surrounding area or taking multiple pictures of the object to gather more information about the object. This process starts with detecting the object using the Intel Realsense RGB-D camera. Once the object is detected, the *find_object_2d* package from ROS can extract the position (x, y, z) and orientation (r, p, y) here r, p, y stands for roll, pitch and yaw respectively. Once it gets the object pose, it can use a cylindrical coordinate system to calculate the waypoints. Figure 3.17 shows the framework for automatic trajectory generation for waypoint-based path planning for a manipulator.

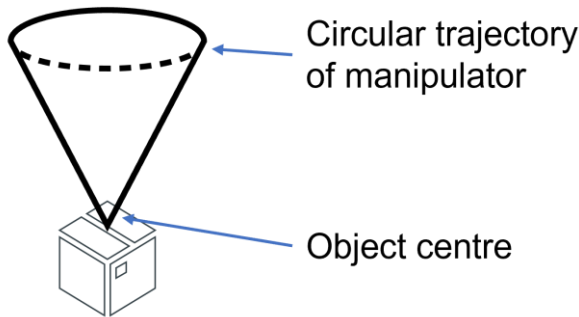


Figure 3.16 Repair task framework for automatic waypoint generations

A cylindrical coordinate system is a simple representation of the 3D coordinate system. It is the simplest way to transform the position of 2D polar coordinates into a 3D coordinate system. In a 2D polar coordinate system, a point in the space is denoted by (r, θ) where r is the distance from the origin and θ is the angle between the axis and the line segment from the origin to the point. These dimensions are calculated in xy plane. If another plane perpendicular to xy plane

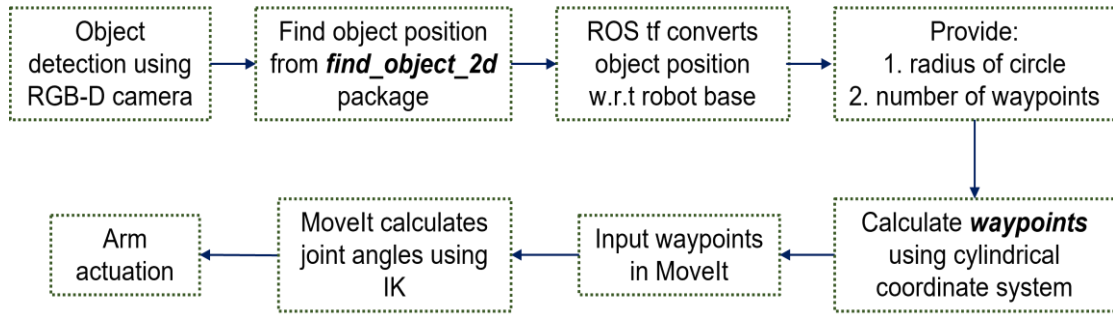


Figure 3.17 Repair task framework for automatic waypoint generations

is added, then it becomes a 3D cylindrical coordinate system. A point in a cylindrical coordinate system can be denoted by (r, θ, z) . Figure 3.18 shows the waypoints for a circular trajectory of a manipulator in a 3D coordinate system.

Points from the cylindrical coordinates system need to convert into a cartesian coordinates system as the manipulator waypoints require the position of a point in a cartesian coordinate system.

$$x = r \cos\theta \quad (3-15)$$

$$y = r \sin\theta$$

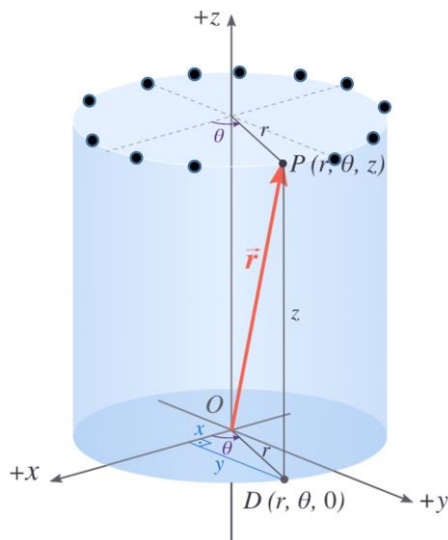


Figure 3.18 Relation between cartesian and cylindrical coordinate system

Here,

θ - is the angle between the axis and the segment of point and base, which can be found by dividing the whole angle of trajectory with the number of waypoints

r – is the radius of the trajectory, which can be defined by the human operator

3.6 Summary

This chapter describes the methodology for this research. It starts with describing the overall research methodology for the thesis and then goes deep into objective-specific methodologies. Initially, an overview of RIRS, description of the software and simulation environment has been given. Then, a coding strategy has been developed based on system engineering approach. Afterwards, kinematic analysis of the RIRS has been done to find the relation among the joints for on-track and off-track operation. Then the way to calculate the repeatability of the RIRS and precision of UR10e has been discussed. In the next sub-chapter, the improved inspection technique has been described in 3 stages starting from the overall workflow, Structure-from-Motion, and fusion technique. Finally, the framework for multiple repair tasks has been illustrated to show the capability of RIRS in performing various repair tasks.

4 RESULTS

This chapter will talk about the results obtained from different experiments to obtain the objectives of this research and the interpretation of the results in the context of RIRS. There will be 4 main sub-sections to discuss the system's kinematics, repeatability and precision, inspection, and repair. Each sub-section will start with an experimental setup, except for kinematic analysis. Then there will be reports of data gathered during the research from multiple experiments. Finally, an explanation will be provided for those data.

4.1 Simulation of RIRS

Simulation of RIRS has been accomplished using Gazebo software as this has very good integration with ROS. To replicate the real-world conditions, simulation has been divided in 2 different scenarios: on-track and off-track. In both the scenarios, the simulation components are divided into 2 different groups. One group of components made the simulation world while the other group of components represents the robots. The difference between world and robot component is motion capability. World components always remain fixed from the reference whereas the robot components move as per the force or motion applied. All the available sensors, such as LiDAR, Camera, IMU were also added in the robot components. The following Table 4.1 shows the components of simulation in different scenarios:

Table 4.1 Components of simulation in different scenarios

	On-track scenario	Off-track scenario
World components	Railway track	Brick Wall Car Tree
Robot components	Trolley Warthog UGV UR10e manipulator	Warthog UGV UR10e manipulator

4.1.1 Simulation environment for on-track

In on-track navigation mode, RIRS navigates along the railway track. After a road-rail conversion, RIRS will sit on the trolley and power will be transmitted through friction and belt and pulley mechanism. In this scenario, the railway track will remain fixed with the reference point as a world component. URDF files and necessary plugins are available in online repository for warthog and UR10e manipulator. But there is no online package available for trolley and railway track. So, SolidWorks has been used to design the trolley and railway track. After that, 3D model has been converted to URDF using a SolidWorks plugin [216]. In real robot, power from the warthog wheel to the trolley wheel is transmitted in 2 steps. First the power is transferred to contact roller of the trolley through friction and then the power is transmitted to the trolley wheel through belt and pulley. But in Gazebo there is no option to simulate the belt and pulley mechanism. So, for simplification, motion has been directly applied in the trolley wheel instead of warthog wheel considering. Following Figure 4.1 shows the simulation environment in Gazebo for on-track navigation.

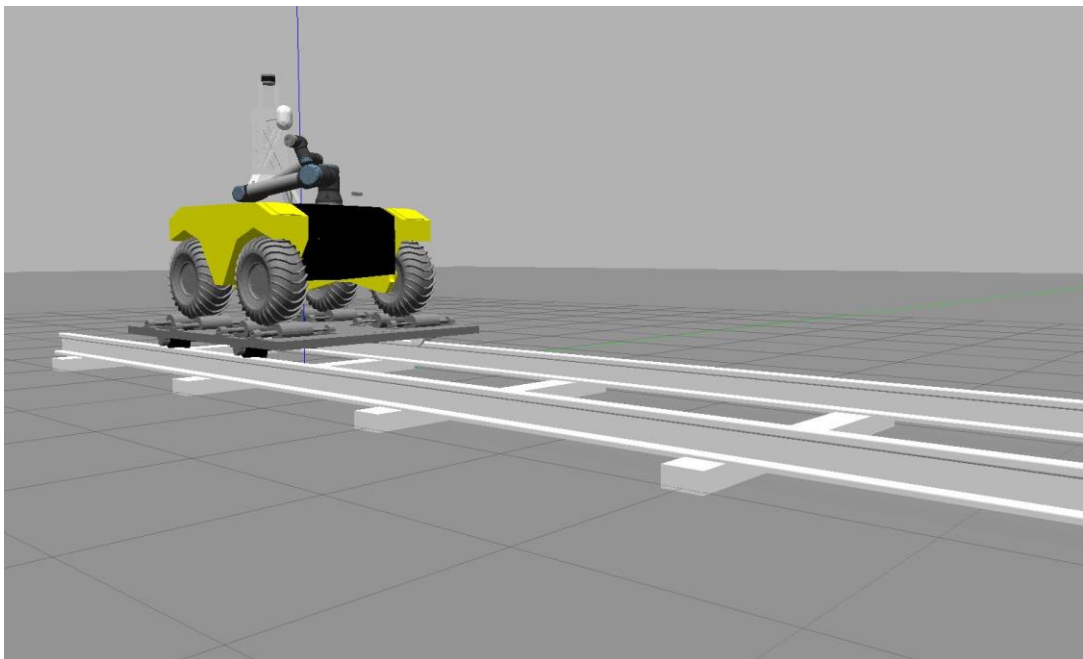


Figure 4.1 Simulation in Gazebo for on-track scenario

4.1.2 Simulation environment for off-track

In off-track navigation, RIRS will run without the track trolley. Though in on-track navigation, the motion of the warthog is only limited to forward and backward, in off-track navigation RIRS can both move in forward or backward direction and rotate in left or right direction. For this, a simulation environment has been created in Gazebo using the available 3D models in Gazebo model library. The world consists of brick walls, a car, and a tree. LiDAR and camera have also been configured with necessary plugin to check the obstacle avoidance and autonomous navigation capability of RIRS. To check the output of the sensors and navigation trajectory or path planning of RIRS, Rviz has also been configured. Following Figure 4.2 shows the environment for off-track navigation of RIRS.

4.2 Kinematics, repeatability, and precision analysis of RIRS

4.2.1 Kinematics of RIRS

Kinematics is an essential and traditional concept in robotics that investigates the connection between different joints and links. Kinematics is independent of force or torque applied in each joint, as kinematics only considers the motion of the

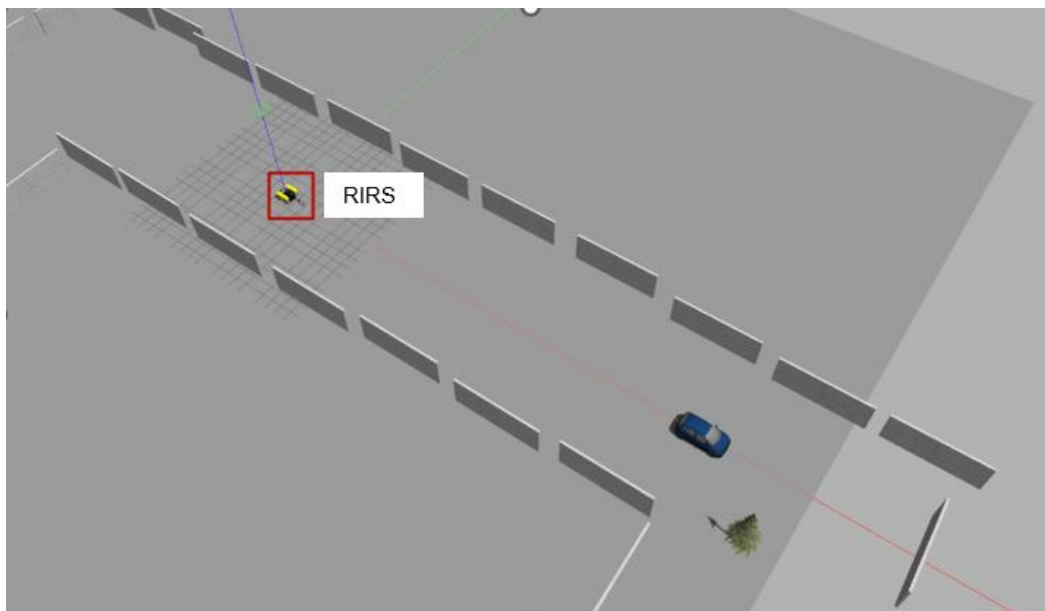


Figure 4.2 Simulation in Gazebo for off-track scenario

robot. Therefore, it benefits robot design, simulation, and control algorithms. Many issues, such as finding the position of the end-effector, avoiding collision with other parts of the robot, predicting robot motion, etc., can be easily calculated using different theories of kinematics. Kinematics does not consider the impact of environments and solely considers the instantaneous conditions of the robot chain.

Understanding the joint type and associated link is necessary to enable the autonomous execution of inspection and repair tasks fully. The arm has 6 revolute joints, and their limit is from 0^0 to 360^0 (J_1 to J_6). Warthog UGV has a total of 6 joints as shown in Figure 4.3(a). Among them, 4 are continuous, and 2 are revolute. 4 continuous joints are situated in 4 wheels (J_9 to J_{12}). 2 wheels on the left side are connected and driven with 1 motor. This is the same for the right-side motors. Both the pairs, “right_diff_unit_link” and “left_diff_unit_link”, have revolute joints to the chassis and have a limit of -45^0 to $+45^0$ (J_7 and J_8). These 2 joints help the UGV to move upward or downward in any terrain. For the trolley (Figure 4.3(b)), 2 contact rollers will contact each UGV wheel, and each pair of contact rollers are considered a single joint for simplification. There is a total of 4 pairs of contact rollers. Finally, 4 10-inch rail wheel has 4 continuous joints.

4.2.1.1 On-track operation

Due to the difference in dynamic motion of the vehicle between the on-track and off-track scenario, the number of joints and performance will vary depending on on-track and off-track operation. When the RIRS is engaged in the on-track operation mode, it has a total of 24 joints. The railway track will guide the RIRS, and all 4 wheels will run at the same speed. There will be no left or right turn initiated from the RIRS. In this scenario, the following simplification can be considered:

- A combination of 1 warthog wheel, 1 pair of contact rollers, and 1 10-inch rail wheel can be considered a single joint. For example, J_9 , J_{13} , and J_{17} act as a single joint. This simplification will reduce the number of joints and eliminate some complexity in programming.

- As the track can be considered as a flat surface, both the revolute joint in the warthog (J_7 and J_8) will be useless. Hence, it can be eliminated.

After simplification, we can consider a prismatic joint on the lower part of the mobile manipulator where the railway track is acting as a guide. Hence, the number of joints has been reduced to 7 from 20. Figure 4.4 shows the joint analysis during on-track operation.

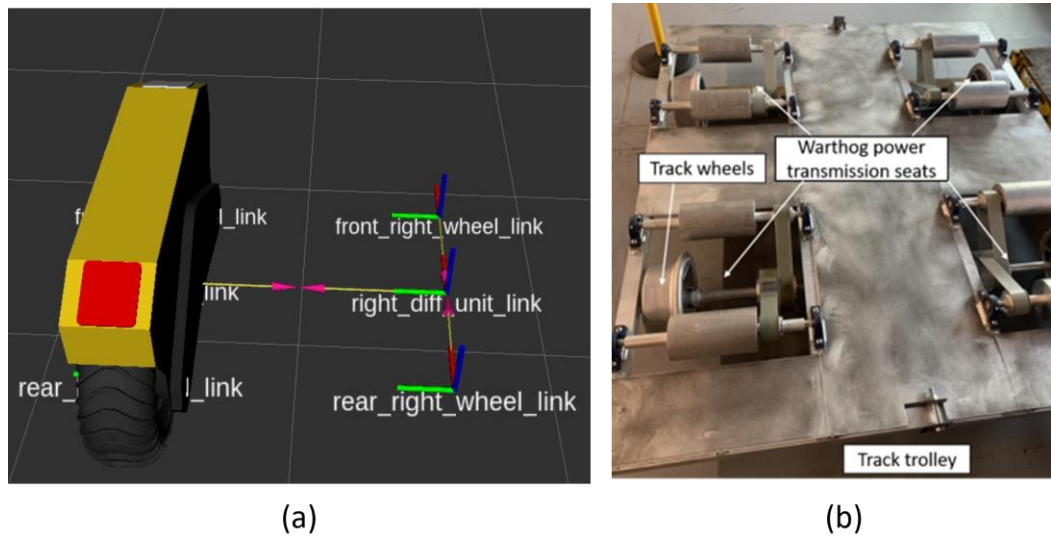


Figure 4.3 Construction of Warthog and specially trolley for RIRS; (a) 6 joints of warthog from cross-section, (b) design and power transmission mechanism of trolley

4.2.1.2 Off-track operation

When the RIRS is navigating off-track, then the trolley is not required anymore. RIRS will run on pneumatic wheels. So, in this scenario, no need to consider the trolley joints. There will be a total of 12 joints: 6 joints of the manipulator and 6 joints of the warthog. Figure 4.5 shows the joints during off-track operation. Among the 6 joints in the warthog, 4 joints are associated with 4 pneumatic wheels. These 4 pneumatic wheels are connected to each other skid steering drive system. On each side, both the front and rear wheels relate to a chain. Rest 2 joints on the warthog help to rotate the wheels freely to overcome obstacles or navigate in rough terrain.

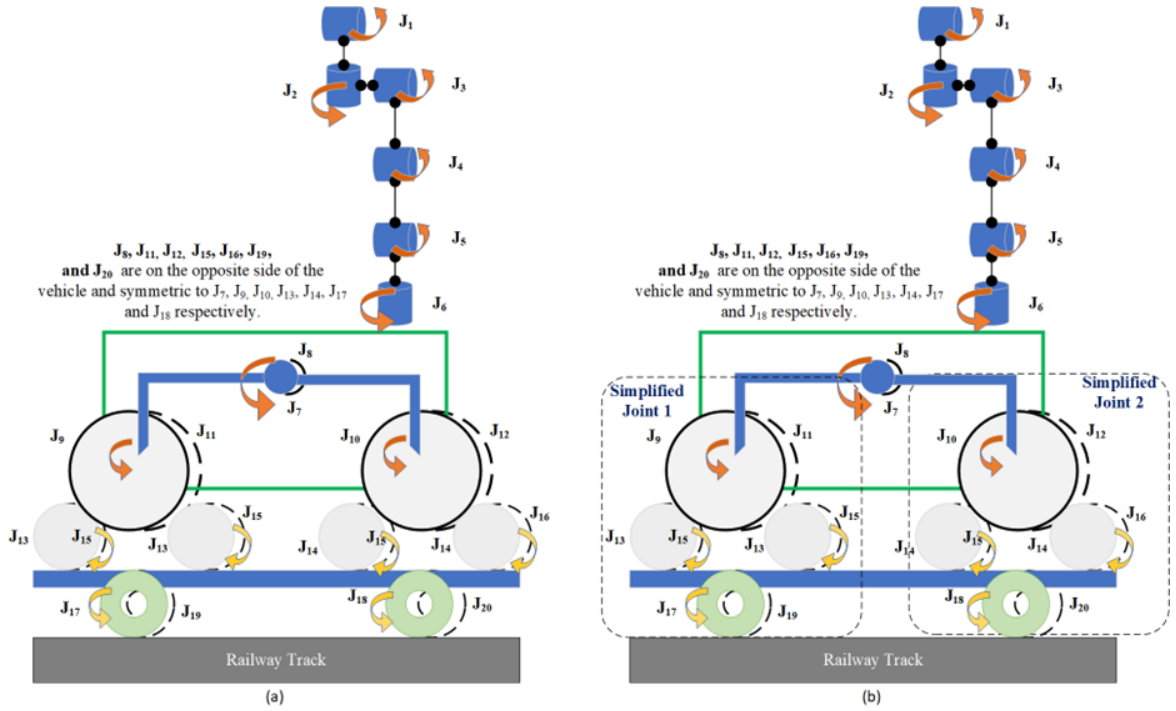


Figure 4.4 Joints of RIRS during on-track navigation; (a) all joints of RIRS, (b) joint configuration of RIRS with 2 simplified joints

4.2.2 Repeatability and precision of RIRS

4.2.2.1 Experiment and data collection procedures

Defects on railway track are varied in size and shapes and successful detection of those defects will require combination of state-of-the-art sensors and computer vision algorithms. As this thesis focuses on the overall inspection workflow, a simulated defect has been used for time constraint and simplification. For inspection, a simulated or dummy defect, a QR code has been attached on top of left rail, as shown in Figure 4.6. An initial RIRS was set in an inspection pose (appendix 6.4.3D.1D.1) so that the monocular camera can see the left rail. The inspection process was started by running a python code and ViSP auto tracker simultaneously. Programmed code moves the RIRS forward with constant speed and detects QRcode position by ViSP auto tracker algorithm. ViSP auto tracker

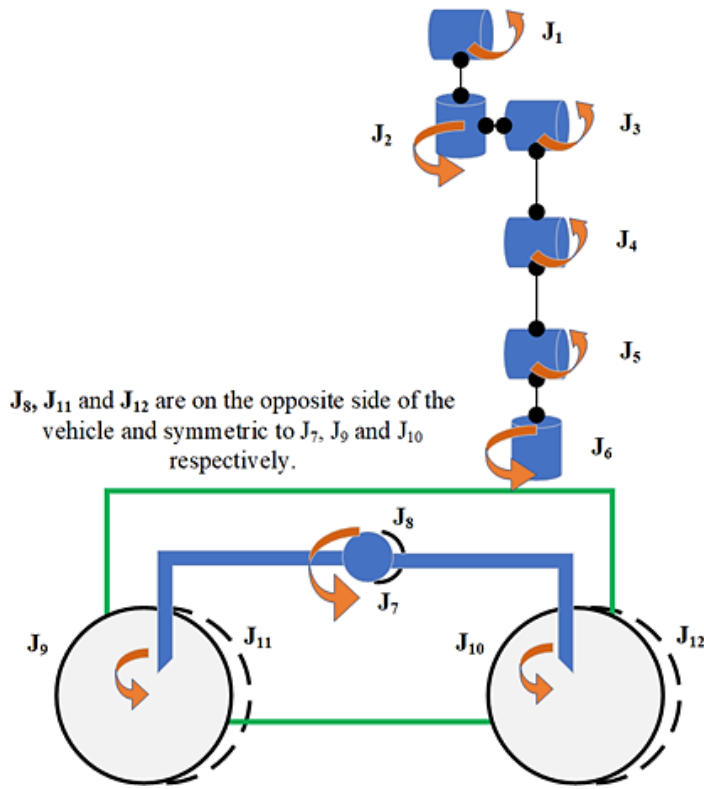


Figure 4.5 Joints of RIRS during off-track navigation

can stably extract and publish the position and orientation, specifically the relative centre position of QRcode to the camera. Once QRcode is detected, the RIRS stops and keep detecting, and it will create a bounding box and display the axis similar to Figure 4.6. RIRS was again moved back to the marked initial position and rerun total of 10 times for the same QRcode position.

For repair task, once the inspection process was finished, 2 different poses with varying angle were defined for the manipulator; one is straight pose, and another is repair pose (appendix D.1). Also, payloads were attached to the manipulator after each set of experiments. Every time manipulator will perform motion from straight pose to repair pose with carrying the payload while UGV was stationary and QRcode position was fixed.

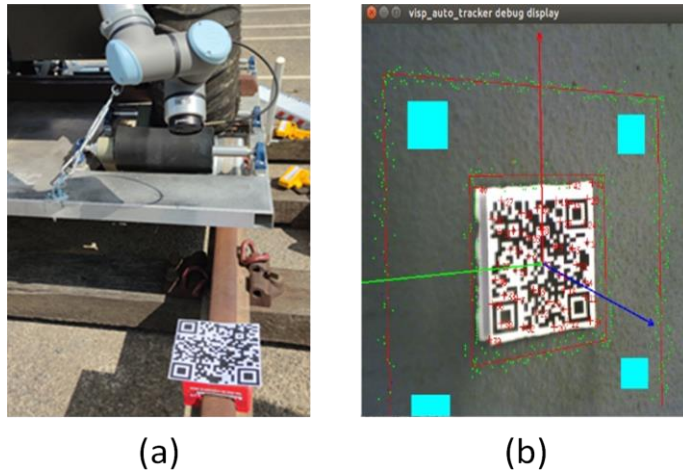


Figure 4.6 QRcode for inspection task, (a) QRcode on the track, (b) output after detection

4.2.2.2 Results

4.2.2.2.1 Repeatability for inspection task

To find the defect position from the initial RIRS position, 3 distances were recorded and summed together. The distance travelled by RIRS was recorded by a laser sensor, distance from manipulator base to manipulator tip was calculated using ROS tf [203] and defect position from camera frame was found from ViSP auto tracker. ROS tf is a standard transformation library for ROS which keeps track of all the joints and links with timeframe and publish the position and rotation in the coordinate frame. Then these 3 distances were computed to find the defect position from initial RIRS position. The real defect position is also captured as a reference by a laser point measure kit. The measured defect position and real defect position were compared to find out the repeatability of system. Figure 4.7 shows the measured result for error in defect position to find the performance of RIRS in inspection.

The actual defect position from the initial RIRS position is 3.75m. The maximum error in finding defect position is 0.056m (56mm) while smallest error is 0.017m (17mm) and the error range (difference between the maximum and minimum

error) is 0.0387m (38.72mm) and the average position error to find the defect position during inspection is 0.89%. Related data are included in Appendix E.1.

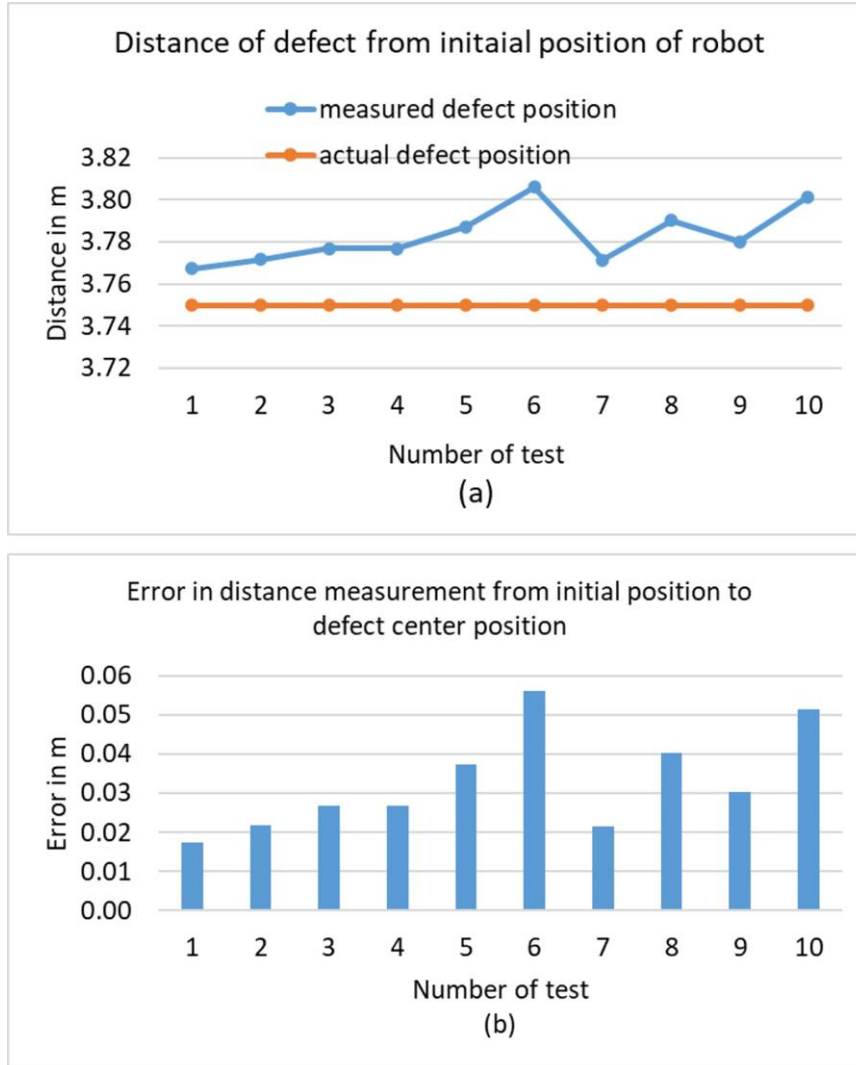


Figure 4.7 Error investigation for inspection job, (a) calculated and actual defect position from initial RIRS position, (b) error in every test

4.2.2.2.2 Precision for repair motion

Maximum and minimum of positions in every axis was calculated and then the range was determined, which is shown in Figure 4.8. The range for deviation was bigger in z axis than the other axis.

In Figure 4.9(a), range for manipulator position without payload in 3D space is [0.63mm, 0.38mm and 2.00mm]. Figure 4.9(b). shows recorded pose of UR10e manipulator after 10 repair motions with 1.75kg payload from initial pose. Few poses deviate more than the mean and deviation in 3D space is [0.21mm, 0.83mm, 2.17mm].

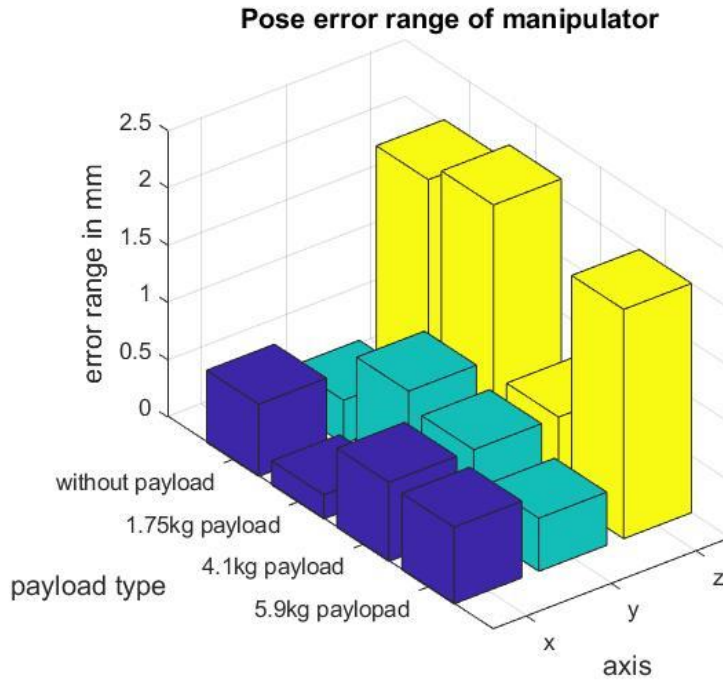


Figure 4.8 Maximum and minimum position of UR10e manipulator in 3D coordinate system

End-effector pose with 4.1kg payload after 10 repair motion in shown in Figure 4.9(c). 3 points are far from mean and error range (difference between maximum and minimum error) increases both in x and y axis compared to no payload condition to 0.69mm and 0.438mm, respectively. Figure 4.9(d) shows the end-effector pose with 5.9kg payload after 10 repair motion from initial pose. Points are much scattered, and the range of deviation is biggest in all 3 axes. The measured precision deviation range is [0.67mm, 0.46mm, 2.00mm] in 3D space.

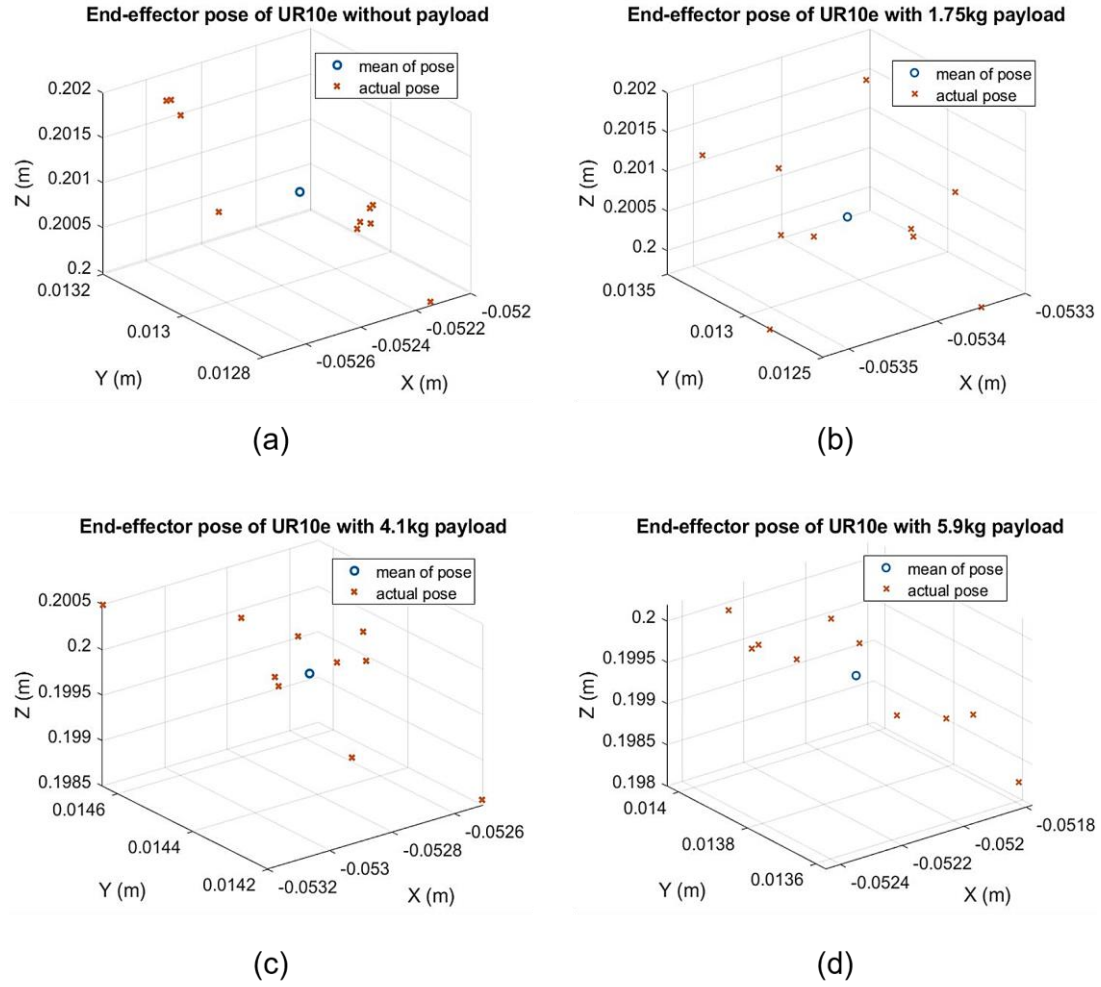


Figure 4.9 Pose error of UR10e from 10 repetitive repair motions (a) without payload, (b) with 1.75kg payload, (c) with 4.1kg payload, (d) with 5.9kg payload

4.3 Improved inspection technique

4.3.1 Experiment and data collection procedures

The experimental system contains 2 major features: the RIRS and the surrounding environment. An open-source Robot Operating System (ROS) was used to program the system. The 3 cameras have different functionality. The (Pan, Tilt, Zoom) PTZ environment camera can be rotated 360° which is useful to control the RIRS remotely using Human Machine Interface (HMI) or from a control station. The Intel Realsense D435i RGB-D camera provides a depth

image of the front environment and can detect an object. The RGB arm camera can be used for close inspection because it is attached to the manipulator so it can be moved.

The experiments have been conducted both in the laboratory, shown in Figure 4.10(a) which mimics an off-track scenario, and also in a simulated railway test environment, shown in Figure 4.10(b) which imitates an on-track scenario.

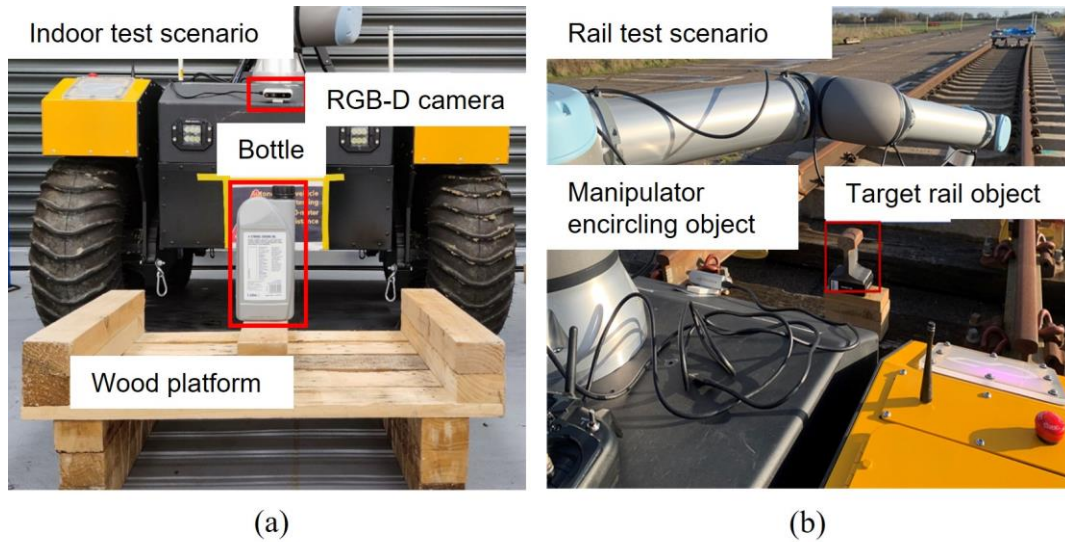


Figure 4.10 Experimental setup (a) inside lab, (b) RITA railway track test scenario

In the laboratory environment, a bottle is used as a target object, while a piece of railway track is used as a target object for the outdoor railway environment. Dimensions of the objects (bottle and rail cut section) and some fixed objects such as wooden platform and sleeper are measured manually using tape measure and the dimensions are shown in Figure 4.11.

Once the Intel Realsense RGB-D camera detects the object, the RIRS will stop. A rosbag has been recorded with depth and tf topic. Later, "pcl_ros" library has been used to convert the bag file to ".pcd" format point cloud for further processing. Then, 20 images of the objects are captured using the RGB camera attached with the manipulator and subsequently processed in VisualSfM to create

3D structural reconstruction. Finally, both the 3D models were scaled, aligned, and oriented using affine transformation.

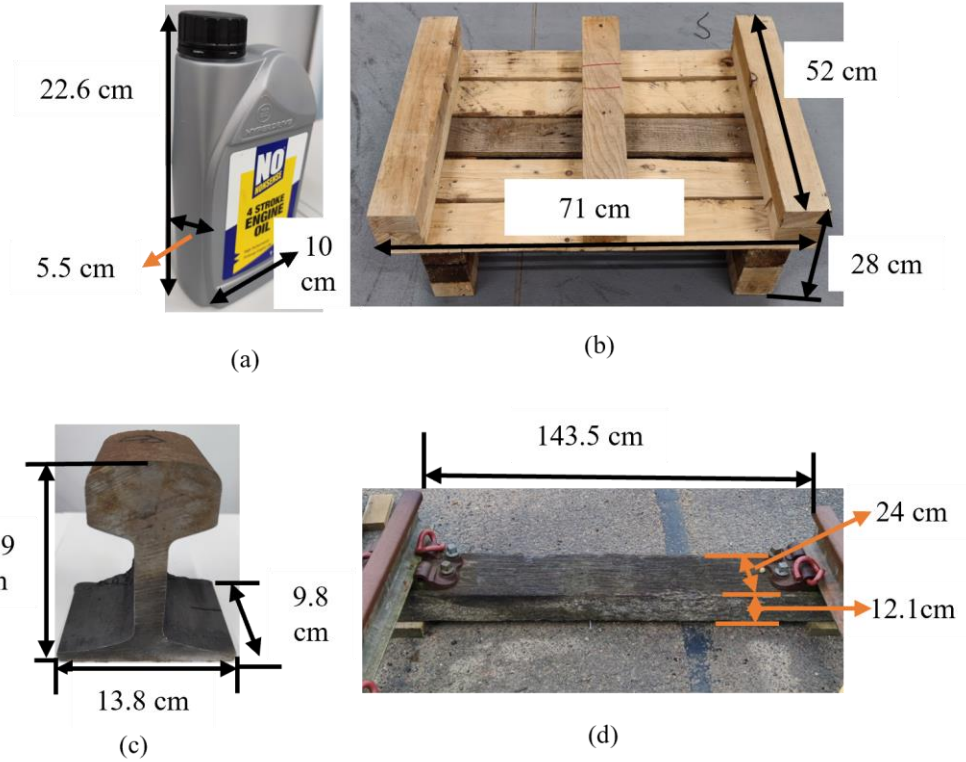


Figure 4.11 Dimensions of objects (a) indoor target object (bottle), (b) wooden pallet, (c) outdoor target object (rail section), (d) sleepers in standard railway track

4.3.2 Results

4.3.2.1 Object detection by depth camera

RGB-D camera is mounted on the bulkhead front mount link of the robot and can only visualize the front of the robot. Figure 4.12 shows the robot's perception of the environment from the RGB-D camera. Figure 4.12(a) and Figure 4.12(b) show the RVIZ visualization for the lab testing scenario (a wood platform, bottle, and nearby environment) and the RITA railway track scenario (rails, sleepers, and rail cut section) respectively.



(a)



(b)

Figure 4.12 Perception of the environment from RGB-D camera in (a) indoor lab

Figure 4.13 shows the multiple side views of the 3D model of the bottle and the rail cut section. The front view from the RGB-D camera shows a clear sensing of the target object, however, other sides are not captured. In this point cloud, only the distance from the object to the RIRS platform can be measured. Meanwhile, due to the fixed angle and sensor limitation, there is a shape distortion happening at the bottle body and the rail foot in the front view in Figure 4.13(a) and Figure 4.13(c). Though the RGB-D camera can create a quite dense model of the bottle at the front side, it failed to capture enough point cloud of the rear area which can be seen from Figure 4.13(b) and Figure 4.13(d).

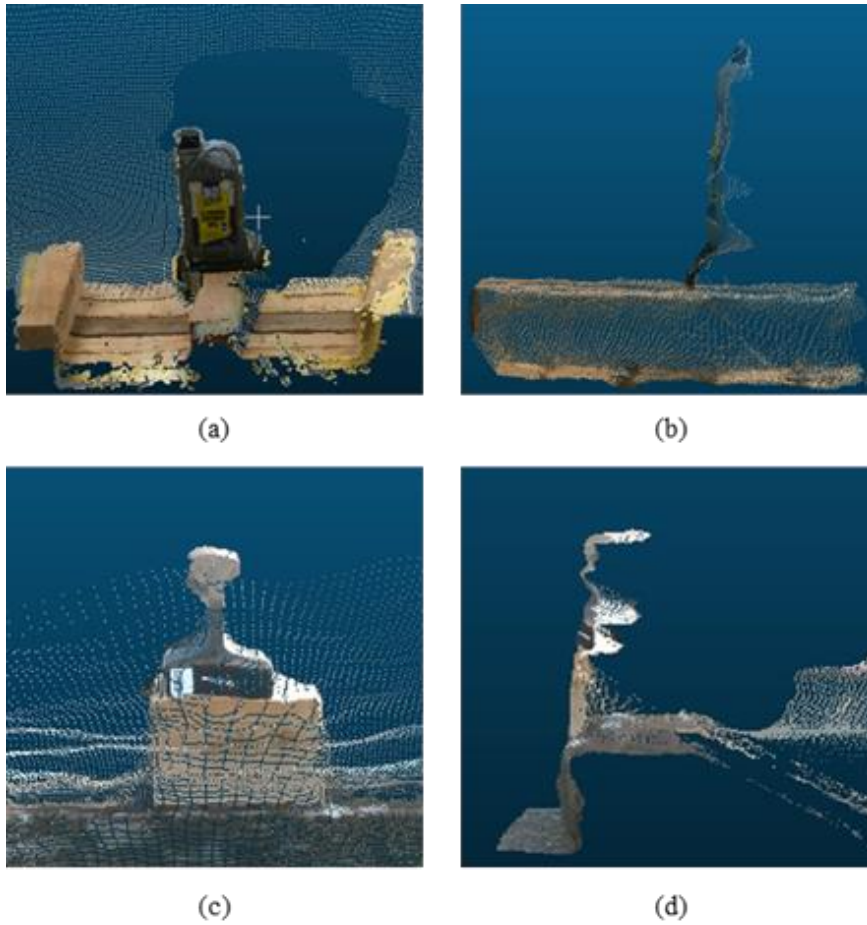


Figure 4.13 Front and side view of 3D model of the bottle (a) front and (b) side view and rail section (a) front and (b) side view

4.3.2.2 3D object modelling

In the next step, structure from motion technique is used to create a 3D model of the object. RGB camera attached to the manipulator tip has been used to capture 20 images from different perspectives where the objects are fully visible. Though more images will provide better results, it will increase the computational burden. For example, as shown in Figure 4.14, reconstruction process from 20 images took 1 minutes while reconstruction from 80 images took almost 5 minutes. Camera position and angles for 20 images were selected manually, details will be provided later in discussion (5.3.1). VisualSfM, an open-source SfM software, was used to create a 3D dense model of the objects for both the indoor lab test and the outdoor railway track test scenarios. A dense 3D reconstructed model of

the bottle and rail cut section is shown in Figure 4.15. Reconstruction of the scene contained many objects other than the target object. So, manual segmentation technique was used to separate the target object from the rest of the model.

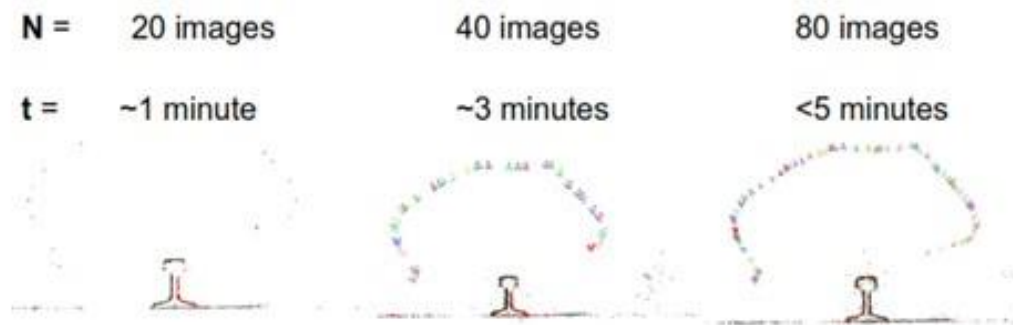


Figure 4.14 Relationship between reconstruction time and number of images

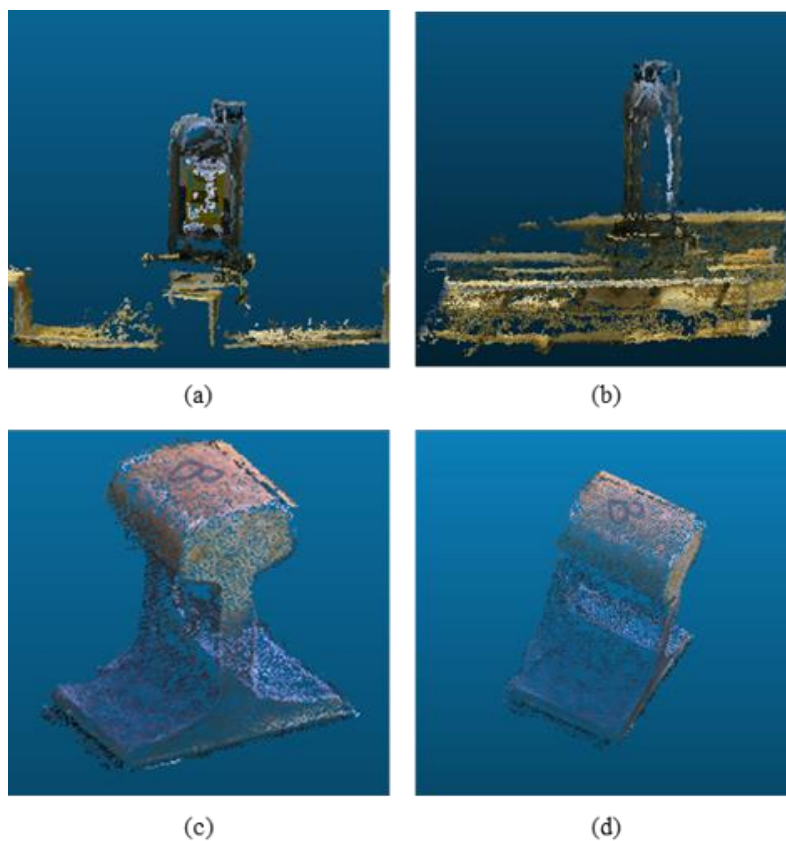


Figure 4.15 Dense 3D model of the bottle (a) front and (b) side view and (c) front and (d) side view generated by SfM technique using RGB camera and manipulator

4.3.2.3 Model fusion for final perception

Finally, both the 3D model generated from the RGB-D camera and SfM are fused in the CloudCompare software. However, the 3D reconstructed models from the SfM do not contain any scaling and alignment information, shown in Figure 4.16. Before fusing both models, it is important to correct the scaling of the SfM model.

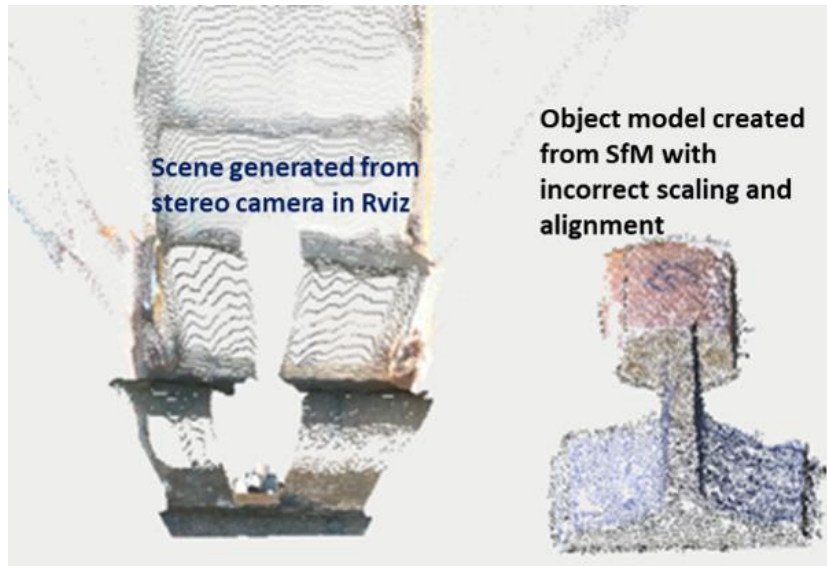


Figure 4.16 Wrong scaling and alignment of 3D reconstructed model generated from SfM

In the railway track test scenario, the dimension of the sleeper between 2 rails are British standard gauge. The distance between 2 rail foot is 1360mm, and the width of the sleeper is 240mm. Using these length and width information, 4 points can be created as a reference coordinate system. Then, these reference points are selected in the 3D reconstructed SfM model to acquire the correct scaling factor. Figure 4.17 shows the coordinates of the 4 reference points created for extracting the scaling factor of the 3D reconstructed model. Once the object model was scaled properly, the next step is to fuse the 3D reconstructed model into the RGB-D camera model. During the alignment correction, the point-cloud from the RGB-D camera is considered as the fixed model as it contains information about alignment with regards to the robot. Rigid transformation requires at least 3 corresponding points from each model. In this experiment, 4 points are selected from each model for the rigid transformation. Figure 4.18(a)

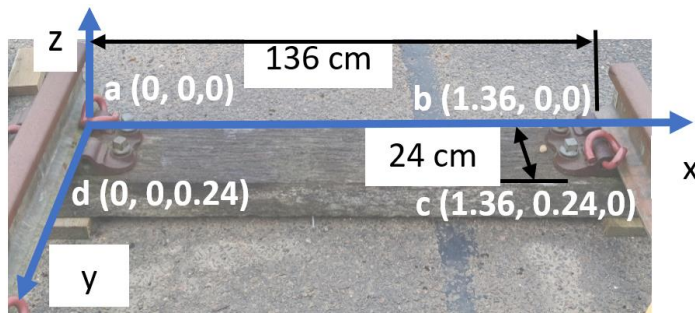


Figure 4.17 Wrong scaling and alignment of 3D reconstructed model generated from SfM

and Figure 4.18(b) shows the front and side view of the rail cut section after fusion into the RGB-D 3D model.

This experiment diminishes one of the shortcomings of robot perception for railway track inspection and repair work. The RIRS can perform more effectively in track maintenance tasks owing to the better 3D model of the defects and objects created from the fusion between the cameras. Moreover, due to the mobile manipulation capability of the RIRS, this technique can be replicated in many other scenarios apart from rail inspection and repair tasks.

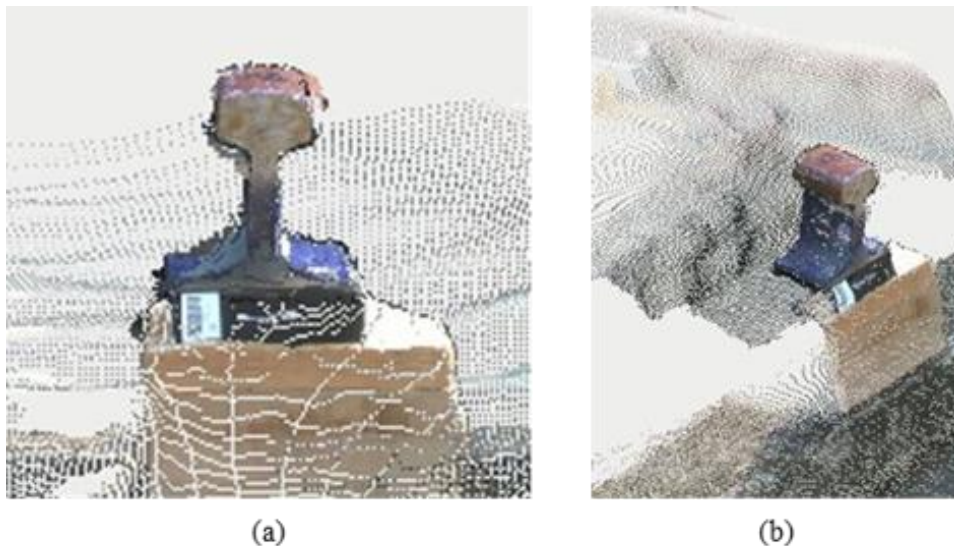


Figure 4.18 Fusion of rail model after aligning and scaling (a) front view (b) side view

4.4 Repair

4.4.1 Human-in-the-loop repair task

4.4.1.1 Experiment and data collection procedures

This repair task includes human involvement in different stages of completion. An HMI has been designed and developed for RIRS using WebSocket and ros_bridge to transfer the operator's command and get real time feedback. The human operator has an HMI connected to the RIRS via a laptop through wi-fi. Graphical User Interface (GUI) of HMI is shown in Figure 4.19. The target for the robot is to deliver the correct repair tools from a selection of various repair tools in the defect area. During different stages of the experiment, the robot will use multiple sensors, such as a monocular camera, RGB-D camera, GPS, LiDAR, etc., to reach the defect location, identify the defect type, interact with the environment, and deliver the payload. Both the robot base and defect position (goal) are defined by latitude and longitude from GPS output. Figure 4.20 shows the base and goal position on a stretch of a rail for experimenting human-in-the-loop (HITL) repair task framework.

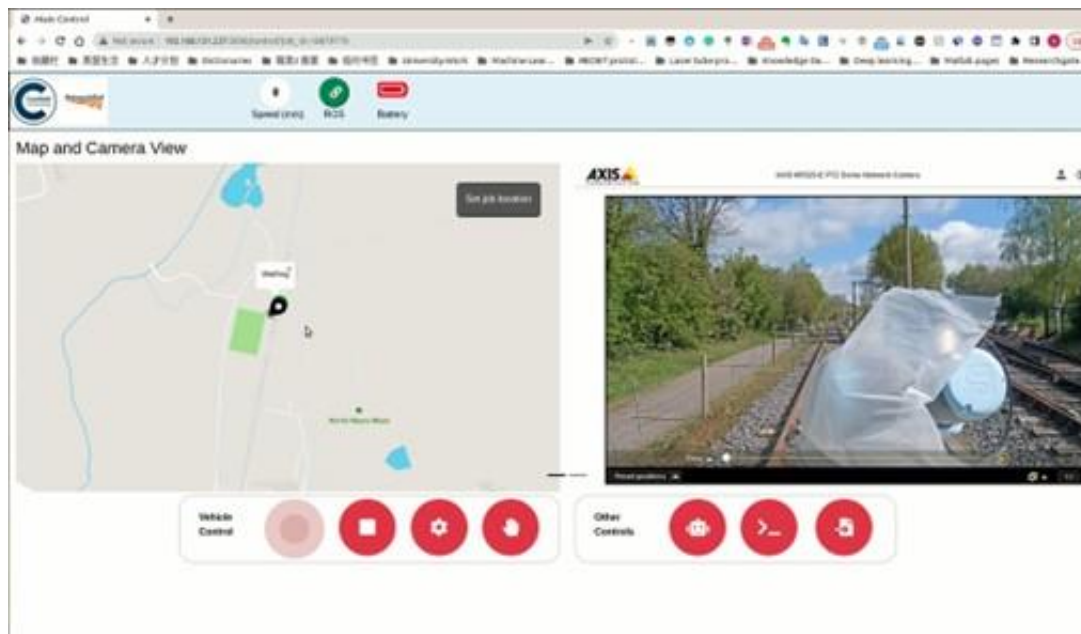


Figure 4.19 GUI of HMI



Figure 4.20 Experiment environment for human-in-the-loop repair task

Initially, a place on the rail was designated as the base position for the RIRS. Both the latitude and longitude of that position have been recorded and updated to the HMI. Then, the previously recorded defect location was defined from GPS and updated on the HMI as well. The whole task will be completed in 3 stages:

1. Navigation from base to goal
2. Delivering the repair tools
3. Returning to base

There are 3 simulated defects, denoted by 3 different QR code containing different message. The RIRS will carry 3 different toolboxes which are mechanical tools, electrical tools and pandrol clips, as shown in Figure 4.21. Total 15 experiments have been conducted for with 3 different simulated defects, 5 times for each defect. Increasing the number of experiments will increase the statistical significance of the result. As the experiments were conducted in Northampton and Lamport Heritage railway instead of laboratory facility. So, the number of experiments were limited to 5 because of time shortage 5 because of time shortage and limited availability of heritage railway facility. Necessary codes are given in the appendices section in D.1 and D.2.

Performance of the RIRS in successfully determined by following criteria:

1. Successful defect detection
2. Successful delivery of toolbox
3. Cycle time

4. Time for payload delivery
5. Latency of the connections

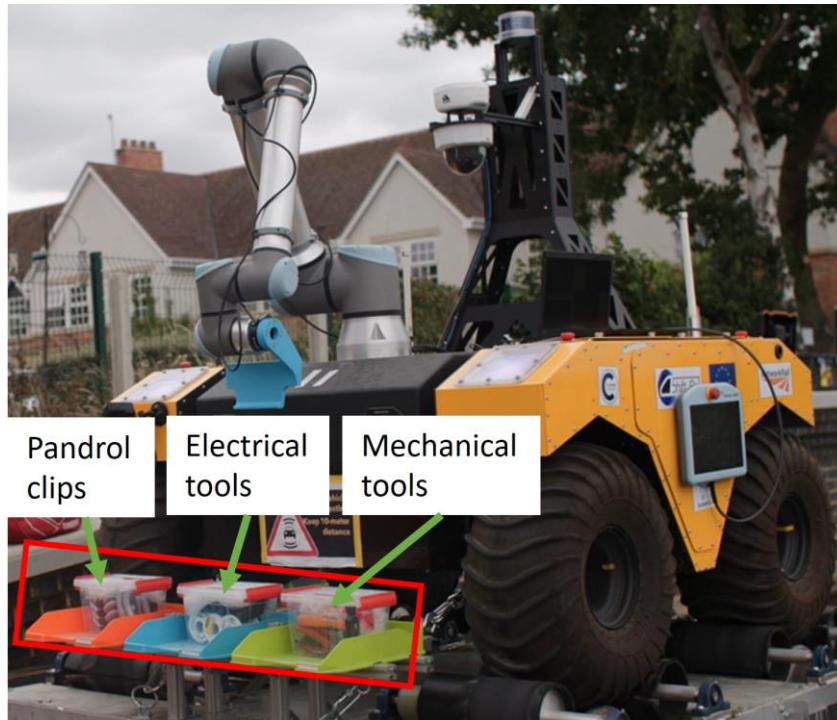


Figure 4.21 RIRS carrying 3 different toolbox

Cycle time is the total time required for completing the whole task. Whole cycle is divided into 3 stages: navigation from base to goal, delivering repair tools and returning to base as shown in equation 4-1. Counting of time starts when the human operator gives command to RIRS to navigate to goal position from HMI and counting ends when the RIRS return to goal position.

$$\text{Cycle time} = \text{Base to goal} + \text{Perform repair task} + \text{Return to base} \quad (4-1)$$

Final performance measures, latency of the connections, are different than the rest of the items as it also includes human interception and confidence. If the latency of the whole framework is too high, it will reduce the confidence of human operator too use the system.

4.4.1.2 Results

Defect detection and toolbox delivery

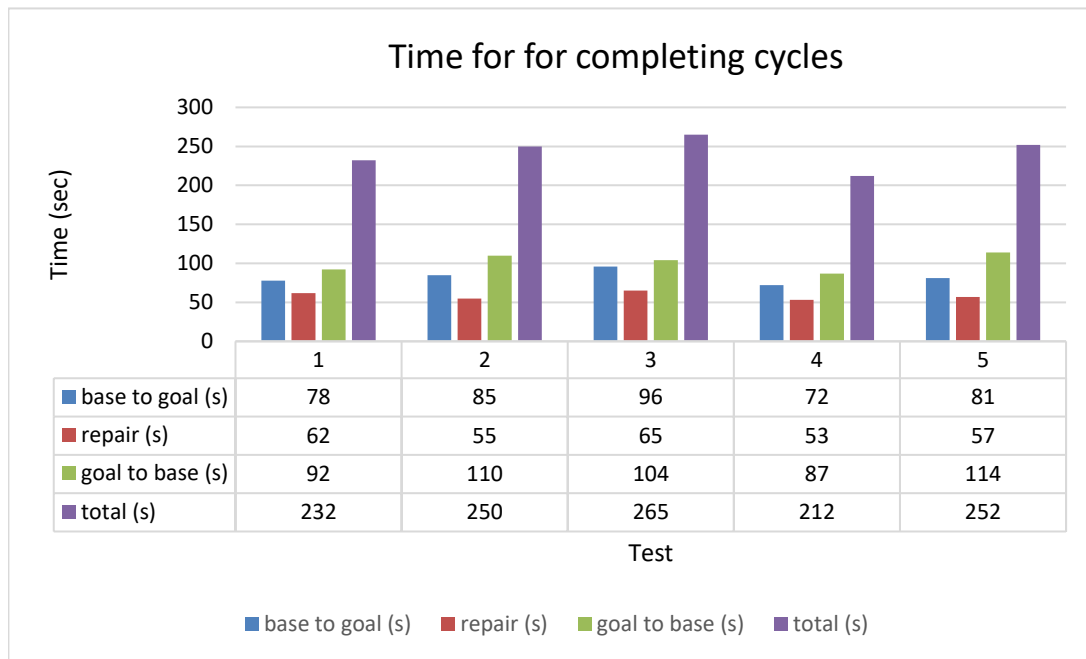


Figure 4.22 Cycle time for human-in-the-loop repair task

5 tests were performed to find the average cycle time for the whole process. Figure 4.22 shows the time in seconds for completing each step and the whole cycle. Lowest recorded cycle time was 212 seconds while the highest was 265 seconds. Moreover, time required for each step were recorder as well. Among all the steps, payload delivery was the fastest one which required 58.4 seconds on average. Navigating from base to goal and goal to base took 83 and 101 seconds on average respectively.

Time for payload delivery

15 payload delivery experiments with 3 different simulated defects were performed to find out the average time. During the tests, robot remained at the same place and payload delivery was executed by passing an order from HMI. Figure 4.23 shows the recorded time for payload delivery. On average, RIRS took 24seconds to complete the payload delivery. The minimum and maximum time recorded for payload delivery actuation was 22seconds and 27seconds respectively.

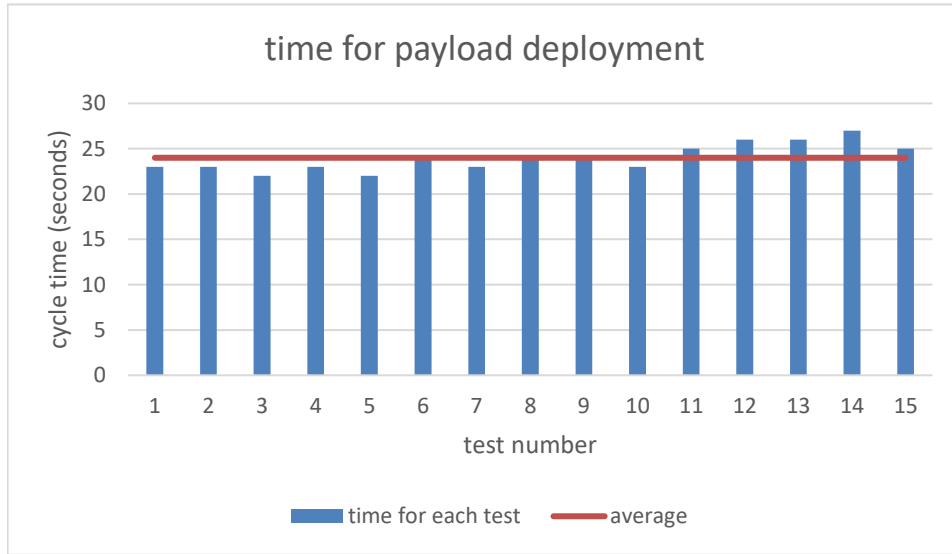


Figure 4.23 Cycle time for only payload delivery

4.4.2 Automatic trajectory generation

4.4.2.1 Experiment and data collection procedures

In many real-world repair tasks, such as welding, manipulators have to go through a set of waypoints. In this experiment, the framework for the RIRS to follow a set a waypoint consisting of 40 points in cartesian space are tested. RGB-D camera has been used to detect the object, as shown in Figure 4.24. Then "*find_object_2d*" package from ROS has been used to calculate the object position with respect to the camera. One useful feature from "*find_object_2d*" is that it can automatically create a tf frame for the detected object.

Once the object is added to the ROS tf frame, pose (position and orientation) of the object is extracted. Position of the object is considered as the centre of the set of waypoints. A python script has been created to calculate the waypoints in cartesian coordinates. The target of this automatic waypoint generation experiment is to create waypoints on a circular trajectory. A predefined radius and geometric equations are used in the python script to calculate the waypoints.



Figure 4.24 Automatic waypoint generation based on detected object

MovelIt! is used to generate the controller for the actuation in these experiments. Once the waypoints are calculated, they are passed to the MovelIt. MovelIt will set the goal for the end-effector as those waypoints sequentially.

To determine the performance of the actuation work, the end-effector position for 40 waypoints and the object position from *find_object_2d* have been recorded and plotted. Distance of all these waypoints is compared with the predefined radius of the circular trajectory. If the distance of these waypoints is same as the predefined radius, then these waypoints will create a circle. This test has been carried out 3 times by changing the position of target object. Figure 4.25 shows the position of different object determined by RGB-D camera with respect to the robot. All the objects are placed parallel to the robot. So, theoretically all the objects will have same x coordinates. Developed algorithms are provided in the appendices section D.1, D.3, and D.4.

4.4.2.2 Results

To find out the performance of waypoint generation framework and actuation performance, 3 tests has been conducted. 3 separate objects have been placed

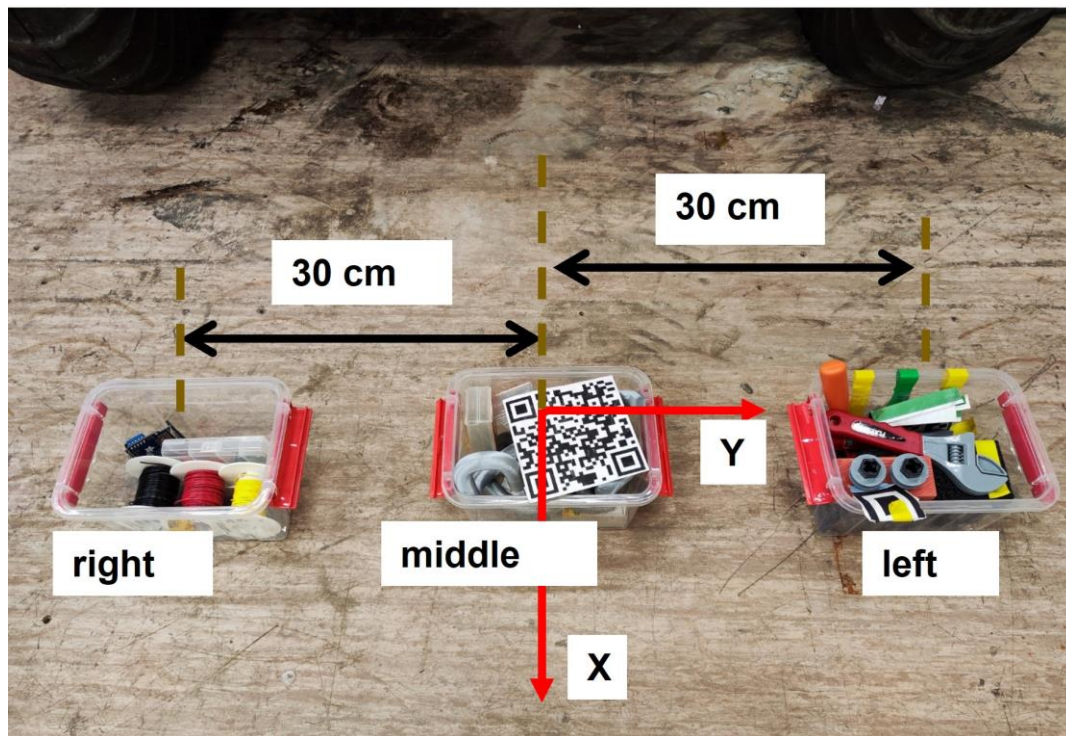


Figure 4.25 Different object for waypoint generation

in 3 different positions. As shown in figure 4.25, objects were placed 30cm apart in y-axis while maintaining the same position in x-axis. The positions of the objects were selected arbitrarily, with keeping the maximum reachability (1200mm) condition of the manipulator in mind. Once the object detection was successful, manipulator started the actuation. End-effector poses and centre of the objects for the 3 tests has been recorded and then plotted. In all the 3 tests, UR10e were able to create a circular trajectory based on the waypoints, as per Figure 4.26. Apart from that, distance between the points and the centre (object centre determined by the RGB-D camera) are calculated. For all the points, the distance was 0.2m which was same as the predefined radius of the circle. Moreover, the centre of the object was also plotted to find out the accuracy of the multiple object detection based on the RGB-D camera, shown in Figure 4.26.

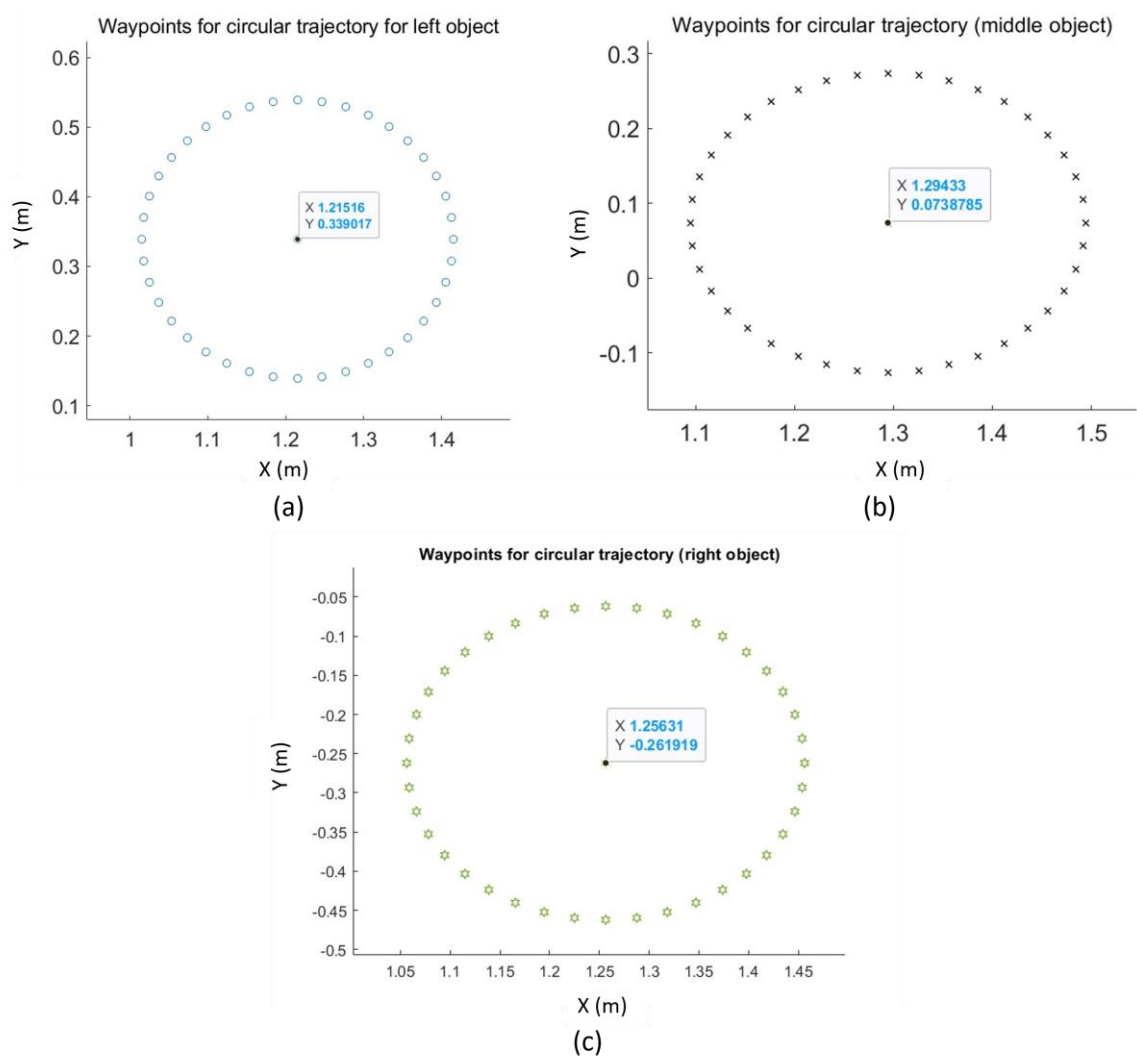


Figure 4.26 Automatic waypoint generation based on detected object

5 DISCUSSION

5.1 General Discussion

The popularity of railways is increasing because the railway is considered one of the greener, safer, and more reliable means of mass transport. These rising demands are stretching the usage of railway tracks which is a crucial part of the railway asset network. The quality of track also matters for the vehicle-overhead infrastructure performance [217], and the passenger comfortability [218]. In response to these challenges, there is a growing interest in leveraging autonomous and robotic solutions, particularly mobile manipulators, to assist or even replace human workers in railway track inspection and repair tasks. Mobile manipulators represent a specialized class of robots that have the potential to enhance the existing railway inspection and repair systems. An example of such a robotic solution is RIRS, which has been meticulously designed, developed, and rigorously tested for track inspection and repair.

RIRS operates in two distinct navigation modes: off-track and on-track. While the on-track navigation mode introduces some frictional power loss between the robot's pneumatic wheel and the trolley's contact roller, precise platform positioning within the control system has effectively addressed this issue.

To ensure the effectiveness of RIRS, a comprehensive simulation environment was established as part of a systematic system engineering approach. This environment allowed for thorough algorithm performance testing before deploying the robot in real-world scenarios. Additionally, a detailed kinematic analysis of RIRS revealed that its joint and link configurations vary between the different navigation modes. Furthermore, extensive repeatability and precision testing demonstrated the robot's suitability for track inspection and repair tasks. Repeatability tests were conducted to pinpoint defect locations from a fixed base, while precision tests assessed the impact of the mobile base on a UR10e industrial manipulator.

Building on these capabilities, a 3D reconstruction method was developed to achieve accurate inspections. This method effectively showcased the advantages of combining an industrial manipulator with vision-based sensor fusion for railway track inspection.

Finally, two distinct repair task frameworks were created and rigorously tested. The first framework, "human-in-the-loop," involved extensive human involvement through a two-way feedback system. In contrast, the second framework focused on automatic trajectory generation for autonomous repair task completion.

In conclusion, the integration of autonomous and robotic solutions like RIRS holds great promise for the future of railway track inspection and repair. By addressing the challenges of increasing demand and maintaining track quality, these technologies are poised to play a pivotal role in ensuring the continued success and sustainability of railways as a mode of mass transportation.

5.2 Kinematics, repeatability, and precision analysis of RIRS

5.2.1 Kinematic analysis of RIRS

Kinematics is important for mobile robots as it helps to identify the robot's pose (position and orientation) in the workspace at every instance. As RIRS is a mobile manipulator, kinematic analysis is important to understand the performance in inspection and repair tasks. In total, RIRS has 24 joints: 6 joints in the UR10e manipulator, 6 joints in the warthog, and 12 joints in the trolley. Depending on the navigation mode, off-track and on-track, RIRS will have different joint configurations.

5.2.1.1 Changes in kinematics for on-track vs. off-track configuration

As the structure and number of joints can be reconfigured based on on-track and off-track operation, structural investigation and joint simplification are important for next-stage programming to ensure optimal trajectory, collision, and obstacle avoidance. Based on the navigation mode, many properties and features such as number of joints, vibration, speed loss, arm position will change for RIRS as follows:

- **Number of joints:** Total number of joints changes based on the navigation mode. During on-track mode, total number of joints can be reduced to 7 after making some simplification as the robot's movement are restricted only in forward or backward motion. On the contrary, all the joints from the warthog UGV and UR10e manipulator are considered during off-track operation.
- **Vibration:** On-track operation and off-track operation have different types of complexity such as the vibration will be different in on-track than the off-track. For off-track operation, pneumatic tyres will absorb most of the vibration. During on-track navigation, RIRS will run smoothly because of the rail wheel.
- **Speed loss:** There will be a speed loss during on-track navigation because of the frictional contact between the pneumatic wheel of warthog UGV and contact roller of the trolley. This velocity loss can be measured accurately as it changes with the tyre pressure, contact area between wheel and roller, and warthog position on the trolley.
- **Position of arm:** Also, the arm position will be different in on-track and off-track operation. In off-track navigation mode, the UR10e manipulator can reach the ground but in off-track navigation there is around 0.45m height gap between the end-effector and ground.

5.2.1.2 Benefits of Road-rail Conversion

On-track and off-track operational capability has made this proposed RIRS unique and provides many advantages over the conventional robotic and autonomous system. This research is intended to provide an autonomous solution for railway track (on-track and off-track) inspection and repair with the help of RIRS. By integrating a manipulator with an UGV, a multi-purpose mobile manipulator has been created for demonstration of the RIRS architecture principles and capability for on-track maintenance tasks. The functionality of the mobile manipulator has been extended for using it for on-track operation by using a simple trolley.

As the RIRS has the capability to move on the track and on the side of the track, it has capability to perform not only track maintenance but also many tracksides maintenance tasks such as removing vegetation, removing debris, inspecting, and repairing trackside furniture, inspecting point machines etc. While the available robotic and autonomous systems in railway maintenance only focus on one area of the asset, developed RIRS can perform task in wider range of areas which is the biggest advantage of the proposed maintenance system.

5.2.2 Repeatability and Precision of RIRS

5.2.2.1 Inspection

During the inspection, the whole RIRS system is moving, and positional error was 32.96mm or 0.27% to the whole distance (distance between initial position of RIRS and detected defect position) 3.76m. The main reasons for this positional inaccuracy were the slippage between warthog wheel and contact rollers of the trolley and initial RIRS position mismatch. Track monitoring and inspection vehicles such as Track Recording Vehicle (TRV) and New Measurement Train (NMT) uses speed and GPS location to identify the defect location [219], [220]. As the GPS signal is not available or weak in many locations along the track, location accuracy of fault or defect can vary from 1m to 20m. A small percentage of error demonstrate the capability of RIRS system for rail inspection job. Accuracy can be improved by incorporating the slippage amount in the program and setting the RIRS initial position more precisely.

5.2.2.2 Repair

During repair, the manipulator was able to reach the position with very high accuracy, less than 1mm deviation from the mean position and around 2mm in z-axis for every repair motion. A bigger error in z-axis was expected as only monocular camera was used which is not good for finding distance in z direction. High precision proves the feasibility of the manipulator for RIRS. This small deviation may be caused by the vibration of robot base during arm manipulation and ViSP auto tracker object position calculation. For finding the end-effector position more accurately, laser-based sensors can be used instead of ViSP auto

tracker and robot base vibration can be reduced by improving the path planning and speed of manipulator.

5.3 Inspection (3D modelling)

5.3.1 Image Capturing

3D model generated from SfM highly depends on the provided images. Among all the factors, two are very important while capturing and selecting images for SfM: visibility of objects in the image frame and matching between the sequence of images. Images captured for the modelling depends on the manipulator motion trajectory. For target object 3D reconstruction experiment, 20 poses of the manipulator have been pre-set manually to ensure that 100% of the objects are visible inside the camera frame, shown in Figure 5.1. As the manipulator motions were predefined manually without any trajectory planning, the angular difference between each pose is not consistent. It would cause a density inconsistency of the model as shown in Figure 4.15(b) and (d)). Although it has no loss for the object dimension characterization, this density inconsistency may cause some losses on surface texture details. To improve the model density, multiple manipulator arm trajectory strategies (circular, spherical, and zig-zag etc.),

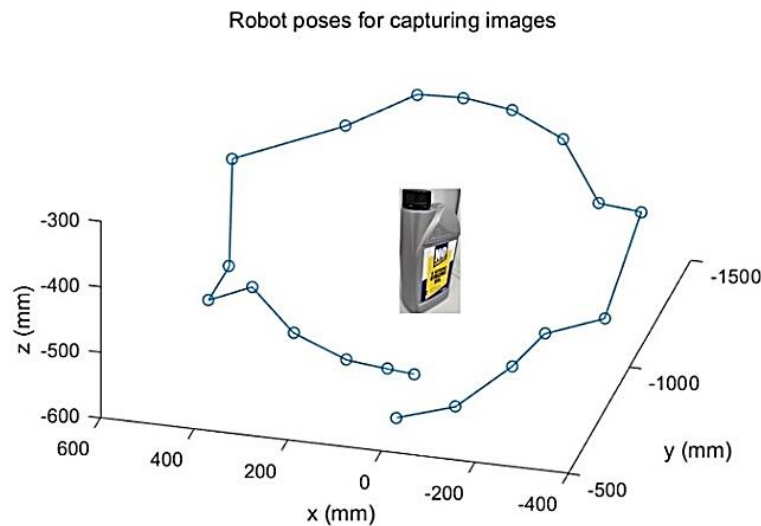


Figure 5.1 Manipulator poses for capturing images for data-rich 3D modelling technique

autonomous trajectory selection, and point cloud interpolation enhancement are worthy of investigating to provide optimal modelling performance in the future research.

5.3.2 Dimensional Accuracy

In the experiments, the scaling factor of the 3D reconstructed model from SfM was corrected using the known dimension of the wood platform and the sleeper. The absolute error was then calculated for the scaled 3D model with respect to the actual dimension of the object. The dimensional error observed in Table 5.1 for the bottle was clearly noticeable at approximately 10.1%, while the error for the rail cut section is clearly more accurate with an estimated error of 2.6%. The error difference in these 2 objects is attributed to their shape. Bottle has round edges while the rail section has sharp edges. Corners with sharp edges are easy to measure than rounded or chamfered edges. Corners are important features for SfM to capture and register for creating a 3D model and to find the dimension of the 3D model. Another possible reason is that, during the scaling correction process, the chosen 4 reference points are manually selected from the background using the point-pair registration method. References are selected from the wooden platform for laboratory scenario and from standard British railway track gauge for track scenario. Low light in indoor and distortion of the

Table 5.1 Dimensional comparison between actual and 3D reconstructed model

		Bottle	Rail section
Length	Ground truth (cm)	10.0	13.8
	Reconstructed model (cm)	10.9	14.16
	Error (%)	9	2.6
Width	Ground truth (cm)	5.5	9.8
	Reconstructed model (cm)	6.06	9.97
	Error (%)	10.1	1.7
Height	Ground truth (cm)	22.6	15.9
	Reconstructed model (cm)	23.4	15.95
	Error (%)	3.57	0.31

wooden platform are the biggest factor for the 4-reference point deviation, which results in the higher dimension error for the bottle.

Additionally, the weight of the rail section can be calculated if required using the information measured from the reconstructed model. For example, as per the British railway standard, weight of a rail section is 60 kg/m for a high-speed railway track. Multiplying the length of the reconstructed model with the per unit weight will give the total weight of the reconstructed model of the rail section.

5.3.3 Realistic Scenario Testing

The realistic railway track test proved the feasibility of the proposed method. Figure 5.2(a) shows the position of the RIRS in terms of global map using the GPS output which is important to localize the RIRS in the Mission Control. Figure 5.2(b) exhibits 2 distinct robot perceptions in the forms of a depth image from the RGB-D camera and a point cloud of the surrounding environment from LiDAR. Finally, Figure 5.2(c) gives an upgraded site-condition model for repair interaction. In this experiment, the tested target object is a pre-set rail section in the middle of the gauge. However, the proposed technique can be easily mirrored to other track components or a foreign obstacle like fastener, nut, fishplate, etc. As the UR10e manipulator used in RIRS is not long enough to visually inspect the outer side of left or right rail, target object was placed in the middle of the track. The experiment was implemented in the daylight with good illumination which is paramount to achieve better 3D modelling accuracy. In the future, more arm-fixed illumination lights might be necessary at the night. For further improving the impact, the method is to be integrated with NDT results, GNSS geo-location, and the assets survey map to upgrade the infrastructure management database.

5.3.4 Flexibility of the Proposed Method

Data-rich high-resolution model of the target environment is one of the desirable features in digitalization of railway maintenance technology. The proposed technique can not only build the data-rich 3D reconstruction for the maintenance track component but also can be used for the railway track survey purposes. In combination with vehicle autonomous control for track travel, this technique can

create the 3D reconstruction of the whole track. Figure 5.3 presents a pre-experiment of a modelling along one side of the track around 5.97m, which includes important track components like fastener clipper, nut & bolt, sleepers, and rail steel. The 3D model well captures the structures and relationship among

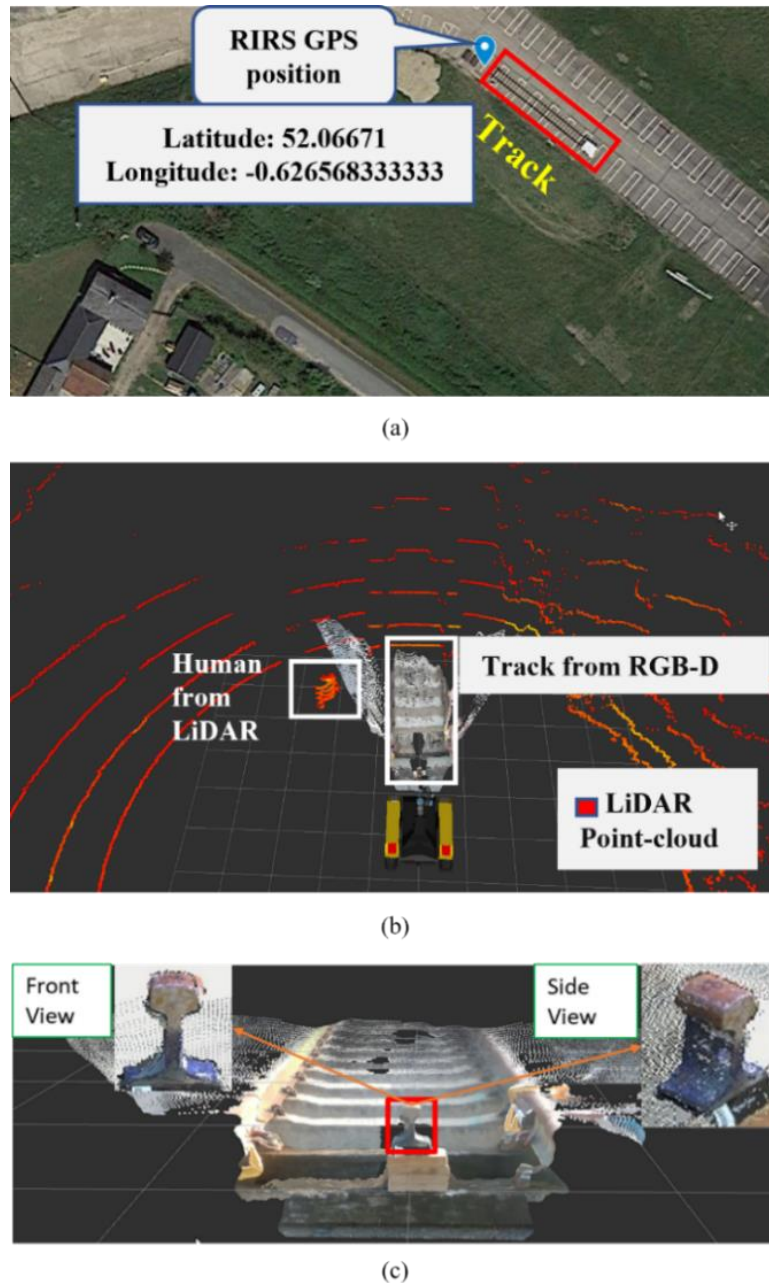


Figure 5.2 Improved RIRS perception with the fused model of maintenance object; (a) RIRS position in global map acquired from GNSS, (b) RIRS perception from RGB-D camera with the surrounding environment from LiDAR, (c) fused 3D model of object in robot perception

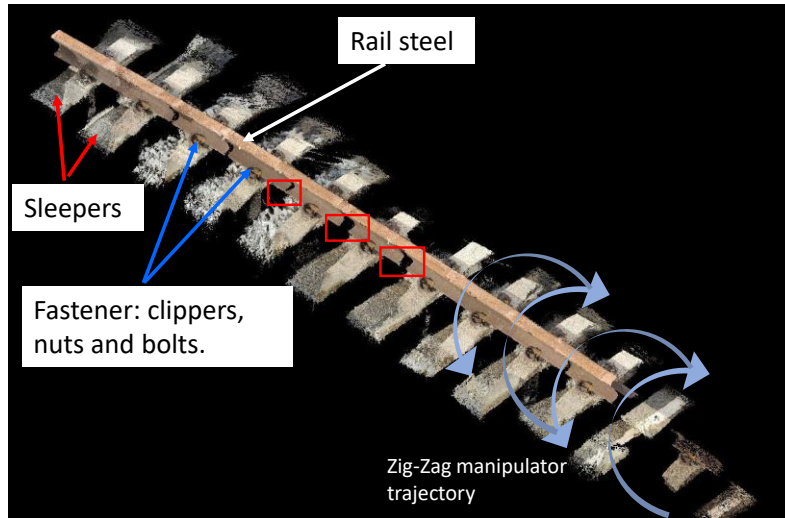


Figure 5.3 3D reconstruction of railway track using SfM with RIRS

all components, verifies the feasibility and potential of the proposed technique in track survey or maybe in geometry measurement.

In this trial, we performed the reconstruction implementation with an automated vehicle step forward moving and arm manipulation. It is clear that the orbiting arm trajectory is no longer appropriate for this trial, therefore a zig-zag shape trajectory is adopted to ensure the manipulation can cover all components of the track.

Most track details are well presented but there are several missing parts on the side of the rail (red boxes in Figure 5.3.). This is caused by the imperfection cooperation between vehicle moving and robotic arm manipulation. It can be concluded that the amount of image captured and arm manipulation for 3D reconstruction should be increased and more redundant compared to reconstruction when vehicle stands still. As vehicle moving, the motion blurry effect would cause the degradation of image quality and failed registration while reconstructing. To improve this, the optimal settings of vehicle moving speed and the resolution for arm manipulation sensing can be investigated in the future research. Nevertheless, this flexibility is a distinctive feature of the proposed

technique which will help to find a solution for the over-massive digital data computation to establish a smart digital infrastructure model.

5.4 Repair

5.4.1 Human-in-the-loop repair task

5.4.1.1 Communication and hardware architecture

Railway repair tasks require heavy tools and materials. Carrying these heavy tools require a hand pushed manual trolley or RRV. Warthog is designed to carry high payload. These HITL repair task experiments demonstrated the capability of an autonomous robot to assist human in railway repair tasks. During the process, a Human Machine Interface (HMI) was used to execute command on the robot. HMI provides safety benefits instead of running the commands directly on the warthog. Figure 5.4 shows the communication and hardware architecture of RIRS for HITL repair task framework.

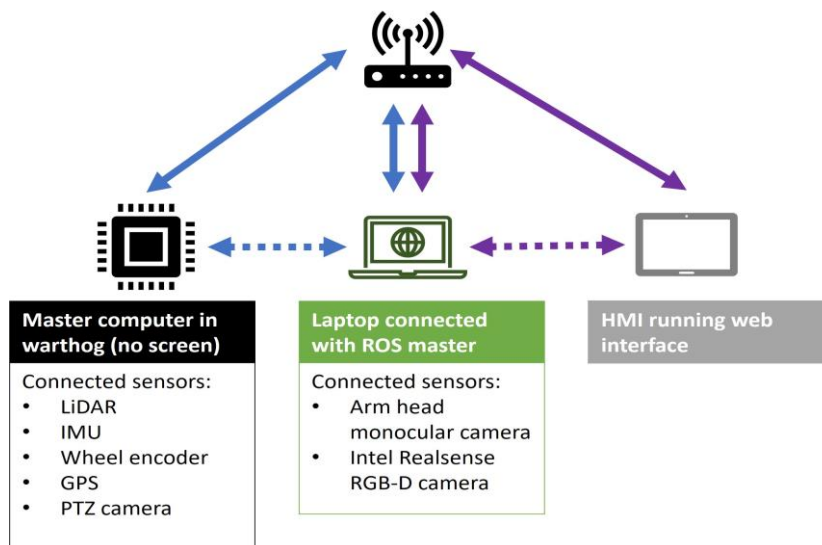


Figure 5.4 Communication and hardware architecture of HMI with computational devices

Master computer inside warthog UGV, do not have any screen which add some inconveniences to add any new executable scripts or modify the existing one. But ROS can maintain synchronized communication between multiple computational

devices using rosnode and rostopic. In RIRS, computer inside warthog is acting as a master computer while the connected laptop is acting as a slave. Master computer is running all the drivers, emergency stops and some sensors (LiDAR, IMU, wheel encoder, GPS, PTZ camera). On the other hand, slave laptop is running the monocular and RGB-D camera, processing the data, and transferring the output with the master via rosnode and rostopic. Moreover, all the executable scripts are running on the laptop. Finally, the HMI is a web interface which can be run on browser in a tablet or laptop. The slave laptop is running the *ros_bridge* package which establishes communication between the laptop and HMI. All the computational device and HMI are connected to the same local wi-fi network.

5.4.1.2 Latency

There is 20seconds difference between the minimum and maximum cycle time as per Figure 4.22. On the other hand, the difference between the minimum and maximum payload time is 5seconds as per Figure 4.23. Cycle time for 5 experiments and payload delivery time for 15 experiments should be same as the base and goal position are same and the manipulator requires same number of poses to deliver repair tools at every instances. These discrepancy in time calculation is coming from the communication latency. Reducing these discrepancies for communication latency will require extensive research. For example, optimizing the frontend GUI of HMI, compressing the camera feed and image while transferring between devices, improving the network quality, using industry standard communication protocol (such as DDS (Data Distribution Service)) might improve the latency. HMI and all the computational devices are connected through a local wi-fi network.

Further analysis of the HMI GUI was performed to assess its responsiveness and network latency when passing commands to the Warthog. The assessment was classified into the following 4 groups:

- Main control page loading
- Vehicle control
- Camera feed
- Image copying

The assessment comprised of analysing the time between sending of a command from the HMI GUI to the Warthog and getting an execution response message. 5 trials were performed for each data group and a summary of findings is presented in Table 5.2.

Table 5.2 Average network latency and standard deviation for HMI GUI components interacting with the Warthog robot

Data Group	Average Network Latency (sec.)	Standard Deviation
Main control page loading (only load once at the beginning)	8.09	0.96
Vehicle control	0.13	0.01
Camera feed	4.06	0.24
Image copying with SSH	3.61	0.40

The main control page loading time had the longest average response time because of the monitoring elements of the camera feed and the GPS map which are inherently image objects. The camera feed had the second highest latency for the already described reason that images consume high bandwidth.

Vehicle control, i.e., sending move and stop commands had the least average network latency of 0.13 seconds. The analysis was performed with commands for the Warthog to move at the speed of 0.5m/s at an acceleration rate of 0.1 over a distance of 0.5m in a lab environment in an off-track configuration. An interesting observation was also made in the similarity of network latency between execution of vehicle commands through ros_bridge and existing Python commands. The move commands were Python based and had an average latency of 0.13 seconds whilst the stop command was executed through ros_bridge with a marginally higher latency of 0.14 seconds.

The image copying analysis was performed by a concurrent transfer of 24 jpeg format images with a total size of 2.11MB from a remote directory on the Warthog to a local directory on the laptop using SSH. The average latency observed was 3.61 seconds which translates to a data transfer rate of 0.58MBps or 4.68Mbps

which is significantly below the 150Mbps rating of the Warthog's WIFI module. In this process command was initiated from the local laptop which was then transferred to the warthog's computer using SSH protocol. Then the image was copied to the local laptop through ROS middleware communication system. Moreover, there were multiple image files instead of a single file. Because of these reasons, there was a big latency and significant slow speed while copying multiple image files. Additionally, 150Mbps is the rated speed of the wi-fi module in ideal scenario but no test was carried out to find the actual speed in our laboratory condition. Note that in computer networking, data size of 1 Megabyte (MB) is equivalent to 8 Megabits (Mb).

5.4.1.3 GPS denied area or low accuracy of GPS

In these experiments, both the robot base and goal position are defined by the GPS. In many areas of the railway network (almost 20% area) are not covered by GPS signal and many places have a very weak GPS signal. As the human-in-the-loop repair framework take GPS position as an input, it will not work properly in the GPS denied area. It is difficult to determine the GPS accuracy at each instance as it depends on the weather condition, the number of available GPS connections etc.

Platform positioning has been implemented to reduce the dependency and error from the GPS data. Once the robot reaches the GPS goal, it will stop and move the manipulator into the inspection pose. Then RIRS will scan a certain distance both forward and backward at a slower speed to locate the defects. The algorithm already defines this distance to cover the GPS error. Though this increases the cycle time for the whole process, platform positioning ensures the detection of defects accurately.

Warthog UGV has a wheel encoder fused with IMU and uses Extended Kalman filter (EKF) to calculate the distance travelled by the robot. But during on-track navigation, the distance travelled by the RIRS depends on the rail wheel of the trolley. Due to variable velocity loss between the contact roller and pneumatic wheel, data from the wheel encoder are unreliable, leading us to define the base and goal position based on GPS. As the GPS output is not trustworthy in many

places of the rail network, a robust SLAM algorithm based on the fusion of multiple sensors (GPS, camera, or LiDAR) can be used to reduce the dependency on GPS which will improve the localization capability of the RIRS. More accurate localization and navigation capability will reduce the necessity of platform positioning at a slower speed hence reducing the overall cycle time.

5.4.1.4 Predefined object library and manipulator poses

Three different toolboxes are used in these experiments with correspondence to 3 different simulated defects. In this current framework, these toolboxes are predefined in the object library in relation to the simulated defect. This means if the RIRS finds a new defect, it will not deliver any toolbox even though the repair work can be done with the available toolboxes on the RIRS. Moreover, in the current scenario, only 1 toolbox is delivered by the RIRS. But in real-world track repair scenarios, multiple toolboxes might be required, which hasn't been addressed in the current version of the framework. These limitations can be solved in future iterations by considering all the repair scenarios and creating a more significant and effective library of repair toolboxes.

Manipulator poses are predefined in the human-in-the-loop. Joints angles configuration are predefined by providing 6 joint angles to reach and deliver the toolbox from the specific location based on the defect type. So, once the RIRS successfully identify the defect type, UR10e manipulator will reach to the specific toolbox (left, centre, or middle) and will deliver the toolbox. If the toolboxes positions are changed, then RIRS will fail to deliver the correct toolbox. Table 5.3 shows the fixed position and type of toolbox based on simulated defect. Because of fixed type and position of the toolbox and predefined poses of the manipulator, the delivery capability of RIRS is confined only to 3 items.

Table 5.3 Toolbox position and type based on the simulated defect type

Simulated defect	Toolbox type	Toolbox position
01	Mechanical tools	left
02	Electrical tools	centre
03	Pandrol clips	right

5.4.2 Automatic trajectory generation

Automatic trajectory generation algorithm can successfully generate circular trajectory around the object. The algorithm was successful in generating 40 waypoints to generate a circular trajectory using the defined 0.2m radius. Length of radius can be modified to change the size of the circle. Moreover, the number of waypoints can be altered to revise the waypoint density. In this framework, MoveIt is the controller for the manipulator. All the waypoints are calculated based on the input from the RGB-D camera, predefined radius, and number of waypoints from the user. Waypoints are passed into MoveIt once the algorithm finishes the calculation. As the waypoints do not modify the MoveIt controller, it is easy to modify the algorithm to generate waypoints for other shapes.

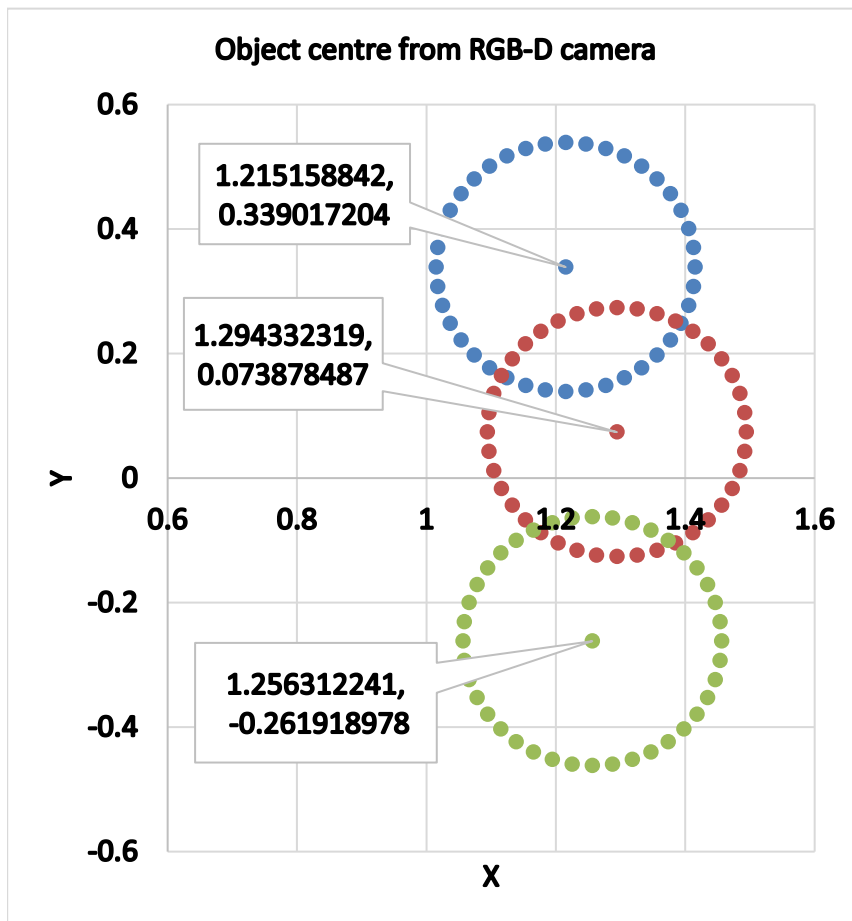


Figure 5.5 Different object for waypoint generation

Though the framework generated the circular trajectory properly, it failed to detect the object centre accurately. A ROS package named "*find_object_2d*" has been used to detect the object centre. As per the experimental setup, all the 3 objects should have same values in x and z coordinates. There was a gap of 30cm for left and right object from the middle object. From Figure 5.5 it is obvious that, the values for the x coordinates are not similar for all the objects. On the other hand, the distance along y axis for left and right object with respect to middle object are 0.265cm and 0.336cm respectively. Moreover, a small variation of 0.05cm along the z axis has been noticed.

6 CONCLUSION

This chapter will conclude the research work done in this thesis. In brief, this chapter will restate the research gap, summarize the findings and describe the impact of this research. Uniqueness of this research compared to previous research will be described in the beginning. Then, objective wise detail implications will be given. Afterwards, limitations and considerations will be provided. Finally, future recommendations will be provided to guide the future research.

6.1 Novelty: a multipurpose railway track maintenance robot

Aim statement: This thesis aims to develop a multipurpose Robotic Inspection and Repair System (RIRS) capable of performing improved railway track inspection by fusion of multiple sensors and demonstrating multiple simulated track repair tasks with and without human involvement.

This thesis introduced a novel multipurpose railway track maintenance robot which provides multiple advantages over previous research. As per the literature review, mobile manipulators are widely used in many industries for inspection and repair, but no example is found for railway. Moreover, most of the robots are only used for inspection in railway industry while repair tasks are mostly human dependent. Additionally, improved inspection method, reconstructed 3D model of the target area, provides more information which is helpful to the asset management system. Hence, the designed mobile manipulator, Robotic Inspection and Repair System (RIRS) comprising of a Warthog UGV and UR10e industrial manipulator, diminishes these technology gap by demonstrating an improved inspection and repair tasks.

As a multipurpose maintenance robot, this thesis demonstrated both the inspection and repair task with RIRS. An intelligent autonomous 3D reconstruction strategy with the autonomous motion of RIRS and sensor fusion is proposed to support railway track inspection tasks. The proposed method created 3D model of the target object with acceptable dimensional accuracy (see section 6.2.3 for more details). Moreover, apart from the inspection, the proposed

method can be used for digitalization of railway track survey or creating digital twin of railway track.

Then, the design and experiment of two simulated repair task frameworks establish the usage of RIRS in track repair. The first framework involved human-in-the-loop delivering the appropriate repair tools based on the defect type, while the second actuated a circular motion over the defected area which represented some potential use of manipulator for repair tasks such as welding, grinding, or drilling in a specific pattern (see section 6.2.4 for more details). Demonstrating the simulated repair tasks by the RIRS exhibited the capabilities of autonomous robotic systems for track repair activities.

6.2 Conclusions on the specific objectives

6.2.1 Gap identification and Simulation of a RIRS

Objective statement: To simulate and design a Robotic Inspection and Repair System (RIRS) by identifying the research gaps in the current railway track maintenance technology after reviewing the conventional and state-of-the-art maintenance technology in railway and other industries.

At the start of this research, railway track and trackside environments were briefly introduced (section 1.5) to differentiate and find the unique characteristics of railway track environments. Afterwards, shortcomings of the current track maintenance technology and prospects of mobile manipulator was discussed to show the prospects of mobile manipulator for railway track inspection and repair tasks (section 1.6). Subsequently, an extensive review of the current railway track maintenance technology unveiled the little usage of robotic technology for track inspection and repair (section 2.1 and 2.2). Next, special focus was given to mobile manipulator as the mobile manipulator provides dexterity and mobility which are 2 important features of a multipurpose robot. Though most mobile manipulators are based on ground-based robots (54%), researchers are also looking into other types of base robots, such as aerial, marine, crawling or climbing. In terms of areas of application, the high-voltage power transmission

line and nuclear industry are the most significant with 20% and 12% usage respectively.

This rigorous literature review led to the primary research gap that a multipurpose mobile manipulator is necessary for the railway track maintenance system to improve the safety of human and increase the cost effectiveness. Currently, the railway track inspection mostly relies on 2D data. The proposed robotic 3D modelling technique provides structural dimension, colour, and texture of the target object which will improve the inspection capability and help the digitalization of asset management system. Another gap was the almost no usage of robotic system for track repair tasks although other industries are using them for a long time. For successful completion of the research and solution for these research gap, Verification and Validation (V&V) model of systems engineering approach was developed for coding and algorithms (section 3.2.3). A simulation model was developed in Gazebo to ensure the safe operation and observe the expected output of the developed code before experimenting in real robot (Appendix A.36.4.3A.3 and section 4.1).

6.2.2 Kinematics, repeatability, and precision of RIRS

Objective statement: To analyse the kinematics of the RIRS in different navigation scenarios and then examine the repeatability and precision of an industrial manipulator on a mobile base with varying payload for railway track inspection and repair tasks.

In RIRS, rail wheels of railway trolley provide mobility during on-track navigation while pneumatic tyres of Warthog UGV provides mobility during off-track navigation. RIRS is capable performing actuation because of the attached UR10e industrial manipulators. Based on 2 different navigation mode (on-track and off-track navigation), links and joints configuration of RIRS changes. Kinematic analysis of RIRS showed that there is total 24 joint in RIRS; 12 for trolley, 6 for warthog and 6 for UR10e (section 3.3.1 and 4.2.1). But, after simplification, number of joints can be reduced to only 7 and 12 for on-track and off-track navigation respectively (section 5.2.1). This understanding of the kinematics is important to develop code for RIRS.

Usually, industrial manipulators are designed and used in the factory or shopfloor attached on a fixed base. But, in the RIRS, the UR10e has been attached to an UGV instead of a fixed base. As the manipulator in RIRS vibrates during motion, the repeatability of positional accuracy and precision of the manipulator for the proposed RIRS system was investigated for railway track inspection and repair tasks. A set of 10 test was conducted to find the repeatability of the defect position from base position (section 3.3.2 and section 4.2.2). The experimental results confirmed that RIRS was successful in finding the defect position repeatedly with a maximum error of 0.27% for 3.7m total distance (section 5.2.2). Additional 10 tests, consisting of multiple payload weights (1.75kg, 4.1kg and 5.9kg) on the manipulator end, were implemented to measure the precision of repair moves (section 4.2.2). The robot manipulator could reach the same position with 1mm precision with different payloads (section 5.2.2). Railway inspection and repair task requires higher precision and accuracy. For example, as per the standards of Network Rail UK, maximum flatness tolerance is $\pm 1\text{mm}$ for vertical alignment - running surface for welding of plain rail line [221]. The results of repeatability and precision proved the usability of the RIRS for railway track inspection and maintenance jobs.

6.2.3 Improved Inspection technique

Objective statement: To propose and test an improved inspection method by generating a 3D model of the maintenance target using vision-based sensor fusion to eliminate the shortcomings of RGB-D camera.

Image based visual inspection can find large surface defects and rail profile wear. But, if a single camera has been used, 2D image based inspection method can hardly find defects on side of the rail [103] based on the fixed position of the camera. Moreover, 3D modelling of the target area can not only detect defects on the side but also provide information for repair planning or robotic repair actuation as 3D model provides more information than 2D image [103], [106]. A 3D reconstruction technique for railway track components, using a robotic sensing system on the RIRS, has been proposed, developed, and tested in this research as an improved track inspection method. By taking advantage of RGB

and RGB-D camera sensor fusion, and incorporating a mobile manipulator's flexibility, a 3D model of the track component was reconstructed with accurate structural dimension, colour, weight, and surface texture. The dimensional accuracy for the developed 3D reconstruction technique was 2.6% for a rail section in railway test environment (section 5.3.2). Moreover, colour and surface texture of the reconstructed model were visible in Figure 4.15, Figure 4.16, and Figure 4.18.

In addition, a trial of modelling the track with the RIRS demonstrated the flexibility and potential of the proposed technique for track survey or geometry measurement (section 5.3.4). This 3D reconstruction method can be used as an autonomous digital twin model generator for the railway maintenance target to support inspection, condition monitoring, and repair tasks. Moreover, the 3D model can provide intuitive visualization, contextualization foundation, remote support, repair preparation and planning to support human-involved maintenance tasks if Augmented Reality (AR) systems are combined in future applications. It provides an automation contribution and data-rich enhancement for railway maintenance digital twins and intelligent infrastructure management.

6.2.4 Repair tasks framework

Objective statement: To demonstrate the usability of a mobile manipulator for multiple railway repair tasks with and without human involvement and investigate the performance.

Two repair tasks frameworks have been designed and tested to demonstrate the capability of RIRS for railway track repair tasks.

- Repair task framework 1: Human involvement in multiple steps, is called the Human-in-the-loop (HITL) repair task framework (section 3.5.1).
- Repair task framework 2: Automatic trajectory generation framework creates circular waypoints after detecting the target object (section 3.5.2).

The HITL repair task framework consists of 3 major steps where the human operator passes commands to the RIRS through Human Machine Interface (HMI). First steps involve the travelling of RIRS to GPS goal, second step is the

detection and delivery of the repair tools and final step is returning to base. In all the steps, a human operator used the HMI to pass the command to the RIRS. After conducting 5 tests, total cycle time for the HITL repair task was found 242 seconds. Moreover, average payload delivery time was 24 seconds after 15 tests which proved the consistency of the framework (section 4.4.1). Satisfactory results in multiple experiments exhibit the capability of RIRS in assisting humans in track repair tasks.

In the second repair task framework, RIRS was given a target object to generate a set of waypoints in a circular trajectory. This repair task framework does not need human intervention in the middle. RIRS detected the object and calculates the object pose. Then, a circular trajectory was generated with the predefined radius and number of waypoints.

3 tests were conducted by placing 3 different objects in separate locations (section 4.4.2). These tests results verified that RIRS could detect various target objects in distinct positions and actuate the end-effector through a set of waypoints in a circular trajectory based on the defined radius. This autonomous trajectory generation established the potential use of RIRS for different types of repair work, such as welding, grinding, drilling where the end-effector needs to follow a certain trajectory.

6.3 Limitations of this research

6.3.1 Limitations of the demonstrator

This thesis successfully demonstrated improved track inspection methods and simulated repair tasks. 2 modes of navigation capability, on-track and off-track navigation modes are unique features of RIRS compared to other robotic and autonomous systems used for maintenance tasks. However, this research made some simplification and assumption to ensure the maximum time spent on achieving the objectives. For example, using QR code as a simulated defect instead of real defect for inspection and repair tasks was a simplification of this research. As the main focus of this study was to develop a robotic technology for track inspection and repair, simplification of the defect supported the

development of RIRS. Additionally, the RIRS performs inspection at a slow speed (maximum 1m/s) which is slower than many current inspection methods. Moreover, the maximum payload capacity is only 10kg which is not feasible for many repair tasks.

6.3.2 Limitations of the inspection method

For the improved inspection method, a monocular camera was used to take pictures of the target object to create a 3D model using Structure-from-Motion (SfM) technique. Though the RIRS detects the target object autonomously, trajectories for capturing images, post-processing those images using SfM, scale correction, and fusion were completed manually. Nevertheless, the camera-based object detection algorithms were sensitive to lighting conditions, illumination, and weather condition. That's why the experiments of improved inspection methods in indoor environment were conducted in controlled and artificial lighting conditions while the outdoor test was conducted in a cloudy day to minimise the illumination and reflection of sunlight.

6.3.3 Limitations of the repair framework

In the Human-in-the-loop repair task framework, RIRS assisted human by delivering repair tools. In these experiments, only 3 types of repair tools and 3 simulated defects were used. Moreover, the poses of the manipulator for delivering the repair tools were predefined. Pose of the manipulator had relation only with the simulated defect and did not verify the tool type before delivering. The repair tools should be set by strictly following the same sequence and specific position on the RIRS. Additionally, the loading of the Warthog on the trolley exemplified the exposure to tolerances for manipulator pose: if incorrectly positioned, there was a need to program offsets.

In the automatic trajectory generation experiments, "*find_object_2d*" ROS package was used which uses template matching for objection detection and pose estimation. Hence, this method requires a library of templates for object detection. Otherwise, this method will fail to detect the object and estimate the object pose.

6.4 Future recommendations

RIRS showed much potential to reform the railway asset management system. However, extensive research is needed to improve the performance of the system by diminishing the shortcomings of current research. Moreover, introducing machine learning algorithms, mixed reality, digital twins, and teleoperation can enhance the effectiveness and efficiency of RIRS in the future.

6.4.1 Introducing machine learning for better image processing

In this research, RGB and RGB-D camera were used during inspection and repair for defect detection, pose estimation using open-source ROS packages such as “*ViSP_auto_tracker*” and “*find_object_2d*”. Though both of these ROS packages were successful in detecting simulated defect (QR code) in good lighting condition, they might fail to detect real rail defect in difficult lighting condition. Machine learning algorithm and deep neural networks can be used to improve the defect detection and pose estimation. A dataset of images created with real rail defects in unstructured lighting condition can train an image processing algorithm for object detection in real world scenario.

6.4.2 Using teleoperation and mixed reality

RIRS can work in autonomous mode, or a human can operate the robot through an HMI. This command-and-control system limits the application of the system only to a few predefined use cases. Hence, there is a room of improvement to introduce more generality in task planning. Moreover, this also failed to completely remove human footsteps on the track. The dexterity and versatility of RIRS can be improved by using teleoperation and mixed reality. Teleoperation is a method of operating the robot from a remote location that is an intuitive way to transfer human expertise into a robotic system. Mixed reality is a safer and simpler way of communicating with the physical and virtual environment. Controlling the RIRS with a combination of teleoperation and mixed reality will enable remote operation, improve versatility, ensure knowledge transfer, and enhance communication, increase safety, introduce generalization in task planning and improve actuation for payload deployment. Different repair tasks

require different tools. Hence, a reconfigurable end-effector is desirable. To maintain the vast railway network, a fleet of robots will be useful. Controlling a fleet of robot with teleoperation and mixed reality will require extensive research before deployment. Moreover, current railway standards, maintenance compliance, training of technology and change of human mindset should be investigated before deploying modern robotics technology for track inspection and repair tasks.

6.4.3 Establishing digital twin using robotic technology

This research created a 3D model of the target object to improve the current inspection method, which can be utilized as a digital twin for the railway inspection task after fusion with non-destructive testing. This modelling technique using mobile manipulator will reduce the human involvement for creating a digital twin of the railway track. Furthermore, the dense 3D model of the environment can be used to train an AI-based robot manipulator for further repair moves with high accuracy and flexibility. Digital twin of railway track can help asset management system by enabling predictive maintenance by continuously monitoring the health of the track based on real-time data such as vibrations, temperature, moisture. This predictive maintenance technique can help avoid unexpected downtime and reduce repair costs. Additionally, digital twin technology can simulate and analyse the behaviour of the track under various conditions, such as heavy loads, weather changes, and train speeds. This can help identify potential issues and optimize track maintenance and repair strategies.

REFERENCES

- [1] C. Maggs, *Steam Trains: The Magnificent History of Britain's Locomotives from Stephenson's Rocket to BR's Evening Star*. Amberley Publishing Limited, 2014.
- [2] L. T. C. Rolt, *George and Robert Stephenson: The Railway Revolution*. Amberley Publishing Limited, 2009.
- [3] 'Fatalities per mode of transport in Great Britain', Statista. Accessed: Oct. 20, 2022. [Online]. Available: <https://www.statista.com/statistics/300601/average-number-of-fatalities-according-to-transport-in-the-united-kingdom/>
- [4] 'Which form of transport has the smallest carbon footprint?', Our World in Data. Accessed: Aug. 05, 2022. [Online]. Available: <https://ourworldindata.org/travel-carbon-footprint>
- [5] 'A Short Guide to Network Rail - National Audit Office (NAO) Report', National Audit Office. Accessed: Jun. 16, 2020. [Online]. Available: <https://www.nao.org.uk/report/a-short-guide-to-network-rail/>
- [6] K. Williams, 'The role of the railway in Great Britain'.
- [7] 'Rail Factsheet - December 2019', Department of Transport, Dec. 2019.
- [8] R. Horgan, 'Ferrovia boss on why high speed rail costs more in UK', New Civil Engineer. Accessed: Feb. 10, 2023. [Online]. Available: <https://www.newcivilengineer.com/latest/ferrovial-boss-on-why-high-speed-rail-costs-more-in-uk-12-07-2022/>
- [9] P. Condon and K. Dow, 'A Cost Comparison of Transportation Modes', Jan. 2008.
- [10] 'Industry organisations | Office of Rail and Road'. Accessed: Mar. 13, 2023. [Online]. Available: <https://www.orr.gov.uk/about/who-we-work-with/industry>
- [11] P. Scarf, R. Dwight, A. Mccusker, and A. Chan, 'Asset replacement for an urban railway using a modified two-cycle replacement model', *Journal of the Operational Research Society*, vol. 58, Sep. 2007, doi: 10.1057/palgrave.jors.2602288.
- [12] 'Asset management', Network Rail. Accessed: Mar. 03, 2023. [Online]. Available: <https://www.networkrail.co.uk/running-the-railway/looking-after-the-railway/asset-management/>
- [13] 'BS EN 13306-2017 Maintenance - Maintenance Terminology'. BSI Standard Publication, Nov. 2017.
- [14] 'On the Definition of Maintenance Management', in *The Maintenance Management Framework*, in Springer Series in Reliability Engineering. ,

- London: Springer London, 2007, pp. 3–10. doi: 10.1007/978-1-84628-821-0_1.
- [15] M. Holmgren, 'Maintenance-related losses at the Swedish Rail', *Journal of Quality in Maintenance Engineering*, vol. 11, no. 1, pp. 5–18, Jan. 2005, doi: 10.1108/13552510510589343.
 - [16] M. Papaelias, A. Amini, Z. Huang, P. Vallety, D. C. Dias, and S. Kerkyras, 'Online condition monitoring of rolling stock wheels and axle bearings', *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 230, no. 3, pp. 709–723, Mar. 2016, doi: 10.1177/0954409714559758.
 - [17] J. Preston, G. Wall, R. Batley, J. N. Ibáñez, and J. Shires, 'Impact of Delays on Passenger Train Services: Evidence from Great Britain', *Transportation Research Record*, vol. 2117, no. 1, pp. 14–23, Jan. 2009, doi: 10.3141/2117-03.
 - [18] Network Rail (Firm), *Network Rail Limited's Annual Report and Accounts 2019*. 2019.
 - [19] 'New Measurement Train (NMT) - Network Rail'. Accessed: Jun. 17, 2020. [Online]. Available: <https://www.networkrail.co.uk/running-the-railway/looking-after-the-railway/our-fleet-machines-and-vehicles/new-measurement-train-nmt/>
 - [20] 'Doctor Yellow', *Wikipedia*. May 04, 2020. Accessed: Jun. 17, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Doctor_Yellow&oldid=954775978
 - [21] 'China Railway comprehensive inspection trains', *Wikipedia*. Apr. 26, 2020. Accessed: Jun. 17, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=China_Railway_comprehensive_inspection_trains&oldid=953328584
 - [22] 'Seasonal track treatment and weather support fleet', Network Rail. Accessed: Jun. 17, 2020. [Online]. Available: <https://www.networkrail.co.uk/running-the-railway/looking-after-the-railway/our-fleet-machines-and-vehicles/seasonal-track-treatment-and-weather-support-fleet/>
 - [23] 'Britain's first "workshop on wheels" set to revolutionise railway maintenance', Network Rail Media Centre. Accessed: May 24, 2022. [Online]. Available: <https://www.networkrailmediacentre.co.uk/news/britains-first-workshop-on-wheels-set-to-revolutionise-railway-maintenance>
 - [24] 'High Output machines', Network Rail. Accessed: May 24, 2022. [Online]. Available: <https://www.networkrail.co.uk/running-the-railway/looking-after-the-railway/our-fleet-machines-and-vehicles/high-output/high-output-machines/>

- [25] 'Our leaf-busting operation', Network Rail. Accessed: May 24, 2022. [Online]. Available: <https://www.networkrail.co.uk/stories/our-leaf-busting-operation/>
- [26] 'Track renewals', Network Rail. Accessed: May 24, 2022. [Online]. Available: <https://www.networkrail.co.uk/communities/living-by-the-railway/track-renewals/>
- [27] K. Smith, 'Britain outlines £44bn maintenance and renewals spending plan for 2024-2029', International Railway Journal. Accessed: Mar. 13, 2023. [Online]. Available: <https://www.railjournal.com/financial/britain-outlines-44bn-maintenance-and-renewals-plan-for-2024-2029/>
- [28] 'Annual-expenditure-2019-20', Network Rail, 2020. Accessed: Mar. 13, 2023. [Online]. Available: <https://www.networkrail.co.uk/wp-content/uploads/2021/02/Annual-expenditure-2019-20.pdf>
- [29] 'About S2R', Europe's Rail. Accessed: Mar. 13, 2023. [Online]. Available: <https://rail-research.europa.eu:443/about-shift2rail/>
- [30] R. K. W. Vithanage, C. S. Harrison, and A. K. M. M. DeSilva, 'Enhance 3D Point Cloud Accuracy Through Supervised Machine Learning for Automated Rolling Stock Maintenance: A Railway Sector Case Study', in *2018 International Conference on Computing, Electronics Communications Engineering (iCCECE)*, Aug. 2018, pp. 242–246. doi: 10.1109/iCCECOME.2018.8658788.
- [31] R. Vithanage, C. Harrison, and A. DeSilva, 'Importance and Applications of Robotic and Autonomous Systems (RAS) in Railway Maintenance Sector: A Review', *Computers*, vol. 8, p. 56, Jul. 2019, doi: 10.3390/computers8030056.
- [32] 'IN2SMART2'. Accessed: Jan. 18, 2023. [Online]. Available: https://projects.shift2rail.org/s2r_ip3_n.aspx?p=IN2SMART2
- [33] M. E. Kahn and B. Roth, 'The near-minimum-time control of open-loop articulated kinematic chains', *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 93, no. 3, pp. 164–172, 1971, doi: 10.1115/1.3426492.
- [34] 'History of Robots and Robotics | Origins of Robots', Robotnik. Accessed: Nov. 16, 2022. [Online]. Available: <https://robotnik.eu/history-of-robots-and-robotics/>
- [35] 'The mobile robots at LAAS'. Accessed: Nov. 16, 2022. [Online]. Available: <https://homepages.laas.fr/matthieu/robots/h2bis.shtml>
- [36] 'The 5 Key Abilities of Mobile Manipulation Systems', Robotics Business Review. Accessed: Mar. 13, 2023. [Online]. Available: <https://www.roboticsbusinessreview.com/opinion/the-5-key-abilities-of-mobile-manipulation-systems/>

- [37] Philphos, *English: Southern portal of Sutton Tunnel, Cheshire*. 2017. Accessed: Mar. 20, 2023. [Online]. Available: https://commons.wikimedia.org/wiki/File:Sutton_Tunnel_Portal.jpg
- [38] J. Coussins, 'Rare glimpse inside forgotten railway tunnel being brought back', BirminghamLive. Accessed: Mar. 20, 2023. [Online]. Available: <https://www.birminghammail.co.uk/black-country/rare-glimpse-inside-forgotten-dudley-19333263>
- [39] C. Erridge, 'Signals explained', Network Rail. Accessed: Mar. 20, 2023. [Online]. Available: <https://www.networkrail.co.uk/stories/signals-explained/>
- [40] 'boundary markers', brucehunt.co.uk. Accessed: Mar. 20, 2023. [Online]. Available: <https://brucehunt.co.uk/bere%20ferrers%20and%20the%20bere%20openinsula/boundary%20markers.html>
- [41] 'Rail Location Cases and Cabinets LOC's | RSP'. Accessed: Mar. 20, 2023. [Online]. Available: <https://www.rsprail.co.uk/products/signalling-products/location-cases/>
- [42] *Reducing error and influencing behaviour*, 2. ed., Reprinted. in HSG, no. 48. Sudbury: HSE Books, 2007.
- [43] K. Itoh, H. B. Andersen, and M. Seki, 'Track maintenance train operators' attitudes to job, organisation and management, and their correlation with accident/incident rate', *Cogn Tech Work*, vol. 6, no. 2, pp. 63–78, May 2004, doi: 10.1007/s10111-003-0135-x.
- [44] S. Singh and R. Kumar, 'Evaluation of Human Error Probability of Disc Brake Unit Assembly and Wheel Set Maintenance of Railway Bogie', *Procedia Manufacturing*, vol. 3, pp. 3041–3048, Dec. 2015, doi: 10.1016/j.promfg.2015.07.849.
- [45] 'Work-related Ill Health in Railway Operatives'. Health and Safety Executive, 2018. [Online]. Available: <https://www.hse.gov.uk/statistics/adhoc-analysis/work-related-ill-health-railway-operatives.pdf>
- [46] 'Passenger rail performance | ORR Data Portal'. Accessed: Dec. 05, 2022. [Online]. Available: <https://dataportal.orr.gov.uk/statistics/performance/passenger-rail-performance/>
- [47] 'How Much Does it Cost to Integrate Robots?', SP Automation & Robotics. Accessed: Jan. 06, 2023. [Online]. Available: <https://sp-automation.co.uk/how-much-does-it-cost-to-integrate-robots/>
- [48] 'ABB 6640 Robots - Robots Done Right'. Accessed: Jan. 06, 2023. [Online]. Available: <https://robotsdoneright.com/Articles/ABB-6640.html>

- [49] R. Hu *et al.*, 'A Simple Framework for the Cost–Benefit Analysis of Single-Task Construction Robots Based on a Case Study of a Cable-Driven Facade Installation Robot', *Buildings*, vol. 11, no. 1, p. 8, Dec. 2020, doi: 10.3390/buildings11010008.
- [50] C. Brosque and M. Fischer, 'Safety, quality, schedule, and cost impacts of ten construction robots', *Constr Robot*, vol. 6, no. 2, pp. 163–186, 2022, doi: 10.1007/s41693-022-00072-5.
- [51] 'Rail safety | ORR Data Portal'. Accessed: Jan. 06, 2023. [Online]. Available: <https://dataportal.orr.gov.uk/statistics/health-and-safety/rail-safety/>
- [52] G. Chen, Q. Liang, W. Zhong, X. Gao, and F. Cui, 'Homography-based measurement of bridge vibration using UAV and DIC method', *Measurement*, vol. 170, p. 108683, Jan. 2021, doi: 10.1016/j.measurement.2020.108683.
- [53] K. Stypułkowski, P. Gołda, K. Lewczuk, and J. Tomaszewska, 'Monitoring System for Railway Infrastructure Elements Based on Thermal Imaging Analysis', *Sensors*, vol. 21, no. 11, Art. no. 11, Jan. 2021, doi: 10.3390/s21113819.
- [54] Railway-News, 'Railway Inspection Robot Takes On Maintenance Challenges', Railway-News. Accessed: Nov. 03, 2022. [Online]. Available: <https://railway-news.com/railway-inspection-robot-takes-on-maintenance-challenges/>
- [55] T. Hisa, M. Kanaya, M. Sakai, and K. Hamaoka, 'Rail and Contact Line Inspection Technology for Safe and Reliable Railway Traffic'.
- [56] V. Ganesh and M. Mm, 'Automotive Crack Detection for Railway Track Using Ultrasonic Sensorz', Accessed: Feb. 13, 2023. [Online]. Available: https://www.academia.edu/38604596/Automotive_Crack_Detection_for_Railway_Track_Using_Ultrasonic_Sensorz
- [57] Z. Li and Q. He, 'Predicting failure times of railcar wheels and trucks by using wayside detector signals', in *2014 IEEE International Conference on Mechatronics and Automation*, Aug. 2014, pp. 1113–1118. doi: 10.1109/ICMA.2014.6885854.
- [58] J. Rajamäki, M. Vippola, A. Nurmikolu, and T. Viitala, 'Limitations of eddy current inspection in railway rail evaluation', *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 232, no. 1, pp. 121–129, Jan. 2018, doi: 10.1177/0954409716657848.
- [59] 'A new electromagnetic acoustic transducer (EMAT) design for operation on rail - ScienceDirect'. Accessed: Feb. 13, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0963869514000413?via%3Dihub>

- [60] H. Wang, M. Silvast, V. Markine, and B. Wiljanen, 'Analysis of the Dynamic Wheel Loads in Railway Transition Zones Considering the Moisture Condition of the Ballast and Subballast', *Applied Sciences*, vol. 7, no. 12, Art. no. 12, Dec. 2017, doi: 10.3390/app7121208.
- [61] H. Wang, L. Chang, and V. Markine, 'Structural Health Monitoring of Railway Transition Zones Using Satellite Radar Data', *Sensors*, vol. 18, no. 2, Art. no. 2, Feb. 2018, doi: 10.3390/s18020413.
- [62] 'Trolleys & Vehicles Archives', Permaquip UK. Accessed: Oct. 28, 2022. [Online]. Available: <https://permaquip.co.uk/shop/product-category/trolleys-vehicles/>
- [63] I. Majstorović, M. Ahac, and S. Lakusic, *Track gauge monitoring scope optimization on small urban railway systems*. 2021. doi: 10.5592/CO/CETRA.2020.1207.
- [64] 'Track Geometry Trolley', Pandrol. Accessed: Oct. 28, 2022. [Online]. Available: <https://www.pandrol.com/product/track-geometry-trolley/>
- [65] 'AS 7502:2016 Road Rail vehicles', Rail Industry Safety and Standard Board, Jun. 2016.
- [66] A. Perry, 'Rail Industry Standard for Engineering Acceptance of On-Track Plant and Associated Equipment', 2014.
- [67] 'Unimog U 219 – U 535: Road-rail - Mercedes-Benz Trucks - Trucks you can trust'. Accessed: Oct. 28, 2022. [Online]. Available: https://www.mercedes-benz-trucks.com/en_GB/models/unimog-implement-carrier/applications/road-rail.html
- [68] 'Inspection & transport Road-Rail Vehicle Runner Wizard - V2R', Geismar. Accessed: Oct. 28, 2022. [Online]. Available: <https://geismar.com/products/inspection-transport-road-rail-vehicle-v2r-runner-wizard/?lang=en>
- [69] Y. B. Yang, Z.-L. Wang, K. Shi, H. Xu, and Y. T. Wu, 'State-of-the-Art of Vehicle-Based Methods for Detecting Various Properties of Highway Bridges and Railway Tracks', *International Journal of Structural Stability and Dynamics*, Aug. 2020, doi: 10.1142/S0219455420410047.
- [70] 'Our Solution', Machines with Vision. Accessed: Oct. 31, 2022. [Online]. Available: <https://www.machineswithvision.com/oursolution>
- [71] 'RILA® 3.0'. Accessed: Oct. 31, 2022. [Online]. Available: <https://www.fugro.com/about-fugro/our-expertise/innovations/rila>
- [72] E. Menendez, J. G. Victores, R. Montero, S. Martínez, and C. Balaguer, 'Tunnel structural inspection and assessment using an autonomous robotic system', *Automation in Construction*, vol. 87, pp. 117–126, Mar. 2018, doi: 10.1016/j.autcon.2017.12.001.

- [73] M. Yucel and N. Ozturk, 'Real-time monitoring of railroad track tension using a fiber Bragg grating-based strain sensor', *Instrumentation Science & Technology*, pp. 1–15, Dec. 2017, doi: 10.1080/10739149.2017.1415930.
- [74] M. M. Trivedi, K. C. Ng, N. Lassiter, and R. Capella, 'New generation of multirobot systems', in *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*, San Diego, CA, USA: IEEE, 1998, pp. 3342–3346. doi: 10.1109/ICSMC.1998.726519.
- [75] P. L. Mazzeo, M. Nitti, E. Stella, and A. Distanto, 'Visual recognition of fastening bolts for railroad maintenance', *Pattern Recognition Letters*, vol. 25, no. 6, pp. 669–677, Apr. 2004, doi: 10.1016/j.patrec.2004.01.008.
- [76] M. Singh, S. Singh, J. Jaiswal, and J. Hempshall, 'Autonomous Rail Track Inspection using Vision Based System', in *2006 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety*, Oct. 2006, pp. 56–59. doi: 10.1109/CIHSPS.2006.313313.
- [77] F. Marino, A. Distanto, P. L. Mazzeo, and E. Stella, 'A Real-Time Visual Inspection System for Railway Maintenance: Automatic Hexagonal-Headed Bolts Detection', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 418–428, May 2007, doi: 10.1109/TSMCC.2007.893278.
- [78] H. Zhang, J. Yang, W. Tao, and H. Zhao, 'Vision method of inspecting missing fastening components in high-speed railway', *Appl. Opt., AO*, vol. 50, no. 20, pp. 3658–3665, Jul. 2011, doi: 10.1364/AO.50.003658.
- [79] E. Resendiz, J. M. Hart, and N. Ahuja, 'Automated Visual Inspection of Railroad Tracks', *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 751–760, Jun. 2013, doi: 10.1109/TITS.2012.2236555.
- [80] Y. Li, H. Trinh, N. Haas, C. Otto, and S. Pankanti, 'Rail Component Detection, Optimization, and Assessment for Automatic Rail Track Inspection', *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 760–770, Apr. 2014, doi: 10.1109/TITS.2013.2287155.
- [81] L. Wang, B. Zhang, J. Wu, H. Xu, X. Chen, and W. Na, 'Computer vision system for detecting the loss of rail fastening nuts based on kernel two-dimensional principal component – two-dimensional principal component analysis and a support vector machine', *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 230, no. 8, pp. 1842–1850, Nov. 2016, doi: 10.1177/0954409715616426.
- [82] X. Gibert, V. M. Patel, and R. Chellappa, 'Robust Fastener Detection for Autonomous Visual Railway Track Inspection', in *2015 IEEE Winter Conference on Applications of Computer Vision*, Jan. 2015, pp. 694–701. doi: 10.1109/WACV.2015.98.

- [83] P. Ranganathan and E. Olson, 'Automated safety inspection of grade crossings', in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 2149–2154. doi: 10.1109/IROS.2010.5651812.
- [84] J. Madejski, 'Autonomous track geometry diagnostics system', *Journal of Materials Processing Technology*, vol. 157–158, pp. 194–202, Dec. 2004, doi: 10.1016/j.jmatprotec.2004.09.029.
- [85] Y. Santur, M. Karakose, and E. Akin, 'A new rail inspection method based on deep learning using laser cameras', in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, Malatya: IEEE, Sep. 2017, pp. 1–6. doi: 10.1109/IDAP.2017.8090245.
- [86] H. Rowshandel, G. L. Nicholson, C. L. Davis, and C. Roberts, 'An integrated robotic system for automatic detection and characterisation of rolling contact fatigue cracks in rails using an alternating current field measurement sensor', *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 227, no. 4, pp. 310–321, Jul. 2013, doi: 10.1177/0954409713486778.
- [87] 'Our Platform – RailPod, Inc.' Accessed: Dec. 25, 2020. [Online]. Available: https://rail-pod.com/?page_id=2527
- [88] 'AutoScan – Rail inspection by autonomous systems'. Accessed: Sep. 25, 2022. [Online]. Available: <https://cordis.europa.eu/project/id/720506>
- [89] 'Mobile robotic inspection for railway infrastructures', Loccioni. Accessed: Nov. 03, 2022. [Online]. Available: <https://www.loccioni.com/en/railway/>
- [90] S. Jung *et al.*, 'Toward Autonomous Bridge Inspection: A framework and experimental results', in *2019 16th International Conference on Ubiquitous Robots (UR)*, Jun. 2019, pp. 208–211. doi: 10.1109/URAI.2019.8768677.
- [91] H.-F. Wang, L. Zhai, H. Huang, L.-M. Guan, K.-N. Mu, and G. Wang, 'Measurement for cracks at the bottom of bridges based on tethered creeping unmanned aerial vehicle', *Automation in Construction*, vol. 119, p. 103330, Nov. 2020, doi: 10.1016/j.autcon.2020.103330.
- [92] S. Phillips and S. Narasimhan, 'Automating Data Collection for Robotic Bridge Inspections', *Journal of Bridge Engineering*, vol. 24, no. 8, p. 04019075, Aug. 2019, doi: 10.1061/(ASCE)BE.1943-5592.0001442.
- [93] D. Liu *et al.*, 'A Robotic System for Steel Bridge Maintenance: Research Challenges and System Design', *Proceedings of the 2008 Australasian Conference on Robotics and Automation, ACRA 2008*, Jan. 2008.
- [94] R. Wang and Y. Kawamura, 'A magnetic climbing robot for steel bridge inspection', in *Proceeding of the 11th World Congress on Intelligent Control and Automation*, Jun. 2014, pp. 3303–3308. doi: 10.1109/WCICA.2014.7053262.

- [95] J.-K. Oh, A.-Y. Lee, S. M. Oh, Y. Choi, B.-J. Yi, and H. W. Yang, 'Design and Control of Bridge Inspection Robot System', in *2007 International Conference on Mechatronics and Automation*, Aug. 2007, pp. 3634–3639. doi: 10.1109/ICMA.2007.4304150.
- [96] W. Zhang, Z. Zhang, D. Qi, and Y. Liu, 'Automatic Crack Detection and Classification Method for Subway Tunnel Safety Monitoring', *Sensors*, vol. 14, no. 10, Art. no. 10, Oct. 2014, doi: 10.3390/s141019307.
- [97] H. Huang, Q. Li, and D. Zhang, 'Deep learning based image recognition for crack and leakage defects of metro shield tunnel', *Tunnelling and Underground Space Technology*, vol. 77, pp. 166–176, Jul. 2018, doi: 10.1016/j.tust.2018.04.002.
- [98] H. Feng, Z. Jiang, F. Xie, P. Yang, J. Shi, and L. Chen, 'Automatic Fastener Classification and Defect Detection in Vision-Based Railway Inspection Systems', *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 4, pp. 877–888, Apr. 2014, doi: 10.1109/TIM.2013.2283741.
- [99] Q. Li and S. Ren, 'A Real-Time Visual Inspection System for Discrete Surface Defects of Rail Heads', *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 8, pp. 2189–2199, Aug. 2012, doi: 10.1109/TIM.2012.2184959.
- [100] M. David Jenkins, T. Buggy, and G. Morison, 'An imaging system for visual inspection and structural condition monitoring of railway tunnels', in *2017 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*, Jul. 2017, pp. 1–6. doi: 10.1109/EESMS.2017.8052679.
- [101] D. Stein, M. Spindler, and M. Lauer, 'Model-based rail detection in mobile laser scanning data', in *2016 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2016, pp. 654–661. doi: 10.1109/IVS.2016.7535457.
- [102] X. Zhou, C. Z. Fan, and S. K. Li, 'A Method for Strategic Migration from Simulation to Real Manipulator System', in *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, Dec. 2018, pp. 216–220. doi: 10.1109/ISCID.2018.00056.
- [103] D. Zhang, S. N. Lingamanaik, and H. Chung, 'Image-based 3D reconstruction for rail profile measurement', *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, p. 09544097221110322, Jun. 2022, doi: 10.1177/09544097221110322.
- [104] K. R. da Silva Santos, W. R. de Oliveira, E. Villani, and A. Dttmann, '3D scanning method for robotized inspection of industrial sealed parts', *Computers in Industry*, vol. 147, p. 103850, May 2023, doi: 10.1016/j.compind.2022.103850.
- [105] C. Popescu, B. Täljsten, T. Blanksvärd, and L. Elfgren, '3D reconstruction of existing concrete bridges using optical methods', *Structure and*

- Infrastructure Engineering*, vol. 15, no. 7, pp. 912–924, Jul. 2019, doi: 10.1080/15732479.2019.1594315.
- [106] J. Ye, E. Stewart, and C. Roberts, 'Use of a 3D model to improve the performance of laser-based railway track inspection', *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 233, no. 3, pp. 337–355, Mar. 2019, doi: 10.1177/0954409718795714.
 - [107] 'Track geometry - inspections and minimum actions NR-L2-TRK-001-MOD11', Network Rail, UK, Standard NR-L2-TRK-001-MOD11.
 - [108] C. D. Martland, 'Analysis of the potential impacts of automation and robotics on locomotive rebuilding', *IEEE Transactions on Engineering Management*, vol. EM-34, no. 2, pp. 92–100, May 1987, doi: 10.1109/TEM.1987.6499032.
 - [109] W. Wiercienski and A. R. Leek, 'Feasibility of robotic cleaning of the undersides of Toronto subway cars', in *IEEE 39th Vehicular Technology Conference*, May 1989, pp. 884–887 vol.2. doi: 10.1109/VETEC.1989.40178.
 - [110] H. Yaguchi, 'Robot introduction to cleaning work in the East Japan Railway Company', *Advanced Robotics*, vol. 10, no. 4, pp. 403–414, Jan. 1995, doi: 10.1163/156855396X00066.
 - [111] H. Xu, K. Xue, S. Yu, X. Gao, C. Qing, and R. Wei, 'Control for an Innovative Robotics Platform of Rinsing System for Vehicles of Urban Mass Transit', in *2007 IEEE International Conference on Mechatronics*, May 2007, pp. 1–6. doi: 10.1109/ICMECH.2007.4280038.
 - [112] T. Tomiyama, L. R. García, A. Kršlin, and G. Taykaldiranian, 'Systems and Conceptual Design of a Train Cab Front Cleaning Robot', *Procedia CIRP*, vol. 59, pp. 61–66, Jan. 2017, doi: 10.1016/j.procir.2016.09.031.
 - [113] L. R. García, 'Conceptual Design of Cab Front Robotic Cleaner', Cranfield University, Bedford, UK, 2016.
 - [114] 'Robot in the maintenance of cars'. Accessed: Nov. 04, 2022. [Online]. Available: <http://www.vale.com/brasil/EN/aboutvale/news/Pages/robos-manutencao-vagoes.aspx>
 - [115] 'Railway Wheel Set Maintenance, Production', NSH USA Corporation. Accessed: Nov. 04, 2022. [Online]. Available: <https://nsh-usa.com/>
 - [116] J. Jaiswal, S. F. Smith, R. Cox, and E. Kabo, 'Repair of Discrete Rail Head Defects – A Novel Technology', Apr. 2018, doi: 10.5281/ZENODO.1473575.
 - [117] M. F. M. Campos, G. A. S. Pereira, S. R. C. Vale, A. Q. Bracarense, G. A. Pinheiro, and M. P. Oliveira, 'A mobile manipulator for installation and removal of aircraft warning spheres on aerial power transmission lines', in

Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), May 2002, pp. 3559–3564 vol.4. doi: 10.1109/ROBOT.2002.1014261.

- [118] P. Debenest *et al.*, ‘Sensor-Arm — robotic manipulator for preventive maintenance and inspection of high-voltage transmission lines’, in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2008, pp. 1737–1744. doi: 10.1109/IROS.2008.4650964.
- [119] J. -F. Allan, G. Lambert, S. Lavoie, and S. Reiher, ‘Development of a Mobile Robotic Platform for the Underground Distribution Lines’, in *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Jul. 2008, pp. 406–411. doi: 10.1109/AIM.2008.4601695.
- [120] J. . -F. Allan, S. Reiher, G. Lambert, and S. Lavoie, ‘Field tests of a robot system prototype for the underground distribution lines’, in *IEEE PES T&D 2010*, Apr. 2010, pp. 1–5. doi: 10.1109/TDC.2010.5484592.
- [121] N. Pouliot and S. Montambault, ‘LineScout Technology: From inspection to robotic maintenance on live transmission power lines’, in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 1034–1040. doi: 10.1109/ROBOT.2009.5152291.
- [122] Jin Jian, Zhang Guoxian, and Zhang Tingyu, ‘Design of a mobile robot for the innovation in power line inspection and maintenance’, in *2009 ASME/IFTOMM International Conference on Reconfigurable Mechanisms and Robots*, Jun. 2009, pp. 444–449.
- [123] Guo Rui, Zhang Feng, Cao Lei, and Yong Jun, ‘A mobile robot for inspection of overhead transmission lines’, in *Proceedings of the 2014 3rd International Conference on Applied Robotics for the Power Industry*, Oct. 2014, pp. 1–3. doi: 10.1109/CARPI.2014.7030054.
- [124] L. Cao *et al.*, ‘Screwing manipulation on power transmission lines with a robotic system’, in *2014 IEEE International Conference on Mechatronics and Automation*, Aug. 2014, pp. 1294–1299. doi: 10.1109/ICMA.2014.6885886.
- [125] J. Feng and W. Zhang, ‘Autonomous Live-Line Maintenance Robot for a 10 kV Overhead Line’, *IEEE Access*, vol. 9, pp. 61819–61831, 2021, doi: 10.1109/ACCESS.2021.3074677.
- [126] H. Wang, F. Zhang, Y. Jiang, G. Liu, and X. Peng, ‘Development of an inspection robot for 500 kV EHV power transmission lines’, in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 5107–5112. doi: 10.1109/IROS.2010.5651966.
- [127] M. Perez-Jimenez, M. A. Montes-Grova, P. Ramon-Soria, B. C. Arrue, and A. Ollero, ‘POSITRON: lightweight active positioning compliant joints robotic arm in power lines inspection’, in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, Sep. 2020, pp. 729–736. doi: 10.1109/ICUAS48674.2020.9214022.

- [128] A. Suarez, R. Salmoral, P. J. Zarco-Periñan, and A. Ollero, 'Experimental Evaluation of Aerial Manipulation Robot in Contact With 15 kV Power Line: Shielded and Long Reach Configurations', *IEEE Access*, vol. 9, pp. 94573–94585, 2021, doi: 10.1109/ACCESS.2021.3093856.
- [129] H. Wang, J. Li, Y. Zhou, M. Fu, and S. Yang, 'Research on the Technology of Indoor and Outdoor Integration Robot Inspection in Substation', in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Mar. 2019, pp. 2366–2369. doi: 10.1109/ITNEC.2019.8729355.
- [130] X. Zhang *et al.*, 'Development of the Maintenance Robot with Electrification Used in Substation', in *2017 Second International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, Dec. 2017, pp. 30–33. doi: 10.1109/ICMCCE.2017.34.
- [131] S. Liu, X. Zhang, R. Huang, X. Dong, J. Li, and Q. Chen, 'Optimization Design of the Maintenance Robot with Charged Used in Substation', in *2018 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, Sep. 2018, pp. 141–144. doi: 10.1109/ICMCCE.2018.00036.
- [132] A. Hibino *et al.*, 'Development of a semi-autonomous remote maintenance robot', in *2009 International Symposium on Micro-NanoMechatronics and Human Science*, Nov. 2009, pp. 396–401. doi: 10.1109/MHS.2009.5352006.
- [133] M. Di Castro, M. Ferre, and A. Masi, 'CERNTAURO: A Modular Architecture for Robotic Inspection and Telemanipulation in Harsh and Semi-Structured Environments', *IEEE Access*, vol. 6, pp. 37506–37522, 2018, doi: 10.1109/ACCESS.2018.2849572.
- [134] A. Dehne, N. Moller, and T. Hermes, 'MARWIN: a mobile autonomous robot for maintenance and inspection in a 4d environment', in *2017 International Conference on Research and Education in Mechatronics (REM)*, Sep. 2017, pp. 1–5. doi: 10.1109/REM.2017.8075228.
- [135] W. Mingfei, C. Wusheng, and F. Jiaxin, 'Development and implementation of a multifunctional inspection and rescue robot system for fusion facility', in *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Mar. 2017, pp. 204–210. doi: 10.1109/IAEAC.2017.8054007.
- [136] D. Morra *et al.*, 'Visual control through narrow passages for an omnidirectional wheeled robot', in *2022 30th Mediterranean Conference on Control and Automation (MED)*, Jul. 2022, pp. 551–556. doi: 10.1109/MED54222.2022.9837221.
- [137] X. Zhang, J. Zhang, J. Yuan, and M. Li, 'Development of an underwater robot for nuclear reactor vessel', in *2013 IEEE International Conference on*

- Robotics and Biomimetics (ROBIO)*, Dec. 2013, pp. 1699–1703. doi: 10.1109/ROBIO.2013.6739712.
- [138] P. Chermprayong, K. Zhang, F. Xiao, and M. Kovac, ‘An Integrated Delta Manipulator for Aerial Repair: A New Aerial Robotic System’, *IEEE Robotics & Automation Magazine*, vol. 26, no. 1, pp. 54–66, Mar. 2019, doi: 10.1109/MRA.2018.2888911.
 - [139] P. Junjie, ‘Design and trajectory planning of blocking plate robot in steam generator’, in *2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, Jul. 2020, pp. 108–111. doi: 10.1109/ICPICS50287.2020.9202353.
 - [140] Y. Qiu, B. Guo, L. Xue, B. Liang, and C. Li, ‘On-Orbit Servicing to GEO satellite using Dual Arm Free-flying Space Robot’, in *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2009, pp. 2451–2456. doi: 10.1109/ROBIO.2009.5420465.
 - [141] X. Du, Y. He, L. Chen, and S. Gao, ‘Pose estimation of large non-cooperative spacecraft based on extended PnP model’, in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2016, pp. 413–418. doi: 10.1109/ROBIO.2016.7866357.
 - [142] E. A. S. Hanlon *et al.*, ‘AMODS: Autonomous mobile on-orbit diagnostic system’, in *2016 IEEE Aerospace Conference*, Mar. 2016, pp. 1–10. doi: 10.1109/AERO.2016.7500512.
 - [143] L. Shi, H. Jayakody, J. Katupitiya, and X. Jin, ‘Coordinated Control of a Dual-Arm Space Robot: Novel Models and Simulations for Robotic Control Methods’, *IEEE Robotics & Automation Magazine*, vol. 25, no. 4, pp. 86–95, Dec. 2018, doi: 10.1109/MRA.2018.2864717.
 - [144] G. E. Dongming, S. Guanghui, Z. Yuanjie, and S. Jixin, ‘Impedance control of multi-arm space robot for the capture of non-cooperative targets’, *Journal of Systems Engineering and Electronics*, vol. 31, no. 5, pp. 1051–1061, Oct. 2020, doi: 10.23919/JSEE.2020.000079.
 - [145] J. García, D. Gonzalez, A. Rodríguez, B. Santamaria, J. Estremera, and M. Armendia, ‘Application of Impedance Control in Robotic Manipulators for Spacecraft On-orbit Servicing’, in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2019, pp. 836–842. doi: 10.1109/ETFA.2019.8869069.
 - [146] H. Wang, B. Zhang, T. Zhang, and A. Jakacky, ‘Tele-operating a Collaborative Robot for Space Repairs with Virtual Reality’, in *2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, Aug. 2019, pp. 175–180. doi: 10.1109/CYBER46603.2019.9066493.
 - [147] H. Wang, R. Mecham, and B. Zhang, ‘A Method Targeting Repair in Space: Tele-operating a Collaborative Robot with Virtual Reality’, in *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation,*

- Control, and Intelligent Systems (CYBER)*, Jul. 2017, pp. 1068–1071. doi: 10.1109/CYBER.2017.8446482.
- [148] D. A. Chamberlain and E. Gambao, ‘A robotic system for concrete repair preparation’, *IEEE Robotics & Automation Magazine*, vol. 9, no. 1, pp. 36–44, Mar. 2002, doi: 10.1109/100.993153.
- [149] S. J. Lorenc, B. E. Handlon, and L. E. Bernold, ‘Development of a robotic bridge maintenance system’, *Automation in Construction*, vol. 9, no. 3, pp. 251–258, May 2000, doi: 10.1016/S0926-5805(99)00040-0.
- [150] H. M. La *et al.*, ‘Autonomous robotic system for high-efficiency non-destructive bridge deck inspection and evaluation’, in *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug. 2013, pp. 1053–1058. doi: 10.1109/CoASE.2013.6653886.
- [151] F. Liu, M. Trkov, J. Yi, and N. Gucunski, ‘Modeling of pure percussive drilling for autonomous robotic bridge decks rehabilitation’, in *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug. 2013, pp. 1063–1068. doi: 10.1109/CoASE.2013.6654053.
- [152] L. Van Nguyen, S. Gibb, H. X. Pham, and H. M. La, ‘A Mobile Robot for Automated Civil Infrastructure Inspection and Evaluation’, in *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Aug. 2018, pp. 1–6. doi: 10.1109/SSRR.2018.8468642.
- [153] M. Kahouadji, O. Lakhal, X. Yang, A. Belarouci, and R. Merzouki, ‘System of Robotic Systems for Crack Predictive Maintenance’, in *2021 16th International Conference of System of Systems Engineering (SoSE)*, Jun. 2021, pp. 197–202. doi: 10.1109/SOSE52739.2021.9497490.
- [154] L. Orr, B. Stephens, B. B. Kocer, and M. Kovac, ‘A High Payload Aerial Platform for Infrastructure Repair and Manufacturing’, in *2021 Aerial Robotic Systems Physically Interacting with the Environment (AIRPHARO)*, Oct. 2021, pp. 1–6. doi: 10.1109/AIRPHARO52252.2021.9571052.
- [155] D. Charlton, J. Taylor, S. Vougioukas, and Z. Rutledge, ‘Innovations for a Shrinking Agricultural Workforce’, May 2019.
- [156] S. Shademani, P. Zarafshan, M. Khashechi, M. H. Kianmehr, and S. M. Hashemy, ‘Design and Analysis of a Dredger Robot for Covered Irrigation Canals’, in *2017 5th RSI International Conference on Robotics and Mechatronics (ICRoM)*, Oct. 2017, pp. 162–167. doi: 10.1109/ICRoM.2017.8466207.
- [157] R. Majdoubi and L. Masmoudi, ‘Eco-design of a mobile agriculture robot based on classical approach and FEM criteria’, in *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, Feb. 2021, pp. 978–982. doi: 10.1109/ICCCIS51004.2021.9397234.

- [158] I. Hrabar, J. Goričanec, and Z. Kovačić, 'Towards Autonomous Navigation of a Mobile Robot in a Steep Slope Vineyard', in *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, Oct. 2021, pp. 1119–1124. doi: 10.23919/MIPRO52101.2021.9596997.
- [159] F. Qingchun, Z. Wengang, Q. Quan, J. Kai, and G. Rui, 'Study on strawberry robotic harvesting system', in *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, May 2012, pp. 320–324. doi: 10.1109/CSAE.2012.6272606.
- [160] D. Khort, A. Kutyrev, R. Filippov, N. Kiktev, and D. Komarchuk, 'Robotized Platform for Picking of Strawberry Berries', in *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Oct. 2019, pp. 869–872. doi: 10.1109/PICST47496.2019.9061448.
- [161] S. Vineet, D. Deshmukh, D. K. Pratihari, A. K. Deb, H. Ray, and N. Bhattacharyya, 'Dynamic Analysis of Tracked Mobile Manipulator Used in Agriculture', in *2021 IEEE 18th India Council International Conference (INDICON)*, Dec. 2021, pp. 1–6. doi: 10.1109/INDICON52576.2021.9691678.
- [162] J. Martin *et al.*, 'A Generic ROS-Based Control Architecture for Pest Inspection and Treatment in Greenhouses Using a Mobile Manipulator', *IEEE Access*, vol. 9, pp. 94981–94995, 2021, doi: 10.1109/ACCESS.2021.3093978.
- [163] E. Helms, R. D. Schraft, and M. Hagele, 'rob@work: Robot assistant in industrial environments', in *11th IEEE International Workshop on Robot and Human Interactive Communication Proceedings*, Sep. 2002, pp. 399–404. doi: 10.1109/ROMAN.2002.1045655.
- [164] M. Hvilshøj, S. Bøgh, O. Madsen, and M. Kristiansen, 'The mobile robot "Little Helper": Concepts, ideas and working principles', in *2009 IEEE Conference on Emerging Technologies & Factory Automation*, Sep. 2009, pp. 1–4. doi: 10.1109/ETFA.2009.5347251.
- [165] 'Mobile Manipulators Robots - Robot Manipulator | Robotnik®', Robotnik. Accessed: Nov. 21, 2022. [Online]. Available: <https://robotnik.eu/products/mobile-manipulators/>
- [166] nate archer I. designboom, 'willow garage PR2', designboom | architecture & design magazine. Accessed: Nov. 21, 2022. [Online]. Available: <https://www.designboom.com/technology/willow-garage-pr2/>
- [167] D. S. Sathia Raj, S. TV, S. Shankar S, V. P. Yashvanth, S. K. Perepu, and S. Mohan, 'An Autonomous Drone with Robotic End Effector for Maintenance Operation of 5G Mobile Towers', in *2020 3rd International Conference on Control and Robots (ICCR)*, Dec. 2020, pp. 11–16. doi: 10.1109/ICCR51572.2020.9344140.

- [168] C. Ding, L. Lu, C. Wang, and C. Ding, 'Design, Sensing, and Control of a Novel UAV Platform for Aerial Drilling and Screwing', *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3176–3183, Apr. 2021, doi: 10.1109/LRA.2021.3062305.
- [169] A. Ollero *et al.*, 'The AEROARMS Project: Aerial Robots with Advanced Manipulation Capabilities for Inspection and Maintenance', *IEEE Robotics & Automation Magazine*, vol. 25, no. 4, pp. 12–23, Dec. 2018, doi: 10.1109/MRA.2018.2852789.
- [170] H. Kim, H. Seo, C. Y. Son, H. Lee, S. Kim, and H. J. Kim, 'Cooperation in the Air: A Learning-Based Approach for the Efficient Motion Planning of Aerial Manipulators', *IEEE Robotics & Automation Magazine*, vol. 25, no. 4, pp. 76–85, Dec. 2018, doi: 10.1109/MRA.2018.2866766.
- [171] N. Staub, D. Bicego, Q. Sablé, V. Arellano, S. Mishra, and A. Franchi, 'Towards a Flying Assistant Paradigm: the OTHex', in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 6997–7002. doi: 10.1109/ICRA.2018.8460877.
- [172] J. Hong *et al.*, 'Design of Window-Cleaning Robotic Manipulator With Compliant Adaptation Capability', *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 4, pp. 1878–1885, Aug. 2020, doi: 10.1109/TMECH.2020.2991670.
- [173] J. Liang, Y. Wang, R. Wang, J. Li, H. Wang, and S. Liu, 'A Patrol Robot with Automatic Capture and Storage of Files', in *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Dec. 2020, pp. 1651–1655. doi: 10.1109/ITAIC49862.2020.9338850.
- [174] Z. Jiang, Z. Li, C. Li, D. Yang, and H. Liu, 'Design and Preliminary Ground Experiment for Robotic Assembly of a Modular Space Telescope', *IEEE Access*, vol. 7, pp. 160870–160878, 2019, doi: 10.1109/ACCESS.2019.2950666.
- [175] S. Teshigawara and H. Harry Asada, 'A Mobile Extendable Robot Arm: Singularity Analysis and Design', in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 5131–5138. doi: 10.1109/IROS40897.2019.8967768.
- [176] A. K. Bouloubasis and G. T. McKee, 'Cooperative transport of extended payloads', in *ICAR '05. Proceedings., 12th International Conference on Advanced Robotics, 2005.*, Jul. 2005, pp. 882–887. doi: 10.1109/ICAR.2005.1507511.
- [177] Z. Liu, P. Li, J. Liu, P. Li, E. Guan, and Y. -Z. Zhao, 'THG: A Multifunctional Wall-climbing Robot for the Large Penstock', in *2021 27th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, Nov. 2021, pp. 693–698. doi: 10.1109/M2VIP49856.2021.9665127.

- [178] Y. F. Yeung and K. Youcef-Toumi, 'An In-Pipe Manipulator for Contamination-Less Rehabilitation of Water Distribution Pipes', in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Jan. 2021, pp. 6390–6397. doi: 10.1109/IROS45743.2020.9341563.
- [179] J. Liu *et al.*, 'Garbage Collection and Sorting with a Mobile Manipulator using Deep Learning and Whole-Body Control', in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, Jul. 2021, pp. 408–414. doi: 10.1109/HUMANOIDS47582.2021.9555800.
- [180] A. Ibarra, A. Córdor, P. Martínez, and E. Tipán, 'Control reengineering used for rehabilitation of Andros Remotec bomb disposal robot', in *2020 IEEE ANDESCON*, Oct. 2020, pp. 1–6. doi: 10.1109/ANDESCON50619.2020.9272161.
- [181] N. M. Limparis and D. L. Akin, 'Design of a distributed control architecture for the SAMURAI deep submergence manipulator', in *2012 IEEE/OES Autonomous Underwater Vehicles (AUV)*, Sep. 2012, pp. 1–8. doi: 10.1109/AUV.2012.6380729.
- [182] J. E. Manley, S. Halpin, N. Radford, and M. Ondler, 'Aquanaut: A New Tool for Subsea Inspection and Intervention', in *OCEANS 2018 MTS/IEEE Charleston*, Oct. 2018, pp. 1–4. doi: 10.1109/OCEANS.2018.8604508.
- [183] E. Topini, A. Bucci, J. Gelli, A. Topini, A. Ridolfi, and B. Allotta, 'Advanced Underwater Manipulation Systems: an Overview of the SUONO Project', in *OCEANS 2021: San Diego – Porto*, Sep. 2021, pp. 1–7. doi: 10.23919/OCEANS44145.2021.9706110.
- [184] 'Industrial Robot Cost Declines Should Trigger Tipping Points in Demand', ARK Invest. Accessed: May 24, 2022. [Online]. Available: <https://ark-invest.com/articles/analyst-research/industrial-robot-cost-declines/>
- [185] F. Panella, N. Roecklinger, L. Vojnovic, Y. Loo, and J. Boehm, 'COST-BENEFIT ANALYSIS OF RAIL TUNNEL INSPECTION FOR PHOTOGRAMMETRY AND LASER SCANNING', *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLIII-B2-2020, pp. 1137–1144, Aug. 2020, doi: 10.5194/isprs-archives-XLIII-B2-2020-1137-2020.
- [186] A. V. Reyes-Acosta, I. Lopez-Juarez, R. Osorio-Comparan, and G. Lefranc, '3D pipe reconstruction employing video information from mobile robots', *Applied Soft Computing*, vol. 75, pp. 562–574, Feb. 2019, doi: 10.1016/j.asoc.2018.11.016.
- [187] 'Intelligent Innovative Smart Maintenance of Assets by integRATED Technologies 2 | IN2SMART2 Project | Fact Sheet | H2020 | CORDIS | European Commission'. Accessed: Feb. 23, 2022. [Online]. Available: <https://cordis.europa.eu/project/id/881574>
- [188] 'ROS Wiki'. Accessed: Jan. 18, 2023. [Online]. Available: <http://wiki.ros.org/kinetic>

- [189] 'Robot Operating System', *Wikipedia*. Sep. 20, 2022. Accessed: Nov. 23, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Robot_Operating_System&oldid=1111385933
- [190] A. Mahtani, L. Sanchez, E. Fernandez, and A. Martinez, *Effective Robotics Programming with ROS - Third Edition*. Packt Publishing, 2016.
- [191] 'MoveIt Motion Planning Framework'. Accessed: Jan. 25, 2023. [Online]. Available: <https://moveit.ros.org/>
- [192] 'rviz - ROS Wiki'. Accessed: Jan. 18, 2023. [Online]. Available: <http://wiki.ros.org/rviz>
- [193] 'Gazebo'. Accessed: Aug. 04, 2020. [Online]. Available: <http://gazebosim.org/>
- [194] 'Simulator for industrial robots and offline programming - RoboDK'. Accessed: Jan. 25, 2023. [Online]. Available: <https://robodk.com/>
- [195] 'RobotStudio | ABB Robotics', Robotics. Accessed: Jan. 25, 2023. [Online]. Available: <https://new.abb.com/products/robotics/robotstudio>
- [196] 'Webots: robot simulator'. Accessed: Jan. 25, 2023. [Online]. Available: <https://www.cyberbotics.com/>
- [197] 'Robot simulator CoppeliaSim: create, compose, simulate, any robot - Coppelia Robotics'. Accessed: Jan. 25, 2023. [Online]. Available: <https://www.coppeliarobotics.com/>
- [198] 'Unity'. Accessed: Aug. 04, 2020. [Online]. Available: <https://unity.com/products/unity-simulation>
- [199] B. S. Blanchard, *System engineering management*. Hoboken, N.J.: Wiley, 2013.
- [200] A. Hijazi, D. Lefebvre, and J. Brethé, 'Characterization of repeatability of XY-Theta platform held by robotic manipulator arms using a camera', *2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2014, doi: 10.5220/0005056104210427.
- [201] R. P. Paul, *Robot Manipulators: Mathematics, Programming, and Control: the Computer Control of Robot Manipulators*. Richard Paul, 1981.
- [202] A. Solyman, 'Computing the Forward Kinematics of 6DOF Robotic Arm', Jan. 2019.
- [203] T. Foote, 'tf: The transform library', in *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, Woburn, MA, USA: IEEE, Apr. 2013, pp. 1–6. doi: 10.1109/TePRA.2013.6556373.
- [204] A. I. Comport, E. Marchand, M. Pressigout, and F. Chaumette, 'Real-time markerless tracking for augmented reality: the virtual visual servoing

- framework', *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 615–628, Jul. 2006, doi: 10.1109/TVCG.2006.78.
- [205] S. Liu, L. Acosta-Gamboa, X. Huang, and A. Lorence, 'Novel Low Cost 3D Surface Model Reconstruction System for Plant Phenotyping', *Journal of Imaging*, vol. 3, p. 39, Sep. 2017, doi: 10.3390/jimaging3030039.
- [206] R. Szeliski and S. B. Kang, 'Recovering 3D Shape and Motion from Image Streams Using Nonlinear Least Squares', *Journal of Visual Communication and Image Representation*, vol. 5, no. 1, pp. 10–28, Mar. 1994, doi: 10.1006/jvci.1994.1002.
- [207] 'Reconstructing Rome'. Accessed: Sep. 07, 2021. [Online]. Available: <https://www.computer.org/csdl/magazine/co/2010/06/mco2010060040/13rRUy3xYe2>
- [208] Y. Xue, S. Zhang, M. Zhou, and H. Zhu, 'Novel SfM-DLT method for metro tunnel 3D reconstruction and Visualization', *Underground Space*, vol. 6, no. 2, pp. 134–141, Apr. 2021, doi: 10.1016/j.undsp.2020.01.002.
- [209] A. Sabato, C. H. Beale, and C. Niezrecki, 'A novel optical investigation technique for railroad track inspection and assessment', in *Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, and Civil Infrastructure 2017*, International Society for Optics and Photonics, Apr. 2017, p. 101692C. doi: 10.1117/12.2257831.
- [210] W. Qiu, L. Jian, Y. Cheng, and H. Bai, 'Three-Dimensional Reconstruction of Tunnel Face Based on Multiple Images', *Advances in Civil Engineering*, vol. 2021, p. e8837309, Apr. 2021, doi: 10.1155/2021/8837309.
- [211] Y. Liu, J. Qiao, T. Han, L. Li, and T. Xu, 'A 3D Image Reconstruction Model for Long Tunnel Geological Estimation', *Journal of Advanced Transportation*, vol. 2020, p. e8846955, Dec. 2020, doi: 10.1155/2020/8846955.
- [212] C.-S. Chen and W.-Y. Chang, 'Pose estimation for generalized imaging device via solving non-perspective N point problem', in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, May 2002, pp. 2931–2937 vol.3. doi: 10.1109/ROBOT.2002.1013677.
- [213] B. Přibyl, P. Zemčík, and M. Čadík, 'Absolute Pose Estimation from Line Correspondences using Direct Linear Transformation', *Computer Vision and Image Understanding*, vol. 161, pp. 130–144, Aug. 2017, doi: 10.1016/j.cviu.2017.05.002.
- [214] 'Theia Vision Library'. Accessed: Jan. 18, 2023. [Online]. Available: <http://www.theia-sfm.org/sfm.html>
- [215] M. L. A. Lourakis and A. A. Argyros, 'Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment?', in

Tenth IEEE International Conference on Computer Vision (ICCV'05)
Volume 1, Oct. 2005, pp. 1526-1531 Vol. 2. doi: 10.1109/ICCV.2005.128.

- [216] 'sw_urdf_exporter - ROS Wiki'. Accessed: Jan. 31, 2023. [Online]. Available: http://wiki.ros.org/sw_urdf_exporter
- [217] Y. Song, Z. Wang, Z. Liu, and R. Wang, 'A spatial coupling model to study dynamic performance of pantograph-catenary with vehicle-track excitation', *Mechanical Systems and Signal Processing*, vol. 151, p. 107336, Apr. 2021, doi: 10.1016/j.ymssp.2020.107336.
- [218] A. Azzoug and S. Kaewunruen, 'RideComfort: A Development of Crowdsourcing Smartphones in Measuring Train Ride Quality', *Frontiers in Built Environment*, vol. 3, 2017, Accessed: Mar. 28, 2023. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fbuil.2017.00003>
- [219] Y.-W. Lin, C.-C. Hsieh, W.-H. Huang, S.-L. Hsieh, and W.-H. Hung, 'Railway Track Fasteners Fault Detection using Deep Learning', in *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, Yunlin, Taiwan: IEEE, Oct. 2019, pp. 187–190. doi: 10.1109/ECICE47484.2019.8942769.
- [220] F. Balouchi, A. Bevan, and R. Formston, 'Development of railway track condition monitoring from multi-train in-service vehicles', *Vehicle System Dynamics*, vol. 59, no. 9, pp. 1397–1417, Sep. 2021, doi: 10.1080/00423114.2020.1755045.
- [221] 'Maintenance Arc Welding of Rails, Switches and Crossings NR-L2-TRK-0132', Network Rail, Standard NR-L2-TRK-0132, 2023.
- [222] 'gazebo_ros_pkgs - ROS Wiki'. Accessed: Jan. 26, 2023. [Online]. Available: http://wiki.ros.org/gazebo_ros_pkgs
- [223] V. Tinoco, B. Malheiro, and M. F. Silva, 'Design, Modeling, and Simulation of a Wing Sail Land Yacht', *Applied Sciences*, vol. 11, no. 6, p. 2760, Mar. 2021, doi: 10.3390/app11062760.
- [224] B. Thacker, S.W.Doebling, F. Hemez, M. Anderson, J. E. Pepin, and E. Rodriguez, 'Concepts of Model Verification and Validation', Oct. 2004, doi: 10.2172/835920.

APPENDICES

Appendix A Softwares and Tools

A.1 ROS

ROS is a platform generally acknowledged and applied in the robotics world. Its primary purpose is to make the significant elements of a robotics system simple to design and distribute so that they may be used on other robots with minimum adjustments. ROS was created in 2007 by the Stanford Artificial Intelligence Laboratory (SAIL) to help with the Stanford AI Robot project. Willow Garage has maintained development of ROS since 2008.

Several educational institutions are working on ROS projects, sharing their code, and adding material. ROS supports most of the sensors and actuators used in robotics. Additionally, some businesses employ ROS and open-source hardware to produce robots that are less expensive and simpler to use. The hardware abstraction, low-level device control with ROS control, implementations of widely used functionality and libraries, message transfer between processes, and package management offered by ROS are only a few of the services it offers. It uses a graphic design with a centralized topology, where processing occurs in nodes that may send and receive messages to interact with other nodes in the graph network as shown in Figure A.1.

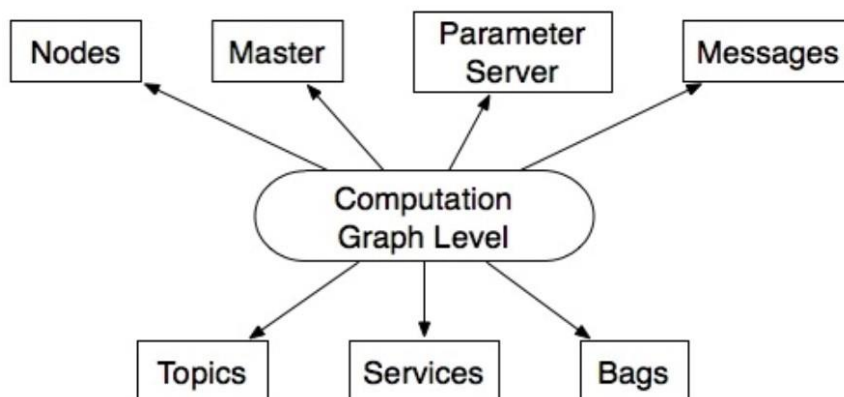


Figure A.1 Computation level architecture of ROS [190]

Nodes: In ROS, all the data transfer happens through a network of nodes. Node is a process or a combination of the process where all the computation takes place. In a ROS network, there can be single or multiple nodes.

Master: Master is responsible for maintaining the network and communication between nodes by registering names and IDs. Master allows the nodes from another computer in a distributed network to communicate.

Parameter server: The parameter server helps store the parameter's value through a key configured easily during computation.

Topic: The topic is a protocol to create a data stream by the nodes. Topic can be a publisher or subscriber. In ROS, multiple topics can subscribe or publish data simultaneously.

Message: Message is the data published or subscribed by the node through a topic. Each topic has a message publishing the data. There are many standard message types, or, based on the requirement, custom message types can be created.

Services: Services are another way of passing information between nodes. A vital difference between a topic and a service is the feedback system. The topic has one-way communication, while service has two-way communication with a feedback system.

Bags: Bags are a format for storing data in ROS. This enables storing sensor data and robot information for future analysis and reuses with a different algorithm.

A.2 MoveIt!

Figure A.3 shows the architecture of MoveIt! with all the communication with ROS and the user interface. The core of the MoveIt! is the `move_group` node which incorporates all the required components of the robot and creates various actions and services based on the user demand.

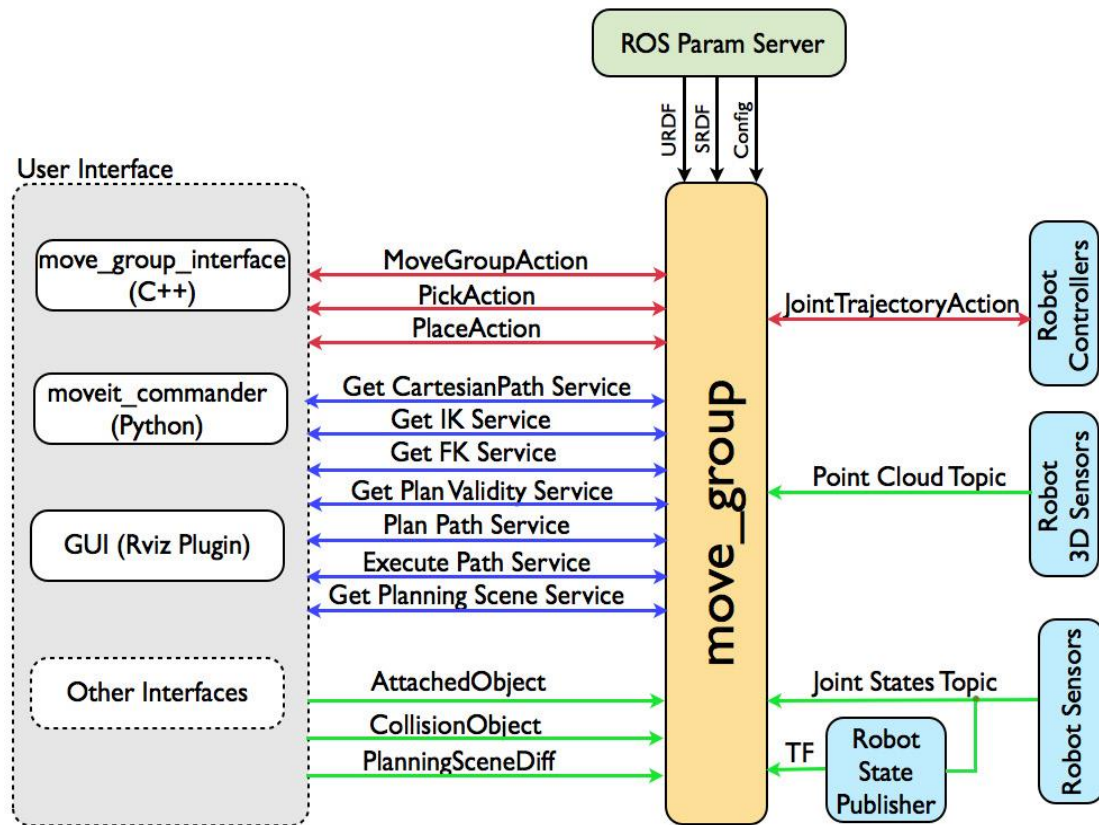


Figure A.2 MoveIt! architecture [192]

At first, `move_group` collects robot kinematics from the ROS param server and then matches those links and joint information with the robot controller. It also collects information from the robot sensor position and transforms the sensor and joints using the `ros_tf` library. These transformations will be used later to solve Inverse Kinematics (IK) and Forward Kinematics (FK) problems.

FK is the process of finding the pose of the end-effector using the joints and link information. It starts from the base joint and finishes at the end-effector. It can be considered a parent-to-child method. So, in forward kinematics, joint angles and links are the input, and the end-effector pose is the output. On the other hand, IK is the opposite. Here, only the pose of the end-effector is known. Then using some constraints, joint angles are calculated. To solve the FK and IK, `move_group` considers 2 types of workspaces: joint space and cartesian space.

Joint space is a vector made from the translation and rotation values of each joint of the robot arm. Motion planning in joint space using joint angles is easier and faster to calculate. When the manipulator repeatedly does some tasks at the predefined positions, it is better to use the joint goal method. But any changes in the end-effector position will require manual adjustment, and it does not consider dynamic obstacles. The pose goal requires the position and orientation of the end-effector in a cartesian workspace. With IK, it will then calculate the joint angles. The most significant advantage of the pose goal is that it can cope with the position change of the end-effector and are suitable for many real-life applications such as welding, picking an object, and tracking a conveyor.

Moreover, algorithms based on the pose goal are not dependent on the manipulator. Hence the same algorithm can be used for different manipulators. The pose goal is computationally heavier than the joint goal, and the solution calculated by IK might not be the shortest trajectory. Figure A.4 shows the difference between the pose goal and joint goal for a manipulator where L_1 and L_2 are the length of the link while θ_1 and θ_2 are the joint angles. The desired pose of the end-effector has 6 variables: 3 for pose (x, y, z) and 3 for orientation (α, β, γ).

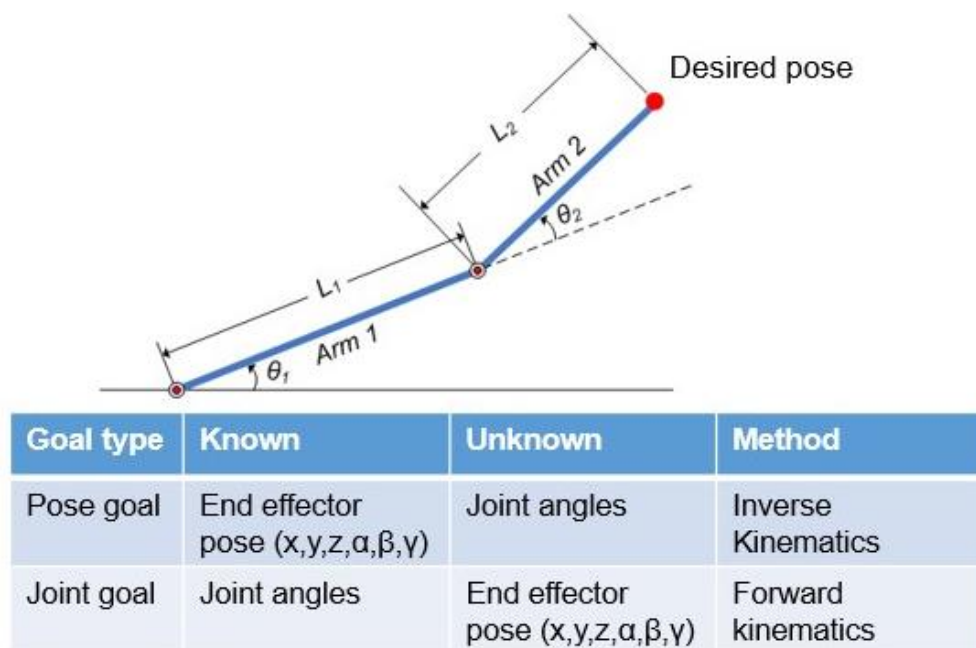


Figure A.3 Difference between pose goal and joint goal

A.3 Gazebo

Gazebo is a popular multi-robot simulator that was initially developed in 2004, is totally open-source, and supports a broad range of sensors and objects. It is compatible with Linux and Windows. Because of close integration with ROS, it is widely used among the ROS robotics developer. It is intended to realistically simulate the dynamics of the various surroundings that a robot may face. It integrates well with ROS, is adaptable, provides realistic simulation by providing option 4 separate physical engines (ODE, Bullet, Simbody, and DART). Gazebo also has a huge and active community which acts as a knowledge sharing platform for this open-source software. The programme supports realistic depiction of surroundings, such as high-quality lighting, shadows, and textures because of OGRE (Object-oriented Graphics Rendering Engine). Sensor data can also be generated by the programme from various imported sensors or cameras.

Gazebo uses Simulation Description Format (SDF) for creating model which is based on Extensible Markup Language (XML). On the other hand, ROS uses Universal Robotic Description Format (URDF) to describe a robot model. Both URDF and SDF uses links and joints to describe the robot hardware. For easy integration, Gazebo has tool to convert the URDF file into SDF format. Moreover, popular 3D mechanical designing software, such a SolidWorks, provides an option to create a URDF file from the 3D model. Once the robot model is ready, Gazebo requires controller to generate motion. Moreover, Gazebo has a sensor plugin library with many popular sensors such as laser, camera, IMU. If necessary, user can define a plugin and add it into Gazebo. Once the robot, sensor and plugins are ready, they can be visualized in the Gazebo world. Flexibility to create and use any world as per user requirement is a very useful feature of Gazebo.

For using Gazebo with ROS, *gazebo_ros_pkgs* [222] were used. This package contains necessary files to create a communication with ROS and control the robot in simulation. To make the data transfer easier, *gazebo_msgs* package use

the same data type for transferring data between ROS and Gazebo. Additionally, *gazebo_plugins* and *gazebo_ros* leverage the available ROS plugins and controller. Figure A.5 shows the integration of Gazebo with ROS [223].

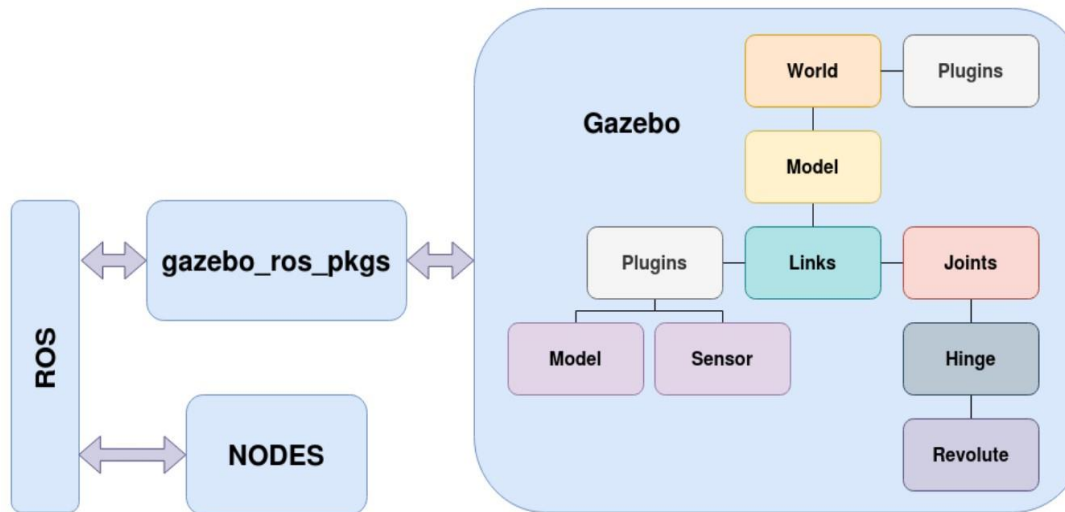


Figure A.4 Gazebo and ROS integration [201]

Appendix B System engineering approach for coding

The implementation of successful systems is made possible by the multidisciplinary field of systems engineering. It emphasises early in the development cycle articulating customer demands and necessary functionality, documenting requirements, and then moving on to design synthesis and system validation while considering the entire challenge. Systems engineering considers both the commercial and technical demands of all clients in order to provide a quality product that fulfils the user's requirements.

A "system" is a design or grouping of several components that when combined yield outcomes that would be impossible for any one component to achieve on its own. People, hardware, software, facilities, rules, and documentation are all examples of pieces or parts that may be used to achieve system-level results. System-level features, attributes, characteristics, functions, behaviour, and performance are among the outcomes. The value produced by the system as a

whole and above what the components offer separately, is largely formed by the interaction between the parts; that is, how they are interrelated.

To develop any system, a system engineering method can ensure a scientific and engineering effort to:

1. Translate the operational requirements into system parameters,
2. Relate and integrate the technical parameters with the functional parameters,
3. Ensure reliability of the system

B.1 Verification and Validation (V&V) model

In system engineering, a verification and validation (V&V) model are required to achieve the benefits of system engineering approach. Among all the system engineering approach, “top-down bottom-up” or “V-model” is very popular. Figure B.2 shows a generic V-model for any system development [199].

In software engineering, verification means the process of software checking to ensure the product output and find out any bugs. On the other hand, validation is the final product after real world or dynamic testing. Some definitions of system engineering model in for software are given below [224]:

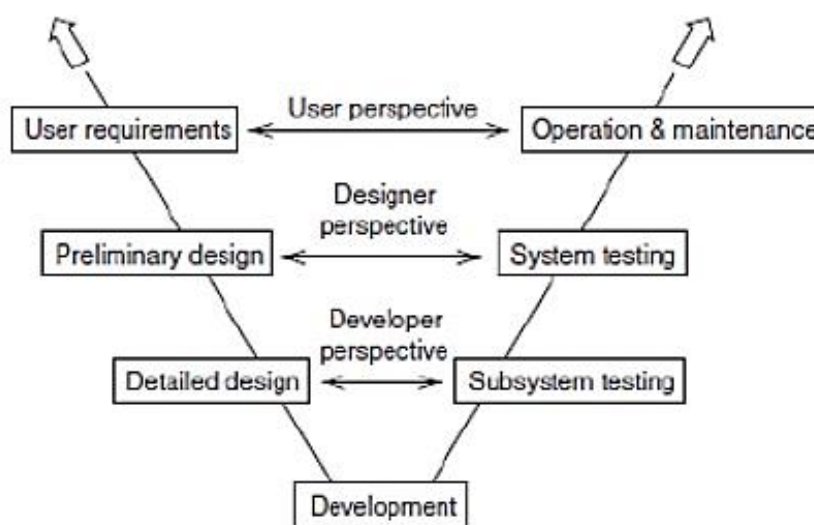


Figure B.1 Generic V-model of system development [202]

Simulation: assembly of models to produce the output from different scenario, especially in any uncertainty

Uncertainty: a probable scenario or action that are not taken care of because of unknown variation or lack of knowledge

Verification: the process to find out the accuracy and precision of the designed system as per the conceptual design of the model

Validation: the process to calculate the degree of confidence for representing accuracy of the model in real world scenario

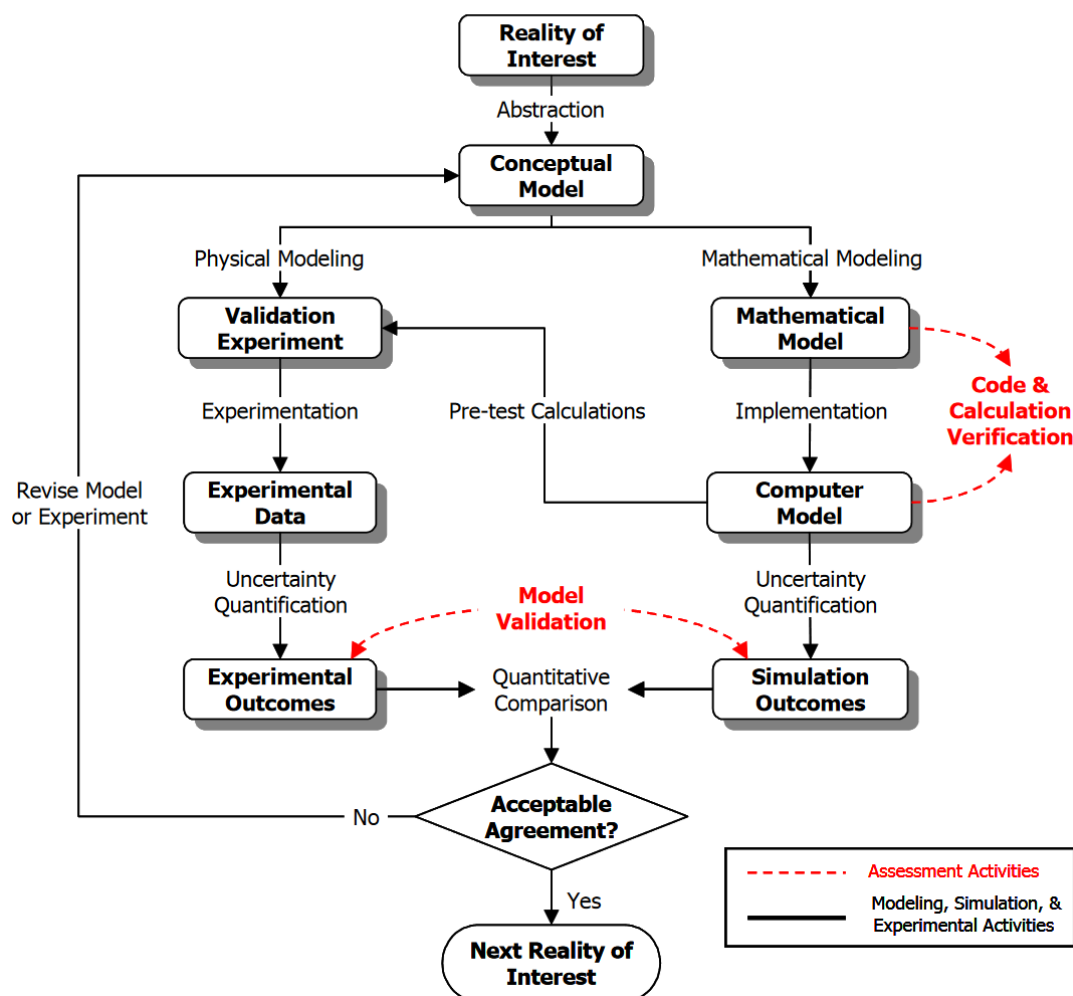


Figure B.2 Detailed verification and validation model for software development [203]

Society for Computer Simulation (SCS) provides detail definition and illustration of V&V model. Figure B. shows the detail illustration of software development using V&V model [224]. The early stage of the development process starts from creating the conceptual model from the reality of interest as per developers or user's requirements. Then, one mathematical model for simulation and one physical model are created. In the simulation model, verified model has been tested in physical model.

Appendix C List of Publications

No	Title	Journal/conference name
01	A Railway Track Reconstruction Method Using Robotic Vision on a Mobile Manipulator: A Proposed Strategy	Computers in Industry, Elsevier
02	Challenges for a Railway Inspection and Repair System from Railway Infrastructure	International Conference on Control, Mechatronics and Automation
03	Investigating Precision and Accuracy of a Robotic Inspection and Repair System	10th International Conference on Through-Life Engineering Services 2021
04	Towards an Autonomous RIRS: Design, Structure Investigation and Framework	7th International Conference on Mechatronics and Robotics Engineering (ICMRE)
05	A review on the prospects of mobile manipulators for smart maintenance of railway track	Applied sciences, MDPI

Appendix D Important Python Scripts

D.1 Manipulator poses

```

#!/usr/bin/env python

import os
from os import path
import sys
import copy

import moveit_commander
import moveit_msgs.msg
import rospy

from math import pi
from std_msgs.msg import String
from moveit_commander.conversions import pose_to_list

class arm_poses_class():
    repair_pose_done=0
    def __init__(self):
        self.move_group_name = "manipulator"
        # arm for gazebo / manipulator for real robot

        moveit_commander.roscpp_initialize(sys.argv)

        robot = moveit_commander.RobotCommander()
        scene = moveit_commander.PlanningSceneInterface()
        self.group =
moveit_commander.MoveGroupCommander(self.move_group_name)
        # manipulator for real robot, arm for gazebo
        display_trajectory_publisher =
rospy.Publisher('/move_group/display_planned_path',
moveit_msgs.msg.DisplayTrajectory, queue_size=1)
        self.group_variable_values1 =
self.group.get_current_joint_values()

    def arm_RITA_pose_verification(self):
        print("Verifying RITA POSE")
        intended_RITA_pose = [-1.4554301, -3.08539305, 2.60211138,
1.6975072, -1.4671238, -0.1373547]
        actual_pose = self.group_variable_values1
        print("Initial pose: ", actual_pose)
        pose_difference = [actual_pose-intended_RITA_pose for
actual_pose, intended_RITA_pose in zip(actual_pose, intended_RITA_pose)]

        if pose_difference[0]>0.01 or pose_difference[1]>0.01 or
pose_difference[2]>0.01:
            print("BIG POSE DIFFERENCE. ABORT! ABORT! ABORT!")

```

```

        print(pose_difference)
        moveit_commander.roscpp_shutdown()
    else:
        print("RITA pose verification successful")

#### old one
def arm_RITA_pose(self):

    group_variable_values = self.group.get_current_joint_values()

    print("moving arm to RITA POSE")
    group_variable_values[0] = -1.4554301
    group_variable_values[1] = -3.08539305
    group_variable_values[2] = 2.60211138
    group_variable_values[3] = 1.6975072
    group_variable_values[4] = -1.4671238
    group_variable_values[5] = -0.1373547
    self.group.set_joint_value_target(group_variable_values)

    plan2 = self.group.plan()
    self.group.go(wait=True)

def arm_straight_pose(self):

    group_variable_values = self.group.get_current_joint_values()

    print("moving arm to INITIAL POSE")
    group_variable_values[0] = -1.23
    group_variable_values[1] = -3.19
    group_variable_values[2] = 0.0
    group_variable_values[3] = 2.155
    group_variable_values[4] = -0.5889
    group_variable_values[5] = -2.477
    self.group.set_joint_value_target(group_variable_values)

    plan2 = self.group.plan()
    self.group.go(wait=True)

def arm_nod_pose_1(self):

    group_variable_values = self.group.get_current_joint_values()

    print("nodding1")
    group_variable_values[0] = -1.23
    group_variable_values[1] = -3.00
    group_variable_values[2] = 0.0

```

```

group_variable_values[3] = 2.155
group_variable_values[4] = -0.5889
group_variable_values[5] = -2.477
self.group.set_joint_value_target(group_variable_values)

plan2 = self.group.plan()
self.group.go(wait=True)

actual_pose = group_variable_values
intended_straight_pose = [-1.23, -3.00, 0.0, 2.155, -0.5889, -
2.477]
pose_difference = [actual_pose-intended_straight_pose for
actual_pose, intended_straight_pose in zip(actual_pose,
intended_straight_pose)]

if pose_difference[0]>0.1 or pose_difference[1]>0.1 or
pose_difference[2]>0.1:
    print("BIG POSE DIFFERENCE. ABORT! ABORT! ABORT!")
    moveit_commander.roscpp_shutdown()
else:
    print("POSE DIFFERENCE INITIAL POSE: ", pose_difference)

def arm_nod_pose_2(self):

    group_variable_values = self.group.get_current_joint_values()

    print("Nodding2")
    group_variable_values[0] = -1.23
    group_variable_values[1] = -3.4
    group_variable_values[2] = 0.0
    group_variable_values[3] = 2.155
    group_variable_values[4] = -0.5889
    group_variable_values[5] = -2.477
    self.group.set_joint_value_target(group_variable_values)

    plan2 = self.group.plan()
    self.group.go(wait=True)

    actual_pose = group_variable_values
    intended_straight_pose = [-1.23, -3.4, 0.0, 2.155, -0.5889, -
2.477]
    pose_difference = [actual_pose-intended_straight_pose for
actual_pose, intended_straight_pose in zip(actual_pose,
intended_straight_pose)]

```

```

        if pose_difference[0]>0.1 or pose_difference[1]>0.1 or
pose_difference[2]>0.1:
            print("BIG POSE DIFFERENCE. ABORT! ABORT! ABORT!")
            moveit_commander.roscpp_shutdown()
        else:
            print("POSE DIFFERENCE INITIAL POSE: ", pose_difference)

##### old one
def arm_inspection_pose(self):

    group_variable_values = self.group.get_current_joint_values()

    # values are in radian
    print("moving arm to LEFT INSPECTION POSE")
    group_variable_values[0] = -1.05679686
    group_variable_values[1] = -3.29867
    group_variable_values[2] = -0.820304748
    group_variable_values[3] = -0.590444886
    group_variable_values[4] = 1.5727162
    group_variable_values[5] = -2.66703763
    self.group.set_joint_value_target(group_variable_values)

    plan2 = self.group.plan()
    self.group.go(wait=True)

def arm_RITA_pose(self):

    group_variable_values = self.group.get_current_joint_values()

    # values are in radian
    print("moving arm to RITA POSE")
    group_variable_values[0] = -1.5596
    group_variable_values[1] = -3.1785
    group_variable_values[2] = -2.6677
    group_variable_values[3] = 5.5993
    group_variable_values[4] = -3.27266
    group_variable_values[5] = -1.3939
    self.group.set_joint_value_target(group_variable_values)

    plan2 = self.group.plan()
    self.group.go(wait=True)

## 4 repair pose for taking 4 pictures from 4 different angle
def arm_img_cap_1(self):

    group_variable_values = self.group.get_current_joint_values()

```

```

## pose 1: up
print("Repair pose 01....")
group_variable_values[0] = -0.75607663
group_variable_values[1] = -3.3388149
group_variable_values[2] = -0.74490652
group_variable_values[3] = -0.62203535
group_variable_values[4] = 1.2527973
group_variable_values[5] = -2.66703
self.group.set_joint_value_target(group_variable_values)

plan1 = self.group.plan()
self.group.go(wait=True)

#moveit_commander.roscpp_shutdown()
arm_poses_class.repair_pose_done=1
#return self.arm_straight_pose

def arm_img_cap_2(self):

    group_variable_values = self.group.get_current_joint_values()

    print("Repair pose 02....")
    group_variable_values[0] = -1.1018264
    group_variable_values[1] = -3.232175
    group_variable_values[2] = -0.05969
    group_variable_values[3] = -1.7367771
    group_variable_values[4] = 1.66399
    group_variable_values[5] = -2.666836
    self.group.set_joint_value_target(group_variable_values)

    plan2 = self.group.plan()
    self.group.go(wait=True)

    arm_poses_class.repair_pose_done=2

def arm_img_cap_3(self):

    group_variable_values = self.group.get_current_joint_values()
    ## pose 3: down
    print("Repair pose 03....")
    group_variable_values[0] = -0.997106
    group_variable_values[1] = -3.495196
    group_variable_values[2] = -0.61365776
    group_variable_values[3] = -0.5103342
    group_variable_values[4] = 1.547758
    group_variable_values[5] = -2.6801276

```

```

self.group.set_joint_value_target(group_variable_values)

plan3 = self.group.plan()
self.group.go(wait=True)

arm_poses_class.repair_pose_done=3

def arm_img_cap_4(self):

    group_variable_values = self.group.get_current_joint_values()
    ## pose 3: left
    print("Repair pose 04....")
    group_variable_values[0] = -1.4248868
    group_variable_values[1] = -3.2710960
    group_variable_values[2] = -0.5900958
    group_variable_values[3] = -1.1166617
    group_variable_values[4] = 2.2376866
    group_variable_values[5] = -2.969677
    self.group.set_joint_value_target(group_variable_values)

    plan4 = self.group.plan()
    self.group.go(wait=True)

    arm_poses_class.repair_pose_done=4

##### these repair poses for pushing the object
#####
def arm_high_up(self):

    group_variable_values = self.group.get_current_joint_values()
    ## pose 3: left
    print("Arm high pose....")
    group_variable_values[0] = -1.2733
    group_variable_values[1] = -2.1126
    group_variable_values[2] = -2.2394
    group_variable_values[3] = 4.2313
    group_variable_values[4] = -2.8855
    group_variable_values[5] = -0.9787
    self.group.set_joint_value_target(group_variable_values)

    plan4 = self.group.plan()
    self.group.go(wait=True)

def arm_high_down(self):

    group_variable_values = self.group.get_current_joint_values()
    ## pose 3: left

```

```

print("Arm down pose....")
group_variable_values[0] = -1.3105
group_variable_values[1] = -2.6850
group_variable_values[2] = -2.072
group_variable_values[3] = 4.5053
group_variable_values[4] = -2.9617
group_variable_values[5] = -0.9920
self.group.set_joint_value_target(group_variable_values)

plan4 = self.group.plan()
self.group.go(wait=True)

### for pushing left side
def arm_push_left_pose_1(self):

    group_variable_values = self.group.get_current_joint_values()
    ## pose 3: left
    print("Pushing left object....")
    group_variable_values[0] = -0.7516
    group_variable_values[1] = -2.6804
    group_variable_values[2] = -1.9316
    group_variable_values[3] = 4.5298
    group_variable_values[4] = -2.3560
    group_variable_values[5] = -0.819
    self.group.set_joint_value_target(group_variable_values)

    plan4 = self.group.plan()
    self.group.go(wait=True)

def arm_push_left_pose_2(self):

    group_variable_values = self.group.get_current_joint_values()
    ## pose 3: left
    print("Pushing left object....")
    group_variable_values[0] = -0.8126
    group_variable_values[1] = -3.0367
    group_variable_values[2] = -1.6148
    group_variable_values[3] = 4.6017
    group_variable_values[4] = -2.4012
    group_variable_values[5] = -0.8643
    self.group.set_joint_value_target(group_variable_values)

    plan4 = self.group.plan()
    self.group.go(wait=True)

def arm_push_left_pose_3(self):

```

```

group_variable_values = self.group.get_current_joint_values()
## pose 3: left
print("Pushing left object....")
group_variable_values[0] = -0.9912
group_variable_values[1] = -3.0784
group_variable_values[2] = -1.3132
group_variable_values[3] = 4.4723
group_variable_values[4] = -2.5886
group_variable_values[5] = -1.1137
self.group.set_joint_value_target(group_variable_values)

plan4 = self.group.plan()
self.group.go(wait=True)

#### for pushing middle
def arm_push_middle_pose_1(self):

    group_variable_values = self.group.get_current_joint_values()
    ## pose 3: left
    print("Pushing middle object....")
    group_variable_values[0] = -1.2975
    group_variable_values[1] = -3.0465
    group_variable_values[2] = -1.7245
    group_variable_values[3] = 4.6331
    group_variable_values[4] = -2.8885
    group_variable_values[5] = -0.9170
    self.group.set_joint_value_target(group_variable_values)

    plan4 = self.group.plan()
    self.group.go(wait=True)

def arm_push_middle_pose_2(self):

    group_variable_values = self.group.get_current_joint_values()
    ## pose 3: left
    print("Pushing middle object....")
    group_variable_values[0] = -1.3950
    group_variable_values[1] = -3.0370
    group_variable_values[2] = -1.4476
    group_variable_values[3] = 4.7105
    group_variable_values[4] = -2.9812
    group_variable_values[5] = -0.9604
    self.group.set_joint_value_target(group_variable_values)

```

```

plan4 = self.group.plan()
self.group.go(wait=True)

### for pushing right
def arm_push_right_pose_1(self):

    group_variable_values = self.group.get_current_joint_values()
    ## pose 3: left
    print("Pushing right object....")
    group_variable_values[0] = -1.8973
    group_variable_values[1] = -2.7363
    group_variable_values[2] = -1.9253
    group_variable_values[3] = 4.8126
    group_variable_values[4] = -3.5155
    group_variable_values[5] = -0.6611
    self.group.set_joint_value_target(group_variable_values)

    plan = self.group.plan()
    self.group.go(wait=True)

    arm_poses_class.repair_pose_done=4

def arm_push_right_pose_2(self):

    group_variable_values = self.group.get_current_joint_values()
    ## pose 3: left
    print("Pushing middle object....")
    group_variable_values[0] = -1.8610
    group_variable_values[1] = -3.0442
    group_variable_values[2] = -1.5917
    group_variable_values[3] = 4.7211
    group_variable_values[4] = -3.4638
    group_variable_values[5] = -0.7147
    self.group.set_joint_value_target(group_variable_values)

    plan = self.group.plan()
    self.group.go(wait=True)

def arm_push_right_pose_3(self):

    group_variable_values = self.group.get_current_joint_values()
    ## pose 3: left
    print("Pushing middle object....")
    group_variable_values[0] = -1.8026
    group_variable_values[1] = -3.0544

```

```

group_variable_values[2] = -1.3611
group_variable_values[3] = 4.7334
group_variable_values[4] = -3.4000
group_variable_values[5] = -0.8982
self.group.set_joint_value_target(group_variable_values)

plan = self.group.plan()
self.group.go(wait=True)

def arm_test_pose(self):

    group_variable_values = self.group.get_current_joint_values()
    ## pose 3: left
    print("Pushing middle object....")
    group_variable_values[0] = 1.5708
    group_variable_values[1] = 3
    group_variable_values[2] = 2.5481
    group_variable_values[3] = -0.8839
    group_variable_values[4] = 1.5104
    group_variable_values[5] = -3.0953
    self.group.set_joint_value_target(group_variable_values)

    plan = self.group.plan()
    self.group.go(wait=True)

def arm_joint_goal_pose(self):

    group_variable_values = self.group.get_current_joint_values()
    print("current pose: ", group_variable_values)
    ## pose 3: left
    print("Testing joint space goal from PSO....")
    group_variable_values[0] = 0.0
    group_variable_values[1] = -1.4
    group_variable_values[2] = 2.5481
    group_variable_values[3] = -0.8839
    group_variable_values[4] = 1.5104
    group_variable_values[5] = -3.0953
    self.group.set_joint_value_target(group_variable_values)

def main():
    try:
        tutorial = arm_poses_class()

        print("Executing goal ..... ")

        tutorial.arm_joint_goal_pose()

```

```

        print ("===== Python tutorial demo complete!")
    except rospy.ROSInterruptException:
        return
    except KeyboardInterrupt:
        return

if __name__ == '__main__':
    main()

```

D.2 Code for Human-in-the-loop-repair task framework

```

#!/usr/bin/env python

import code
import os
from os import path
import sys
import time

import pynput
import roslib
import rospy

from pynput import keyboard

from arm_poses import arm_poses_class
from gps_saver import gps_saver_class
from image_saver import image_saver_class
from move_forward_backward import movement
from geometry_msgs.msg import Twist
from math import atan2
from std_msgs import msg
from std_msgs.msg import String, Int8

class inspection_and_repair():
    def __init__(self):
        self.defect_found = Int8()
        self.aux = Twist()
        self.defect_found = 0
        self.status_msg = Int8()
        self.status_msg.data = 0
        self.code_msg = String()
        self.gps_obj = gps_saver_class()
        self.image_obj = image_saver_class()

```

```

qr_code_msg_1 = 'http://Old'
qr_code_msg_2 = 'http://'
qr_code_msg_3 = 'http://B Type. Do cutting.'

rospy.Subscriber('/visp_auto_tracker/status', Int8,
self.defect_callback, queue_size = 10)
self.speed_pub = rospy.Publisher('/cmd_vel', Twist, queue_size =
1)

while self.defect_found<1:
    # staus_sub =
rospy.wait_for_message('/visp_auto_tracker/status', Int8)
    print("Looking for defects.....?")
    if self.status_msg.data < 2:

        print (self.status_msg.data)
        self.aux.linear.x =0.0 ##### use values for forward or
backward motion
        self.speed_pub.publish(self.aux)
    else:
        self.aux.linear.x = 0
        self.speed_pub.publish(self.aux)
        rospy.sleep(3)
        print (self.status_msg.data)
        print("Found defect, lets find out more details >>>")
        while not (self.code_msg==qr_code_msg_1 or
self.code_msg==qr_code_msg_2 or self.code_msg==qr_code_msg_3):
            code_msg_sub =
rospy.wait_for_message('/visp_auto_tracker/code_message', String)
            self.code_msg = code_msg_sub.data
            print("SEARCHING!!!SEARCHING")
            #return code_msg

        if self.code_msg == qr_code_msg_1:
            print("This is an old defect: drop left object")
            arm_poses_class().arm_high_up()
            arm_poses_class().arm_push_left_pose_1()
            arm_poses_class().arm_push_left_pose_2()
            arm_poses_class().arm_push_left_pose_3()
            arm_poses_class().arm_high_up()
            rospy.on_shutdown()
            # return defect_found
        elif self.code_msg == qr_code_msg_2:
            print("qr code msg2: ", self.code_msg)
            defect_found = 1
            print("This is a new defect; Type A: push middle
object")

```

```

        arm_poses_class().arm_high_up()
        arm_poses_class().arm_high_down()
        arm_poses_class().arm_push_middle_pose_1()
        arm_poses_class().arm_push_middle_pose_2()
        arm_poses_class().arm_high_up()
        rospy.on_shutdown()

        # return defect_found
    else:
        print("qr code msg3: ", self.code_msg)
        defect_found = 1
        print("This is a new defect; Type B: push right
object")

        arm_poses_class().arm_high_up()
        arm_poses_class().arm_push_right_pose_1()
        arm_poses_class().arm_push_right_pose_2()
        arm_poses_class().arm_push_right_pose_3()
        arm_poses_class().arm_high_up()
        rospy.on_shutdown()
        # return defect_found

def defect_callback(self,msg):
    self.status_msg = msg

def loop(self):
    rospy.spin()

if __name__=='__main__':
    rospy.init_node('inspection_repair', anonymous=False)

    try:
        print("***LETS INITIALIZE***")

        arm_poses_class().arm_test_pose()
        print("### LETS START INSPECTION")

        print("JOB COMPLETED")
        print("Let's go back to initial pose")

    except KeyboardInterrupt:
        print("SHUTTING DOWN")

```

D.3 Code for automatic trajectory generation framework

```
#!/usr/bin/env python
```

```
## BEGIN_SUB_TUTORIAL imports
```

```

import sys
import copy
from turtle import pos, position
import rospy
import moveit_commander
import moveit_msgs.msg
import geometry_msgs.msg
from math import pi
from std_msgs.msg import String
from moveit_commander.conversions import pose_to_list
import tf
import tf2_ros
import tf_conversions

from geometry_msgs.msg import Pose
import math

from fileinput import filename
from datetime import datetime
import os
from os import path
import csv

## END_SUB_TUTORIAL

def all_close(goal, actual, tolerance):
    """
    Convenience method for testing if a list of values are within a
    tolerance of their counterparts in another list
    @param: goal      A list of floats, a Pose or a PoseStamped
    @param: actual    A list of floats, a Pose or a PoseStamped
    @param: tolerance A float
    @returns: bool
    """
    all_equal = True
    if type(goal) is list:
        for index in range(len(goal)):
            if abs(actual[index] - goal[index]) > tolerance:
                return False

    elif type(goal) is geometry_msgs.msg.PoseStamped:
        return all_close(goal.pose, actual.pose, tolerance)

    elif type(goal) is geometry_msgs.msg.Pose:
        return all_close(pose_to_list(goal), pose_to_list(actual), tolerance)

```

```

return True

class MoveGroupPythonInterfaceTutorial(object):
    """MoveGroupPythonInterfaceTutorial"""
    def __init__(self):
        super(MoveGroupPythonInterfaceTutorial, self).__init__()

        moveit_commander.roscpp_initialize(sys.argv)
        rospy.init_node('move_group_python_interface_tutorial',
                        anonymous=True)

        translation = rospy.wait_for_message('translationBetweenFrames', Pose)

        self.obj_position_x = translation.position.x
        self.obj_position_y = translation.position.y
        self.obj_position_z = translation.position.z+0.45
        print("positions ", self.obj_position_x, self.obj_position_y)

        robot = moveit_commander.RobotCommander()
        scene = moveit_commander.PlanningSceneInterface()
        self.group_name = "manipulator"
        self.group = moveit_commander.MoveGroupCommander(self.group_name)

        display_trajectory_publisher =
rospy.Publisher('/move_group/display_planned_path',
                                                         moveit_msgs.msg.Displa
yTrajectory,
                                                         queue_size=20)

        planning_frame = self.group.get_planning_frame()
        print ("===== Reference frame: %s" % planning_frame)

        eef_link = self.group.get_end_effector_link()
        print ("===== End effector: %s" % eef_link)

        group_names = robot.get_group_names()
        print ("===== Robot Groups:", robot.get_group_names())

        # Misc variables
        self.number_of_waypoints = 40
        self.object_width = 0.2
        self.object_length = 0.3
        self.circle_radius = 0.2
        self.theta = 0

    def go_to_pose_goal1(self):

```

```

group = self.group
current_pose = self.group.get_current_pose().pose

print ("Current pose: ", current_pose)
pose_goal = geometry_msgs.msg.Pose()
pose_number = 0
decrement = (2*pi)/self.number_of_waypoints
while self.theta < (2*pi):
    self.theta = self.theta + decrement
    pose_number = pose_number + 1
    pose_goal.position.x =self.obj_position_x +
(self.circle_radius*math.cos(self.theta))
    pose_goal.position.y =self.obj_position_y +
(self.circle_radius*math.sin(self.theta))
    pose_goal.position.z = self.obj_position_z
    pose_goal.orientation =
geometry_msgs.msg.Quaternion(*tf_conversions.transformations.quaternion_f
rom_euler(0.0, 1.57, 3.1416))
    # for looking down 0.0, 1.57, 3.1416 in radians
    self.group.set_pose_target(pose_goal)
    print("Goal pose: ", pose_goal)
    print("Waypoint no and theta ", pose_number, self.theta)

    plan = self.group.go(wait=True)
    self.group.stop()
    self.group.clear_pose_targets()

    csv_file = 'poses.csv'
    header = ['pose_number', 'x', 'y', 'z']
    poses_to_save = self.obj_position_x, self.obj_position_y,
self.obj_position_z, pose_goal.position.x, pose_goal.position.y,
pose_goal.position.z
    data = [pose_number, poses_to_save]
    with open(csv_file, mode='a') as f:
        writer = csv.writer(f)
        writer.writerow(data)
    rospy.loginfo("finish saving poses")

    return all_close(pose_goal, current_pose, 0.01)

def go_to_pose_goal2(self):
    group = self.group

    current_pose = self.group.get_current_pose().pose
    print("Current POSE: ", current_pose)

```

```

    pose_goal = geometry_msgs.msg.Pose()
    pose_goal.position.x = self.obj_position_x
    pose_goal.position.y = self.obj_position_y
    pose_goal.position.z = self.obj_position_z + 0.35
    pose_goal.orientation =
geometry_msgs.msg.Quaternion(*tf_conversions.transformations.quaternion_f
rom_euler(0.0, pi/2, pi))
    group.set_pose_target(pose_goal)
    print("2nd Goal pose: ", pose_goal)

    ## Now, we call the planner to compute the plan and execute it.
    plan = group.go(wait=True)
    group.stop()
    group.clear_pose_targets()

    return all_close(pose_goal, current_pose, 0.01)

def display_trajectory(self, plan):
    # Copy class variables to local variables to make the web tutorials
more clear.
    # In practice, you should use the class variables directly unless you
have a good
    # reason not to.
    robot = self.robot
    display_trajectory_publisher = self.display_trajectory_publisher

    display_trajectory = moveit_msgs.msg.DisplayTrajectory()
    display_trajectory.trajectory_start = robot.get_current_state()
    display_trajectory.trajectory.append(plan)
    # Publish
    display_trajectory_publisher.publish(display_trajectory)

def execute_plan(self, plan):
    # Copy class variables to local variables to make the web tutorials
more clear.
    # In practice, you should use the class variables directly unless you
have a good
    # reason not to.
    group = self.group
    group.execute(plan, wait=True)

def main():
    try:
        tutorial = MoveGroupPythonIntefaceTutorial()

```

```

    print("Executing pose goal ..... ")
    tutorial.go_to_pose_goal1()

    print ("==== Python tutorial demo complete!")
except rospy.ROSInterruptException:
    return
except KeyboardInterrupt:
    return

if __name__ == '__main__':
    main()

```

D.4 Code for tf transformation of object

```
#!/usr/bin/env python
```

```

from array import array
from logging import shutdown
import rospy
import math
import tf
from std_msgs.msg import Float64,Float64MultiArray
from geometry_msgs.msg import Pose
import geometry_msgs.msg
import tf_conversions

def translation_tf():
    pub = rospy.Publisher('translationBetweenFrames',Pose, queue_size=30)
    rospy.init_node('translation', anonymous=False)
    trans1 = []
    trans2 = []
    listener = tf.TransformListener()
    rate = rospy.Rate(10) # 10hz
    while not rospy.is_shutdown():
        try:
            (trans1, rot1) =
listener.lookupTransform('/ur_arm_base_link','/ur_arm_ee_link',
rospy.Time(0))

            data = Pose()
            data.position.x = trans1[0]
            data.position.y = trans1[1]
            data.position.z = trans1[2]

            print("x_trans: ", data.position.x)

```

```

        print("y_trans: ", data.position.y)
        print("z_trans: ", data.position.z)
        pub.publish(data)
    except (tf.LookupException, tf.ConnectivityException,
tf.ExtrapolationException):
        continue

if __name__ == '__main__':
    try:
        translation_tf()
    except rospy.ROSInterruptException:
        pass

```

Appendix E Repeatability and precision

E.1 Data table for repeatability from 10 experiments

Table E-1 Experimental data from repeatability tests

	test no	distance moved by vehicle	distance between base to arm head (fixed)	distance between arm head to object	total distance from vehicle start point to defect	mean of total distance	actual	error (calculated- actual)	error percent
	1	2.750	1.003	0.015	3.767	3.782	3.750	0.017	0.464
	2	2.732	1.003	0.037	3.772	3.782	3.750	0.022	0.579
	3	2.757	1.003	0.017	3.777	3.782	3.750	0.027	0.717
	4	2.748	1.003	0.026	3.777	3.782	3.750	0.027	0.717
	5	2.765	1.003	0.019	3.787	3.782	3.750	0.037	0.997
	6	2.762	1.003	0.041	3.806	3.782	3.750	0.056	1.496
	7	2.740	1.003	0.029	3.772	3.782	3.750	0.021	0.573
	8	2.768	1.003	0.019	3.790	3.782	3.750	0.040	1.074
	9	2.731	1.003	0.047	3.780	3.782	3.750	0.030	0.804
	10	2.753	1.003	0.046	3.801	3.782	3.750	0.051	1.371
mean		2.750		0.030	3.783		3.750	0.033	0.879
maximum		2.768		0.047	3.806	3.782	3.750	0.056	
minimum		2.731		0.015	3.767	3.782	3.750	0.017	
range		0.038		0.032	0.039	0.000	0.000	0.039	