DEGREE PROJECT FOR MASTER OF SCIENCE WITH SPECIALIZATION IN ROBOTICS
DEPARTMENT OF ENGINEERING SCIENCE
UNIVERSITY WEST

Configuration Sensitivity and Robustness of Single Robot Automatic Path Planning in ABB RobotStudio

Dhanya Prabhakaran

A THESIS SUBMITTED TO THE DEPARTMENT OF ENGINEERING SCIENCE

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE WITH SPECIALIZATION IN ROBOTICS

AT UNIVERSITY WEST

2026

Date:	September 11 , 2026
Author:	Dhanya Prabhakaran
Examiner:	Sudha Ramaswamy
Advisor:	Bassam Massouh (University), Mikael Norrlof (ABB Robotics)
Programme:	Master Programme in Robotics
Main field of study:	Automation with a specialization in industrial robotics
Credits:	30 Higher Education credits
Keywords:	Collision free, Path Planning, ABB Robotstudio, Automatic Path Planning, CFree, Industrial Robotics, Parameter Sensitivity
Template:	University West, IV-Master 2.10
Publisher:	University West, Department of Engineering Science S-461 86 Trollhättan, SWEDEN Phone: + 46 520 22 30 00 Fax: + 46 520 22 32 99 Web: www.hv.se

Summary

Since industrial automation shifts towards flexible manufacturing, the need for reliable, collision free robot trajectories have become very critical and thus Collision Free Path Planning can be considered as an important technology. But, while academic research focuses heavily on developing new path planning algorithms, the practical configuration of existing commercial tools remains underexplored. This thesis investigates the configuration sensitivity and parameter robustness of ABB's Automatic Path Planning technology. Using a simulation based One-Factor-At-A-Time (OFAT) methodology, three core parameters such as optimality effort, minimum distance to obstacles, and timeout were systematically evaluated across three robot cell layouts (Tight, Moderate, and Open) using an ABB CRB 1100 manipulator. The results demonstrate that parameter performance is entirely bounded by the physical geometry of the cell. Also attempting to enforce safety clearances greater than the target proximity results in 100% planning failure, and maximizing optimality effort in unconstrained spaces causes severe computational waste (tripling planning time with 0% path improvement). Ultimately, this thesis compiles these findings into practical Configuration Profiles (Speed, Balanced, Quality) and Color-Coded Decision Matrices, thereby providing industrial commissioning engineers with a data driven methodology to configure commercial path planners based on specific factory environments.

Preface

This Master thesis was conducted in collaboration with ABB Robotics. I would like to express my sincere gratitude to my University supervisor Bassam Massouh and ABB Robotics supervisor, Mikael Norrlöf, for his invaluable guidance, technical insights, and continuous support throughout this project. I also extend my thanks to Morten Åkerblad and the APP software development team at ABB for their expert advice and willingness to discuss the underlying mechanics of the planning system. Finally, I would like to thank University West for providing the academic framework that made this research possible.

Affirmation

This Master degree report, Configuration Sensitivity and Robustness of Single-Robot Automatic Path Planning in ABB RobotStudio, was written as part of the Master degree work needed to obtain a Master of Science with specialization in Robotics degree at University West. All material in this report, that is not my own, is clearly identified and used in an appropriate and correct way. The main part of the work included in this degree project has not previously been published or used for obtaining another degree.

Signature by the author

Date

Dhanya Prabhakaran

Contents

Preface

SUMMARY ..... **III**

PREFACE ..... **IV**

AFFIRMATION ..... **V**

CONTENTS ..... **VI**

SYMBOLS AND GLOSSARY ..... **VIII**

Main Chapters

1 INTRODUCTION **1**

1.1 Background and Context 1
1.2 Problem Definition 2
1.3 Aims and Research Questions 3
1.4 Contribution 3
1.5 Limitations 3

2 RELATED WORK (BACKGROUND) **5**

2.1 State of the Art in Automatic Path Planning 5
2.2 Algorithmic Sensitivity to Parameter Configuration 6
2.3 Planning in Constrained and Complex Geometries 7
2.4 The Gap in Evaluating Commercial Path Planning Tools 8

3 METHOD **9**

3.1 Research Approach 9
3.2 Experimental Design (OFAT) 9
3.3 Data Collection and Evaluation Metrics 9
3.4 Tools, Reliability, and Error Margins 10
3.5 Workflow of the Study 10

4 EXPERIMENTAL SETUP AND IMPLEMENTATION **12**

4.1 Parameter Identification and Exclusion 12
4.2 Simulated Industrial Environments 13
4.3 OFAT baseline configuration 15
4.4 Formulation of Configuration Profiles 16

5 RESULTS AND DISCUSSION **18**

5.1 OFAT Parameter Sensitivity Results 18
5.2 Configuration Profile Trade offs 19
5.3 Guidelines for Industrial Commissioning 23
5.4 Discussion and Industrial implications 25

6 CONCLUSION **26**

6.1 Future Work and Research 26

6.2 CRITICAL DISCUSSION 27
6.3 GENERALIZATION OF THE RESULT 27
7 REFERENCES 28

Appendices

A. APPENDIX

Symbols and glossary

APP	Automatic Path Planning feature from ABB Robotics used to automatically generate collision-free robot trajectories based on a digital model of the robot cell.
CFree	The backend planning engine in ABB RobotStudio responsible for computing collision-free paths.
PPD	Path Planning Data. An exported file format from RobotStudio containing the geometric meshes and kinematics required by the planning server.
D	The physical, real-world distance between the robot's tool and the nearest resting surface at the approach/retract target.
KPI	Key Performance Indicator (e.g., planning time, path length, success rate).
OFAT	One-Factor-At-A-Time. An experimental design method where a single parameter is varied while all others are held at a fixed baseline.
RRT	Rapidly-exploring Random Tree. A sampling-based algorithm used for path planning in high-dimensional spaces.
SOTA	State of the Art
TCP	Tool Center Point. The reference point at the end of the robot tool used for positioning.

1 Introduction

Industrial robots are widely integrated across manufacturing sectors, supporting operations such as assembly, material handling, welding, spray painting, inspection, and flexible automation. As production systems evolve towards more complex layouts, modularity, and human-robot coexistence, the requirement for safe and reliable robot motion is increasingly important. Path planning, which is the computation of a collision-free, feasible robot trajectory based on geometric, kinematic, and process constraints, plays a central role in modern industrial automation.

For the last several years, path planning has continued to be a core research area. However, the behaviour of commercial planning systems in realistic industrial environments is not sufficiently explored. Many academic methods assume perfect geometric knowledge and idealised environments, but these are rarely met in production cells. This gap between theoretical planning models and industrial reality motivates the present thesis, which focuses on the configuration sensitivity and robustness of ABB's Automatic Path Planning (APP) system under varying geometric conditions.

1.1 Background and Context

Path planning is recognised as a fundamental requirement for safe robot operation, especially where robots share densely cluttered workspaces [1]. At its core, path planning seeks to determine a collision-free motion from a defined start configuration to a goal configuration within a high-dimensional configuration space that reflects the robot's kinematic structure.

There are various algorithmic approaches that have been developed over time. Classical methods such as A* and Dijkstra operate on discretised search spaces and offer determinism, but they require exhaustive discretisation, making them impractical for high-dimensional industrial manipulators. Sampling based planners, including Probabilistic Roadmaps (PRM) and Rapidly exploring Random Trees (RRT), are particularly attractive for industrial robots due to their ability to handle high-dimensional spaces efficiently. Recent research has also explored topology-aware formulations to address complex constraints [2], [3].

In response to the need for faster commissioning and reduced manual programming effort, commercial robot manufacturers have integrated these algorithms into proprietary automatic path planning tools. ABB's Automatic Path Planning (APP), available through the CFree backend in Robotstudio, is an example for this transition from manual teach-in to automated trajectory generation. APP generates collision-free robot paths based on a digital representation of the robot cell. It allows users to configure planning parameters that influence how the search space is explored and how safety margins are enforced.

However, planning performance is strongly influenced by how the configuration space is restricted. More advanced studies addressing multi-robot coordination and dynamic environments further highlight the importance of maintaining reliable planning performance under spatial constraints [4], [5]. Hybrid frameworks combining sampling-based exploration with evolutionary refinement have been proposed to mitigate these issues [6], [7]. Yet, studies focusing on constrained environments such as micro-assembly [8] consistently report that planning performance degrades when geometric complexity increases. In this thesis, realistic geometric constraints refer to the varying spatial layouts, obstacle densities, and clearance bottlenecks found in industrial CAD models. Evaluating configuration sensitivity means testing how well the APP system and its parameters handle these geometric restrictions without failing or producing highly inefficient paths.

1.2 Problem Definition

In industrial practice, automatic path planning tools are expected to generate reliable motions. However, deployment experience indicates that the behaviour of such tools is highly sensitive to parameter choices and cell geometry. For high-degree-of-freedom industrial manipulators, redundancy resolution introduces additional complexity [9]. Application-specific reviews, such as those addressing robotic spray painting and control-cabinet wiring, further illustrate that modelling fidelity directly affects the resulting path quality [8], [9]. A review of existing literature reveals a distinct gap between academic research and industrial application, explaining why this problem remains largely unsolved:

a. The gap in academic research and algorithm design
Academic research heavily focuses on developing new algorithms to improve convergence or path quality under idealized, static conditions. For example, recent studies on improved RRT algorithms [10] prove that environmental setup drastically affects path quality. However, academia rarely evaluates the commercial, "black-box" tools actually used in industry. In automated control-cabinet wiring, planners often produce highly variable results or fail entirely due to spatial bottlenecks and dense obstacle placement [11]. Similarly, studies indicate that classical planning approaches degrade significantly when faced with complex conditions [12]. Existing studies on algorithms like RRT* or PRM show they are really sensitive to parameter settings [10]. While the literature proves that algorithms are inherently sensitive, it fails to provide practical configuration guidelines for existing commercial software. This thesis is necessary to bridge the gap between academic algorithm theory and industrial software application.

b. Sensitivity to configuration choices
Commercial Automatic path planning systems like APP provide users with configuration settings (e.g., timeouts, optimality effort, minimum distance). In industrial environments, these settings are often chosen based on experience or default values rather than quantified knowledge. Planning performance varies significantly due to configuration changes, yet it is unclear which settings are critical and how sensitive the overall planning outcome is to these choices.

c. Missing guidance for industrial commissioning
Due to the absence of quantitative robustness data, industrial commissioning relies on trial-and-error tuning . When planning failures occur in constrained geometries, it is

challenging to determine whether the cause lies in improper configuration choices or the inherent geometric bottlenecks of the cell. This uncertainty reduces the effectiveness of automated planning.

1.3 Aims and Research Questions

Aim:
The aim of this thesis is to develop a practical, empirically grounded reference for the industrial use of ABB's Automatic Path Planning system. This is done by analysing how planning reliability and robustness are affected by configuration choices and realistic geometric constraints in a simulated robot cell. The study aims to establish an understanding of the trade-offs involved rather than prescribing a single optimal solution by systematically characterising how changes in planner configuration and cell geometry influence key performance indicators (KPIs). The final outcome is intended to provide guidance to support industrial users in making informed decisions during commissioning, particularly with respect to acceptable operating ranges and robustness limits.

Research Question:
What trade-offs and acceptable operating ranges can be identified for ABB's Automatic Path Planning system, considering the combined effects of configuration choices and realistic geometric deviations in industrial robot cells?

Sub Questions:

1. How does planning behaviour change in response to variations in Automatic Path Planning configuration settings, and which settings have the greatest influence on planning performance and stability?
   (Note: Planning performance refers to the efficiency of the outcome, such as computation time and path length. Stability refers to the consistency of the planner to successfully find a valid, collision-free path without failure).
2. How do geometric constraints and bottleneck variations in the simulated robot cell affect the robustness of Automatic Path Planning outcomes?

1.4 Contribution

This thesis contributes the following:

- A systematic sensitivity and robustness evaluation of ABB's Automatic Path Planning system under realistic industrial conditions.
- A quantitative characterisation of planning KPIs across multiple scenarios involving parameter configuration sweeps and geometric layout changes.
- A practical reference and decision-support guide to aid industrial commissioning, including recommendations on acceptable parameter operating ranges based on specific application priorities.

1.5 Limitations

This study evaluates parameter configuration within a specific, representative boundary. The experimental setup utilizes an ABB CRB 1100 collaborative robot performing a

standard pick-and-place operation. While the ABB APP system supports various manipulators, this study restricts its evaluation to the CRB 1100 to maintain controlled variables.

Also, industrial robot cells vary drastically in geometry and application (eg: continuous welding, spray painting). The findings of this thesis, specifically the raw numerical thresholds discovered, are limited to the specific pick-and-place scenario evaluated. Additionally, this thesis isolates internal software parameters within a digital twin and external physical deployment errors such as real-world calibration inaccuracies are excluded. The purpose of this study is not to provide universal numerical values applicable to all manufacturing tasks, but rather to present a validated methodology that engineers can adapt to systematically identify parameter limits for their own specific cell geometries. In the context of this thesis, 'robustness' is evaluated as the algorithmic stability of the planner when subjected to varying degrees of cell clutter and spatial constraints. Robustness against physical hardware disturbances such as workpiece displacement, base calibration offsets etc is designated for future testing.

2 Related work (Background)

Robot path planning is a mature research area including classical, sampling-based, and meta-heuristic approaches. While significant academic effort has been dedicated to algorithmic optimization, the practical configuration and geometric limitations of commercial path planners are not much explored. This chapter establishes the current state of the art (SOTA) in Collision Free Path Planning, reviews the sensitivity of underlying algorithms, and highlights the gap in evaluating commercial industrial tools.

2.1 State of the Art in Automatic Path Planning

The current state of the art in industrial robot programming is undergoing a major transition from manual teach in toward computer intelligence approaches. Dias et al. [2] provided a comprehensive review of this shift, detailing how the field has evolved from classical analytical methods to swarm optimization, evolutionary algorithms, and neural networks. Simultaneously, Jin et al. [3] established that for contact-intensive tasks in industrial automation, trajectory generation strategies are increasingly reliant on highly adaptable, learning-based models. Traditionally, robot trajectories were generated via manual teach-in or standard offline programming (OLP), both of which are highly time-consuming, operator-dependent, and inflexible when adapting to new product variants [8]. To meet the Industry 4.0 demands for flexible manufacturing and "Plug-and-Produce" automation, the field is shifting towards Automatic Path Planning (APP). In general, APP refers to software architectures that automatically derive collision-free robot motions directly from digital CAD models and cell layouts, minimizing or entirely removing the need for human programming intervention [3].

The algorithms driving SOTA APP systems are predominantly sampling-based methods, such as Rapidly-exploring Random Trees (RRT) and Probabilistic Roadmaps (PRM), due to their ability to efficiently handle high-dimensional, non-convex spaces [1]. Recent studies have successfully demonstrated the industrial viability of general APP frameworks across diverse applications. The industrial necessity of these tools is demonstrated by Larsen et al. [12], who utilized APP to drastically reduce the programming time for cooperating robots. Similarly, Ramasamy et al. [13] established the practical viability of APP in a Plug-and-Produce cutting-tool application, demonstrating that automated systems can reliably generate collision-free trajectories to handle frequent production changes.

The State of the Art Gap

Despite these advancements, a fundamental gap remains in the SOTA regarding the operational reliability of APP software. As noted by Juříček et al. [14] and Shisialia et al. [15], the vast majority of research focuses on proposing new, custom-coded algorithms rather than evaluating how deployed APP systems actually behave under varied conditions. Modern APP tools require users to configure critical parameters (e.g.,

clearance margins, optimization limits, timeouts) and rely heavily on the geometric accuracy of the digital twin. However, the SOTA literature lacks systematic empirical data on how sensitive these automated planners are to their own configuration settings or to spatial bottlenecks. While the literature thoroughly documents how to build APP algorithms, it fails to address how engineers should configure them to prevent failures in highly constrained industrial cells. This thesis addresses this general SOTA gap by systematically evaluating the configuration sensitivity of a widely deployed commercial representative of this technology (ABB APP).

2.2 Algorithmic Sensitivity to Parameter Configuration

The algorithms that power commercial APP tools inherently rely on user-defined configuration parameters to dictate their search behaviour. Zhuang et al. [16] demonstrated that traditional RRT algorithms suffer from redundant sampling and low efficiency in complex environments, leading them to propose an Adaptive Field Cooperative Sampling (AFCS) algorithm to improve environmental adaptability. Similarly, Secil and Ozkan [1] proposed an optimization-based local path planning method, confirming that planning performance is highly sensitive to configuration choices such as step size and sample counts. And, Yu et al. [17] demonstrated that standard RRT variants are highly sensitive to parameter initialization, noting that incorrect step sizes lead to redundant nodes and inconsistent success rates. While these studies identify parameter sensitivity as a core algorithmic weakness, their evaluations focus exclusively on altering custom algorithm code under idealized conditions. This thesis treats the commercial algorithm as a fixed "black-box" system, investigating how its performance degrades when user-defined parameters are varied.

This parameter sensitivity extends to multi-objective and meta-heuristic algorithms. Shinde and Pawar[18] evaluated a Fuzzy Particle Swarm Optimization (FPSO) method, showing that configuration choices cause measurable differences in convergence and solution quality. Wang et al. [19] and Jin et al. [5] both conducted hyperparameter sensitivity analyses on multi-robot planners, confirming that algorithm performance is intrinsically tied to parameter tuning. And, Kumar and Sikander [6] demonstrated that success rates in hybrid frameworks vary significantly purely by altering the sample count. However, all of these studies focus on custom algorithm tuning and do not systematically analyze how configuration parameter variations affect the reliability of existing commercial tools in a 3D industrial cell. This thesis addresses this limitation by establishing a parameter configuration methodology specifically designed for commercial software.

The interaction between configuration choices and environment layout is examined by Ramasamy et al. [20], [21], who implemented an Adaptive RRT* (ARRT*) algorithm for constrained industrial environments. Their work demonstrates that configuration parameters such as zone values and speed along with obstacle density, measurably affect both path quality and planning time. Building on this evidence that configuration

choices dictate planning efficiency, the present thesis systematically varies these parameters across changing cell layouts to identify explicit failure boundaries. While their study focused on optimizing specific algorithm parameters for performance, this work systematically varies configuration parameters across changing cell layouts to identify explicit failure boundaries.

2.3 Planning in Constrained and Complex Geometries

Beyond the internal parameters, the spatial layout and geometric complexity of the industrial robot cell play a decisive role in planning success. Zhang et al.[22] highlighted the difficulties of applying randomized planners in complex environments. By testing an improved RRT algorithm in narrow, maze-like spaces, they demonstrated that environmental constraints cause standard sampling-based algorithms to suffer from high failure rates. Fu et al. [23] observed similar degradation when evaluating an improved A* algorithm and Zhang et al.[22] observed degradation when evaluating the Fast-IBi-RRT* algorithm noting that environmental complexity directly increases planning failure risk. However, their work evaluates algorithm efficiency under fixed simulation conditions and does not investigate how varying these spatial bottlenecks affects a commercial planner. This thesis addresses this limitation by systematically altering the robot cell layout from unconstrained to highly restricted to empirically map how spatial bottlenecks bound APP configuration parameters.

The strict influence of geometric constraints is further supported by studies addressing collision detection and constraint modelling. Zhang et al. [24] proposed a collision avoidance strategy based on virtual body deformation, demonstrating that collision outcomes are highly sensitive to joint angle values and obstacle geometry modeling. Similarly, Zhou et al. [4] established that tight safety margins strictly trade off against planning efficiency and Liu et al. [25] demonstrated how sensor mounting configurations determine feasibility in welding. While these studies confirm that spatial restrictions limit operational freedom, they do not establish empirical failure thresholds or configuration guidelines. This thesis extends this concept by quantifying the exact physical clearance limits that determine whether an automated planning query will succeed or fail.

The specific challenge of dense, cluttered industrial environments is documented by Milloch et al. [9], who investigated automated path planning for control cabinet wiring. They reported that high obstacle density resulted in dangerous clearances and planning failures. Likewise, Chen et al. [8] showed that geometric simplification (e.g., using tessellated CAD models) significantly impacts path reliability in spray-painting applications. Although these studies provide evidence that path planning reliability drops in cluttered environments, they do not analyze which configuration settings contribute to these failures. This thesis fills this gap by mapping the exact geometric thresholds that cause planning failure, providing engineers with systematic configuration guidelines for heavily constrained cells.

2.4 The Gap in Evaluating Commercial Path Planning Tools

Despite decades of research, the literature remains focused on theoretical algorithm development rather than the practical evaluation of existing industrial software. Shisialia et al. [15] conducted a systematic literature review, explicitly identifying "poor testing and validation" as a major open challenge in the field. The theoretical limitations of these commercial tools were explored by Radhakrishnan and Gueaieb [26], [27], who established that model-based methods, the algorithmic family that underpins commercial planners like APP are inherently dependent on the accuracy of the represented configuration space. However, their work remains entirely theoretical. This thesis builds upon this foundation by supplying the missing empirical data, systematically quantifying how configuration choices and geometric constraints dictate the success of an industrial planner.

Finally, while some studies utilize commercial simulation environments to validate custom path planning algorithms, they rarely evaluate the built-in planning software itself. For example, Ramasamy et al. [20] utilized ABB RobotStudio to measure computational time and energy consumption, successfully quantifying how zone values and speed affect path quality. Similarly, Zhang et al. [22] investigated 7-DOF redundant manipulators to show that the feasible solution space can become entirely empty under certain constraint combinations. However, evaluating the inherent sensitivity of the commercial APP tool natively provided by ABB remains unaddressed. By isolating configuration parameters and geometric constraints using a One-Factor-At-A-Time (OFAT) methodology, this thesis provides the empirical characterization and decision-support framework currently missing from the academic literature.

Summary of the Research Gap

In conclusion, the reviewed literature extensively covers how path planning algorithms can be structurally improved to handle complex geometries and optimize performance. However, there is a clear absence of research treating commercial path planners as fixed systems to systematically measure their sensitivity to user-defined configurations and spatial bottlenecks. Because industrial engineers rely on tools like ABB's APP rather than custom-coded algorithms, this lack of empirical data forces commissioning decisions to be based on trial-and-error. By isolating configuration parameters and geometric constraints using a One-Factor-At-A-Time (OFAT) methodology, this thesis provides the empirical characterization and decision-support framework currently missing from the academic literature

3 Method

This chapter describes the scientific methodology applied to investigate the configuration sensitivity of ABB's Automatic Path Planning (APP) system. It outlines the research approach, the experimental design, the metrics used for data collection, and the strategies employed to ensure the reliability of the results.

3.1 Research Approach

To answer the research questions regarding parameter sensitivity and geometric constraints, a quantitative, simulation-based experimental approach was selected. Rather than developing or modifying a path planning algorithm at the source-code level, this study treats the commercial APP software as an industrial "black-box" system. By systematically manipulating the inputs (configuration parameters and cell geometry) and measuring the outputs (planning efficiency and success), the study isolates the practical effects of configuration choices. A simulation-based approach was deemed highly appropriate and beneficial, as it allows for the rapid, safe, and controlled iteration of spatial constraints that would be time-consuming and hazardous to reproduce on physical industrial hardware.

3.2 Experimental Design (OFAT)

The core of the methodology relies on a One-Factor-At-A-Time (OFAT) experimental design. The APP system features multiple interacting configuration variables. To accurately characterize the influence of each variable without confounding effects, the OFAT method requires establishing a fixed baseline configuration. During each test cycle, a single independent variable is swept across a defined range while all other parameters are locked to their baseline values.

The primary benefit of the OFAT approach in this context is its ability to identify explicit failure boundaries. By pushing a single parameter until the planner fails (e.g., increasing minimum obstacle clearance until no path can be found), the methodology mathematically identifies the geometric limitations of the cell. Once the individual parameter behaviors are mapped, they are synthesized into multi-parameter configuration profiles to evaluate system trade-off.

3.3 Data Collection and Evaluation Metrics

To provide objective decision-support for industrial commissioning, the performance of the APP system must be quantified using standard industrial metrics. Data collection was automated via Python scripts communicating with the CFree path planning server. For every planning query, the following Key Performance Indicators (KPIs) were extracted directly from the server's API response object:

- **Planning Time (ms):** The total computational time required by the server to return a collision-free path. This metric evaluates the computational efficiency of the parameter setting, which is critical for online, real-time trajectory generation. While this study focuses on offline planning, computational efficiency remains critical. Modern automated factories often generate hundreds of path variants, so inefficient parameters can delay offline programming by hours.
- **Path Length (rad):** The total joint-space distance of the generated path. In sampling-based joint-space planners, path length is not measured as the Cartesian distance of the Tool Center Point (TCP) in millimeters. Instead, it is calculated as the cumulative sum of the angular distances (in radians) traversed by all robot joints along the path. This metric serves as a proxy for physical path efficiency and cycle time. Minimizing this joint-space distance reduces mechanical wear and directly improves cycle time. Lower path lengths indicate a more highly optimized motion.
- **Success Rate (%):** A binary pass/fail metric indicating whether the planner successfully generated a valid path or returned an error (e.g., ERR_START_INVALID or ERR_GOAL_INVALID). This metric evaluates the stability and robustness of the configuration against geometric constraints.

3.4 Tools, Reliability, and Error Margins

The experiments were conducted using ABB RobotStudio as the digital twin environment and the CFree Python API to interface with the APP backend solver. To represent a standard industrial application, a collaborative robot (CRB1100) was utilized for a representative pick-and-place task.

A recognized challenge in evaluating sampling-based path planners (such as the RRT variants underlying the APP system) is their inherent stochasticity. Because these algorithms rely on random sampling of the configuration space, identical planning queries can yield slightly different path lengths and computation times across multiple executions. To account for this variance and ensure statistical reliability, every individual parameter configuration was executed 20 times. The logged data was then averaged, and any failed runs were excluded from the performance averages to prevent infinite error values from distorting the time and length. Across all 20 runs, path length showed negligible variance (standard deviation < 0.02 rad) due to the deterministic behaviour of CFree's shortcutting heuristic, while the computation times shows standard deviations within 2 – 5 ms. Because these spreads were minor, mean values are reported in subsequent summary tables to evaluate macro trends.

3.5 Workflow of the Study

To ensure reproducibility and a logical progression of the experiments, the study was conducted in four distinct chronological phases:

1. **Phase 1: Environment and Pipeline Setup.** The digital twin of the robot cell was modelled in ABB RobotStudio. The geometrical data was exported as Path Planning Data (.ppd format) to the CFree server, and a Python-based communication pipeline was established to automate the planning queries and log the results into CSV files.

2. **Phase 2: Parameter Identification.** Unstructured interviews and consultations were held with ABB path-planning developers and Field Service Engineers. This phase filtered out hardware dependent and task-altering parameters, and thereby shortlisting the core variables for evaluation.
3. **Phase 3: OFAT Execution.** The One-Factor-At-A-Time testing was executed. A single parameter was swept across a defined range while the others remained at the baseline. This phase identified the absolute geometric boundaries and failure thresholds of the parameters.
4. **Phase 4: Profile Synthesis and Visualisation.** The limits discovered in Phase 3 were used to synthesize three practical configuration profiles (Speed, Balanced, and Quality). These profiles were tested across three different cell layouts, and the resulting KPIs were normalized to generate Color-Coded Decision Matrices for industrial commissioning.

4 Experimental Setup and Implementation

This chapter details the practical execution of the methodology described in Chapter 3. It defines the specific configuration parameters tested, the geometric environments created, and the formulation of the decision-support profiles. Few importantPython scripts used to automate the testing and data logging are provided in Appendix A.

4.1 Parameter Identification and Exclusion

The ABB Automatic Path Planning (APP) software exposes numerous configuration parameters to the user. To determine which parameters to evaluate, a systematic filtering process was conducted. Throughout this process, the official ABB APP Application Manual served as the primary guiding document, supplemented by expert elicitation from ABB Line Builders and the core APP software development team.

The initial parameter universe considered for this study included: num_planning_threads, path_type (MoveJ vs. MoveL), multiple_goals, orientation_constraint, min_distance_to_obstacles, optimality_effort, and timeout. To maintain a focused study on geometric sensitivity, several parameters were systematically excluded:

- **Hardware and Task Parameters:** As stated in the APP manual, num_planning_threads dictates CPU multi-threading. This was excluded because it is a hardware-dependent metric, not a geometric one. Similarly, multiple_goals and goal_regions were excluded because modifying them fundamentally changes the task definition, rather than testing the planner's sensitivity to a fixed task.
- **Kinematically Restrictive Constraints:** The orientation_constraint parameter was initially tested to enforce a rigid tool posture. However, the representative pick-and-place task required a tilted intermediate position relative to the start and end targets. Following consultation with ABB path-planning experts (M. Åkerblad, personal communication, August 2026), it was confirmed that enforcing strict orientation constraints on tilted paths inherently forces the feasible configuration space to an empty set, rendering the problem mathematically unsolvable. Consequently, this parameter was excluded from the OFAT sweeps.

Definition of Evaluated Parameters
Following these exclusions, three core parameters governing computational effort and geometric safety were shortlisted for systematic testing:

1. **min_distance_to_obstacles:** Defines the strict global safety clearance (in meters). The planner guarantees that no part of the robot's kinematic chain will come closer to any modeled obstacle than this specified value.

2. **optimality_effort:** A user-defined value ranging from 0.0 to 1.0 that dictates the computational resources allocated to path optimization. A value of 0.0 instructs the algorithm to return the first collision-free path it discovers. Values greater than 0.0 instruct the algorithm to spend additional computational time running post-processing passes (such as node-pruning and shortcutting) to shorten the path.
3. **timeout:** The absolute maximum time (in milliseconds) the solver is permitted to execute. If the algorithm cannot find a valid path within this limit, the process is forcefully terminated.

4.2 Simulated Industrial Environments

To quantify how spatial constraints bound the effectiveness of the planning parameters, three distinct simulation environments were constructed in RobotStudio. The core task across all environments remained identical: a Collaborative robot picking an object from the tray on a conveyor belt and placing it on a tray on a workstation table. The environments were named Tight, Moderate, and Open to represent the standard spectrum of industrial obstacle density.

The geometric constraints around this task were systematically varied:

- **Station A (Tight):** A highly restricted environment where the robot is closely surrounded by structural walls and a densely placed workstation. This represents a worst-case industrial layout where path options are severely limited by dense obstacles.
- **Station B (Moderate):** A partially constrained environment created by changing the obstacle position and expanding the distance of the side walls. This setup provides the planner with more configuration space while retaining standard cell boundaries.
- **Station C (Open):** An unconstrained environment where all walls were removed and obstacles are placed further away. This idealized setup serves as a control environment to observe algorithm performance strictly based on the distance between the robot and the task objects (table and conveyor).

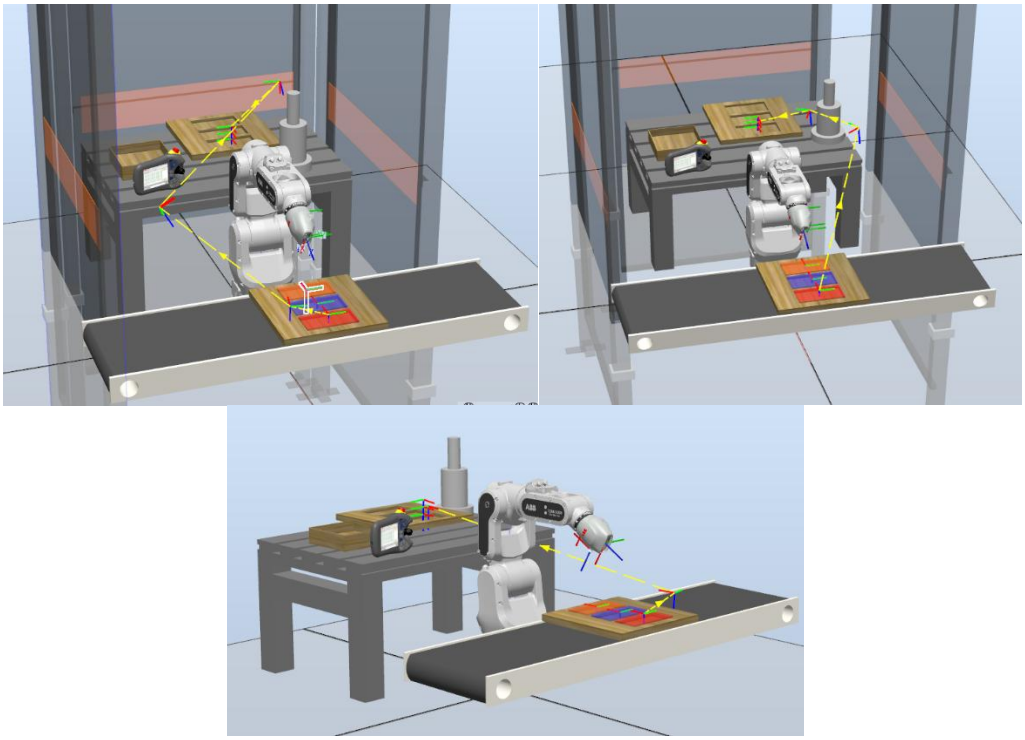


Figure 1 The three simulated industrial environments: (a) Station A: Tight, (b) Station B: Moderate, and (c) Station C: Open, illustrating the varying degrees of geometric constraints.

To mimic realistic industrial path planning, the start and goal targets were defined as approach and retract points hovering immediately above the objects (e.g., table and conveyor). This guarantees that the start and end states are fundamentally collision-free, ensuring that any subsequent planning failures are the result of the spatial bottlenecks encountered along the path. Additionally, based on expert feedback, the goal target's kinematic configuration was left mathematically free (cfg=None). This allowed the APP inverse kinematics solver to automatically evaluate all valid joint postures and select the one resulting in the most efficient path.

Collision Geometry Representation

While RobotStudio provides the high-fidelity digital twin for the user, the underlying APP algorithm relies on simplified mathematical representations of these objects to perform collision checking. When the station data (.ppd) is exported to the CFree Python API, the physical objects are translated into discrete collision bodies (e.g., convex hulls and bounding meshes).

Visualizing the environment through the Python API confirms that the algorithm calculates safety margins based strictly on these collision geometries. This emphasizes the critical nature of the min_distance_to_obstacles parameter, as any inaccuracies between the physical world and these digital collision bodies can lead to undetected physical collisions during deployment.

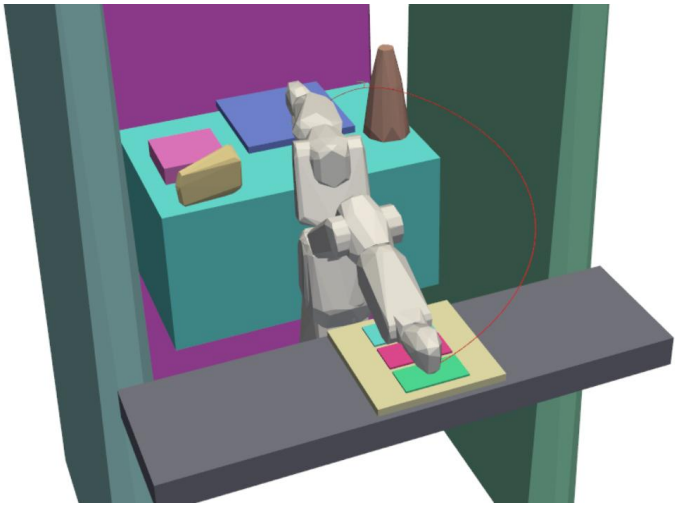


Figure 2: Visualization of the collision geometries and the generated path using the CFree Python API. The algorithm evaluates clearances between the robot's geometric links and the surrounding obstacle meshes.

4.3 OFAT baseline configuration

To systematically execute the One-Factor-At-A-Time (OFAT) parameter sweeps, a fixed baseline configuration was established. This baseline represents a standard, responsive industrial planning scenario. The fixed values were selected based on manufacturer documentation (ABB Automatic Path Planning Application Manual) and practical industrial limits:

- **Optimality Effort (0.1):** The ABB manual recommends a value of 0.1 for applications where computation time is limited. This was chosen as the baseline to ensure the planner operates quickly, preventing excessive optimization times from masking the effects of the geometric constraints.
- **Minimum Distance to Obstacles (0.002 m):** The software's absolute mathematical minimum limit is 1 mm to avoid numerical collision-checking errors. A baseline of 2 mm (0.002 m) was selected as the tightest practical safety margin, allowing the robot maximum manoeuvrability in the constrained cell without triggering false mathematical collisions.
- **Timeout (1000 ms):** In typical industrial operations, generating a single point-to-point motion should not introduce significant latency. A 1-second (1000 ms) timeout was chosen as a strict but realistic upper bound for evaluating planner efficiency.

During the execution of the OFAT tests, one parameter was varied across a defined testing range (e.g., sweeping optimality_effort from 0.0 to 1.0) while the remaining parameters were locked to the baseline values. The sweep ranges were determined as follows:

- **min_distance_to_obstacles (0.001 m to 0.025 m):** The upper bound (25 mm) was selected because visual inspection indicated the approach targets were hovering less than 20 mm from the surfaces. Sweeping up to 25 mm ensured the capture of the exact threshold where geometric failure occurs.

- **optimality_effort (0.0 to 1.0):** The sweep was designed to capture the absolute minimum (0.0), the manual-recommended default (0.1), a midpoint (0.5), and the absolute maximum (1.0).
- **timeout (50 ms to 3000 ms):** The range spans from an aggressive kill-switch threshold (50 ms) to a conservative offline allowance (3000 ms).

4.4 Formulation of Configuration Profiles

While the OFAT sweeps successfully identify the isolated mathematical limits of each parameter, commissioning engineers rarely configure parameters in isolation. To translate the raw sensitivity data into an actionable industrial decision-support tool, the findings from the parameter sweeps were synthesized into three distinct Configuration Profiles. These profiles represent common, competing priorities in manufacturing environments:

1. **Speed-Oriented Profile**

   Real-time, online trajectory generation where minimizing computational latency is the absolute priority, often at the expense of physical path efficiency. Target users could be Integrators developing online, dynamic applications (e.g., vision-guided picking), where the robot must calculate paths on the fly. Computational latency is the primary bottleneck.

- optimality_effort: **0.0 – 0.1** (Bypasses deep optimization to return the first valid path).
- timeout: **50 – 100 ms** (Acts as an aggressive kill-switch, aborting the process quickly if a geometric bottleneck makes the path impossible, thereby preventing the robot controller from freezing).
- min_distance_to_obstacles: **< *D*** (Where *D* is the physical distance between the robot's target approach point and the resting surface. Must be kept safely below *D* to prevent start/goal validation errors).

2. **Balanced Profile**

   Standard offline cell commissioning, providing a middle ground between calculation speed and path length. Target users could be general commissioning engineers who require a reliable, responsive default setting while testing standard cell layouts.

- optimality_effort: **0.1 – 0.3** (Triggers sub-millisecond path smoothing without incurring the heavy computational penalties of deep optimization).
- timeout: **300 – 1000 ms** (A standard industrial fallback limit).
- min_distance_to_obstacles: **< *D***

The recommended optimality effort range of 0.1 to 0.3 represents an engineering interpolation between the measured baseline at 0.1 (which triggers immediate path smoothing) and the saturation threshold observed at 0.5, offering an operational middle ground without causing heavy optimization overhead.

3. Quality-Oriented Profile

High-volume mass production (e.g., automotive assembly) where minimizing the physical path length is critical for cycle-time reduction, and high computation time during the offline programming phase is acceptable. Target users are integrators configuring high-volume mass production, where minimizing the physical path length is critical for cycle-time reduction.

- optimality_effort: **0.4 – 0.5** (Provides maximum path optimization. Values above 0.5 are explicitly excluded as they drastically increase computation time with no further improvement to path length).
- timeout: **1000+ ms** (Grants the solver maximum allowable time to optimize the trajectory).
- min_distance_to_obstacles: **Maximised up to *D*** (Provides the largest possible safety margin around the tool without triggering an ERR_START_INVALID failure).

These three profiles were subsequently tested across all three simulated stations (Tight, Moderate, and Open) to generate the final performance trade-off matrices

5 Results and discussion

This chapter presents the empirical data collected from the automated planning queries across the three simulated environments. The first section details the isolated parameter behaviors discovered during the OFAT sweeps. The second section presents the synthesized Configuration Profiles and decision-support visualisations. Finally, the results are analysed in the context of industrial deployment and the theoretical framework.

5.1 OFAT Parameter Sensitivity Results

The One-Factor-At-A-Time (OFAT) parameter sweeps successfully identified the operational boundaries and scaling behaviours of the ABB Automatic Path Planning (APP) software.The raw data generated during these sweeps is summarized in Table 1, illustrating the bounds and performance variations within the Moderate Station.

Table 1 Summary of OFAT Parameter Sweeps (Moderate Station)

Parameter Swept	Values tested	Min Computation time	Max computation time	Path length range	Success rate
Minimum distance	1, 2, 5, 10, 15, 25 mm	11 ms	29 ms	3.56 – 3.73 rad	Fails at ≥ 10 mm
Optimality Effort	0.0, 0.1, 0.5, 1.0	12 ms	240 ms	3.47 – 3.81 rad (0.1 : 3.57 rad, 0.5/1.0: 3.47 rad)	100%
Timeout	50, 100, 300, 1000, 3000 ms	12 ms	14 ms	3.56 rad (Static)	100%

Minimum Distance to Obstacles (min_distance_to_obstacles)
Across all three stations (Tight, Moderate, and Open), the planner maintained a 100% success rate when min_distance_to_obstacles was set between 1 mm and 5 mm. However, in all three stations, setting the parameter to 10 mm or higher resulted in a 100% failure rate, consistently throwing an ERR_START_INVALID error. This uniform failure occurred because the starting and ending approach targets were physically located less than 10 mm above the resting surfaces. It is important to emphasize that while the 10mm numerical failure boundary is specific to the approach clearance (*D* ~approx 8-9 mm) of this particular station, the principle behind it that the global clearance cannot exceed the local physical contact gap *D* is universal across all robot planning tasks. The data demonstrates that global clearance parameters are strictly bounded by the physical contact geometry of the specific task (denoted as *D*), regardless of the free space available in the rest of the cell.

Optimality Effort (optimality_effort)
Varying the optimality_effort parameter revealed a distinct non-linear relationship between computation time and path efficiency, as summarized across the range in Table 1.

Across the evaluated range (3.57 to 3.81 rad), the planner showed sharp diminishing turns. Setting effort to 0.0 produced an initial path of 3.81 rad in 12 ms. Increasing effort to 0.1 improved the path to 3.57 rad while remaining at 12 ms computational time. Because the algorithm executes a rapid line of sight shortcutting pass at 0.1, the path is significantly smoothed without any measurable penalty to planning speed.

However, pushing the effort higher showed rapid saturation. At an effort of 0.5, the path length shortened to 3.47 rad with an average computation time of 75 ms. Increasing the effort further from 0.5 to 1.0 caused the computation time to increase significantly to 240 ms, yet yielded zero additional path reduction and the path length remained unchanged at 3.47 rad. This demonstrates that for this cell geometry, an effort of 0.5 serves as the practical performance ceiling. Also, it allocates computational budget beyond 0.5 results in computational waste.

Timeout (Early Convergence)
The OFAT sweeps for the timeout parameter confirmed that the APP solver is highly efficient. Across the baseline scenarios, the planner consistently converged on a solution in under 20 ms. Consequently, increasing the offline timeout parameter from 50 ms to 3000 ms had no measurable difference effect on either the computation time or the path length

5.2 Configuration Profile Trade offs

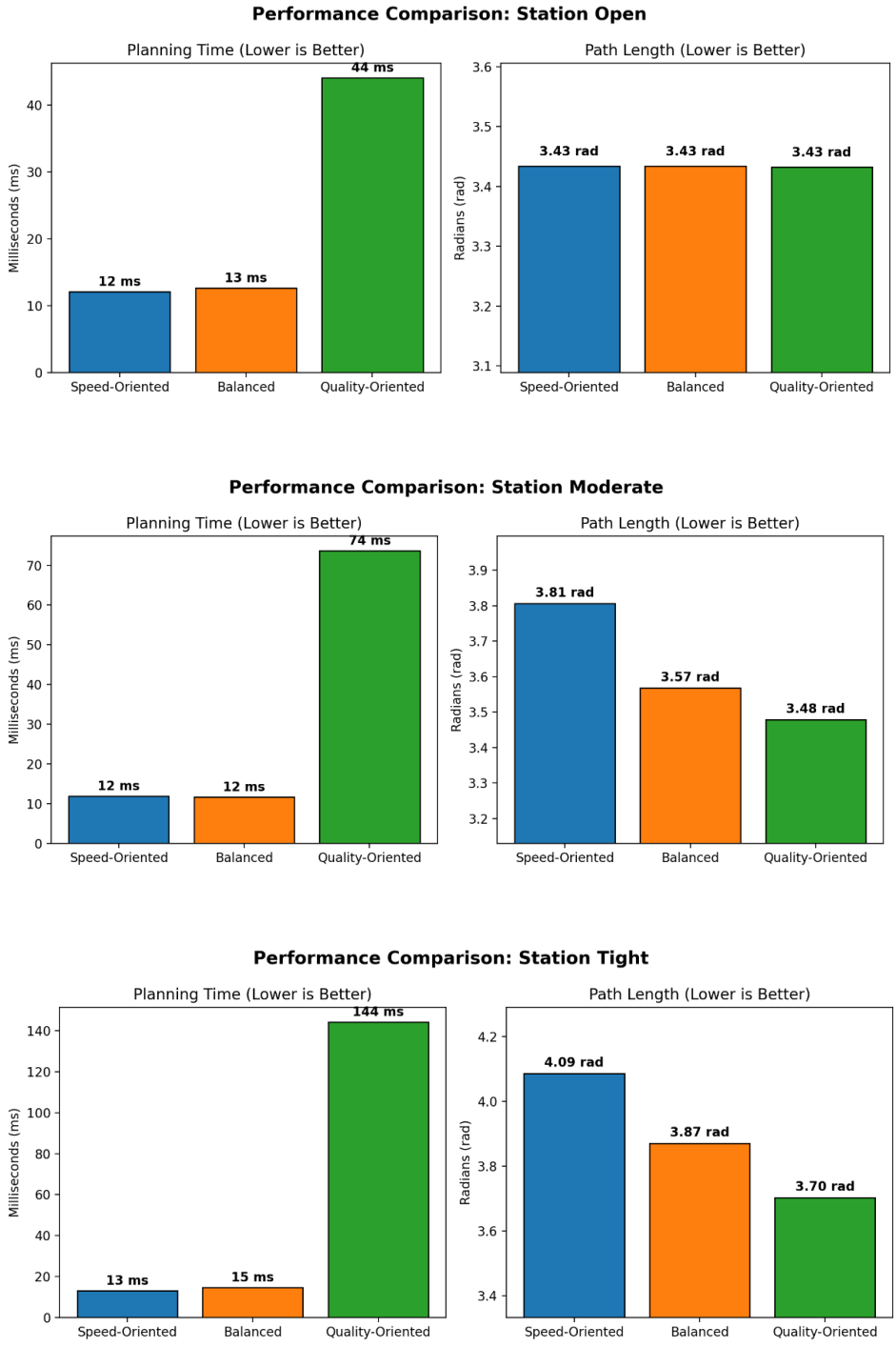
To evaluate the practical trade-offs between planning speed and path efficiency, the three Configuration Profiles (Speed-Oriented, Balanced, and Quality-Oriented) were executed across the three stations. Because the OFAT sweeps purposefully bounded the parameters within safe limits (e.g., keeping min_distance $\leq$ 5 mm), all profiles achieved a 100% reliability (success) rate. Therefore, the comparative analysis focuses strictly on the trade-off between planning time and path length. The resulting performance metrics are documented in Table 2.

Table 2: Profile Performance Across Simulated Environments (Averaged over 20 runs)

Station	Profile	Planning Time ms	Path length (rad)	Success rate
Open	Speed – Oriented	12 ms	3.43 rad	100%
Open	Balanced	13 ms	3.43 rad	100%
Open	Quality Oriented	44 ms	3.43 rad	100%
Moderate	Speed – Oriented	12 ms	3.81 rad	100%
Moderate	Balanced	12 ms	3.57 rad	100%
Moderate	Quality Oriented	74 ms	3.48 rad	100%
Tight	Speed – Oriented	13 ms	4.09 rad	100%
Tight	Balanced	15 ms	3.87 rad	100%
Tight	Quality Oriented	144 ms	3.70 rad	100%

To understand the operational trade-offs, the performance of the three profiles must be analyzed concurrently. As illustrated in the Performance Bar Charts (Figure 3), an inverse relationship exists: achieving a marginal reduction in path length requires an exponential increase in computation time.

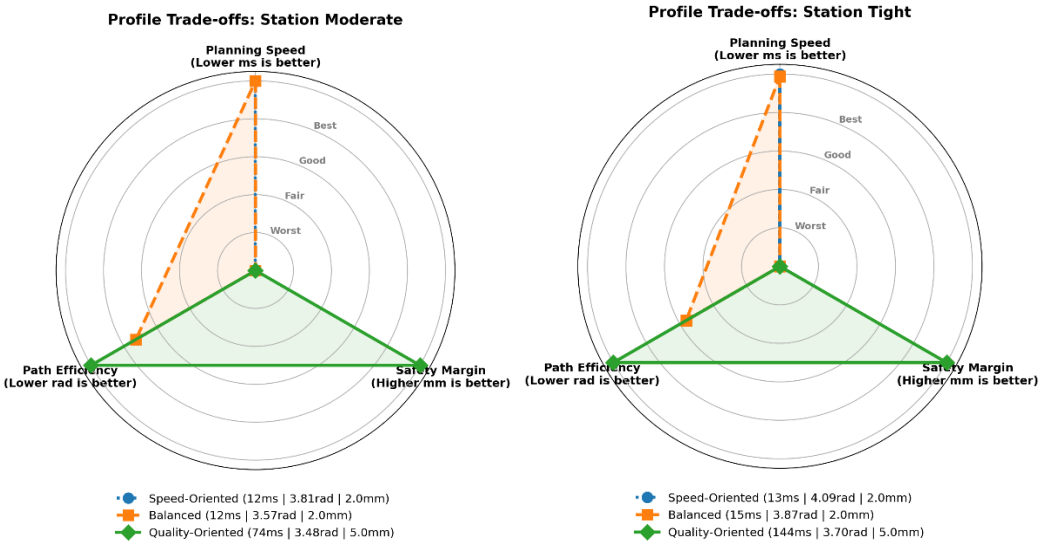
As shown in Figure 3, the physical layout of the cell directly determines the computation time. Using the same Quality-Oriented profile, the computation time scaled exponentially as environmental clutter increased: taking 44 ms in the Open station, 74 ms in the



Figur 3: Performance comparisons illustrating the relationship between planning time (ms) and path length (rad) across the three environments.

Moderate station, and 144 ms in the Tight station. This quantifies the heavy computational burden of collision-checking in densely populated industrial cells. Furthermore, the geometric "squeeze" on the robot is evident in the optimal path lengths: the best achievable path length increased from 3.43 rad (Open) to 3.48 rad (Moderate), up to 3.70 rad (Tight), as the walls forced the manipulator to take wider detours.

To provide a holistic visual representation of the competing priorities, the profile performances were also plotted on 3-axis radar charts (Spider Webs), as shown in Figure 4. Because the OFAT sweeps proved that all profiles achieved 100% reliability within their bounded limits, the traditional "Success Rate" axis was replaced with "Safety Margin" to provide a more descriptive third metric. The safety margin represents the physical clearance utilized by the profile (e.g., 2 mm versus 5 mm).



The radar charts instantly visualize the "shape" of the operational trade-offs. The Speed-Oriented profile stretches fully toward the Planning Speed axis but retracts significantly on Path Efficiency and Safety Margin. Conversely, the Quality-Oriented profile maximizes both Safety and Path Efficiency but collapses entirely on the Planning

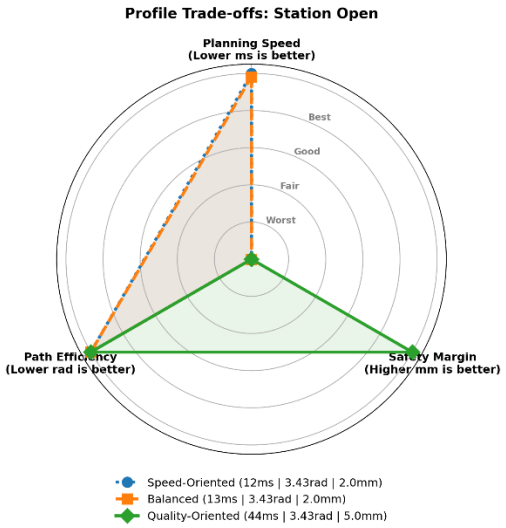


Figure 4: Spider web diagrams illustrating the multi-dimensional trade-offs between Planning Speed, Path Efficiency, and Safety Margin. Distinct line styles and markers are utilized to highlight overlapping profiles

Speed axis. Plotting the data in this format allows integrators to visually assess the profile shapes against their factory's specific operational needs at a single glance.

Interpreting the Radar Charts
To utilize these charts as a practical decision support tool, the axes and profile shapes must be interpreted as follows:

- **The Scale (Center vs. Outer Edge):** The data is Min-Max normalized so that the outermost edge of the web always represents the "Best" or most optimal outcome in that specific cell, while the center point represents the "Worst."
- **Planning Speed (Lower is Better):** A profile reaching the outer edge on this axis indicates the fastest computation time (eg: 12 ms).
- **Path Efficiency (Lower is Better):** A profile reaching the outer edge indicates the shortest physical path (e.g.: 3.47 rad).
- **Safety Margin (Higher is Better):** A profile reaching the outer edge utilizes the maximum safe clearance (e.g.: 5 mm) without triggering a geometric failure.
- **Overlapping Geometries:** If two profiles share the same line or marker on a specific axis (e.g.: Speed and Balanced on the Planning Speed axis in the Moderate station), it indicates that both configurations deliver identical performance for that metric.

5.3 Guidelines for Industrial Commissioning

To ensure the raw data serves as a practical tool for industrial users, Color-Coded Decision Matrices (Figure 3) were generated. To resolve the differing scales of measurement (milliseconds vs. radians), Min-Max normalization was applied to the raw data, converting the metrics into standardized scores to generate the colour gradients (Green = Best, Red = Poorest).

The Color-Coded Decision Matrices translate these metrics into a practical decision-support guide. It clearly visualize extreme cases of computational waste. For example, in the Open station, the Quality-Oriented profile increased planning time by 264% compared to the Speed-Oriented profile, but yielded a 0% improvement in path length (all profiles achieved 3.43 rad). This visually proves to the commissioning engineer that high optimality efforts should be strictly avoided in unconstrained cells, as the fastest valid path is already kinematically optimal.

Configuration Guide : Station Tight

Profile	Planning Speed (ms)	Path Efficiency (Length,rad)
Speed Oriented [dist: < D | opt_eff: 0.0 - 0.1 | timeout: 50-100ms]	13 ms (BEST)	4.09 rad (+10.4%)
Balanced [dist: < D | opt_eff: 0.1 - 0.3 | timeout: 300-1000ms]	15 ms (+12%)	3.87 rad (+4.5%)
Quality-Oriented [dist: Max up to D | opt_eff: 0.4 - 0.5 | timeout: 1000ms+]	144 ms (+1006%)	3.70 rad (BEST)

Color Scale: Green = Optimal Trade-off | Red = Poorest Trade-off

Configuration Guide : Station Moderate

Profile	Planning Speed (ms)	Path Efficiency (Length,rad)
Speed-Oriented [dist: < D | opt_eff: 0.0 - 0.1 | timeout: 50-100ms]	12 ms (+2%)	3.81 rad (+9.4%)
Balanced [dist: < D | opt_eff: 0.1 - 0.3 | timeout: 300-1000ms]	12 ms (BEST)	3.57 rad (+2.6%)
Quality-Oriented [dist: Max up to D | opt_eff: 0.4 - 0.5 | timeout: 1000ms+]	74 ms (+530%)	3.48 rad (BEST)

Color Scale: Green = Optimal Trade-off | Red = Poorest Trade-off

Configuration Guide : Station Open

Profile	Planning Speed (ms)	Path Efficiency (Length,rad)
Speed-Oriented [dist: < D | opt_eff: 0.0 - 0.1 | timeout: 50-100ms]	12 ms (BEST)	3.43 rad (BEST)
Balanced [dist: < D | opt_eff: 0.1 - 0.3 | timeout: 300-1000ms]	13 ms (+5%)	3.43 rad (BEST)
Quality-Oriented [dist: Max up to D | opt_eff: 0.4 - 0.5 | timeout: 1000ms+]	44 ms (+264%)	3.43 rad (BEST)

Color Scale: Green = Optimal Trade-off | Red = Poorest Trade-off

Figure 5: Configuration Decision Matrices providing normalized trade-off scores and recommended parameter ranges for industrial commissioning.

The following step-by-step guideline outlines how commissioning engineers could utilize these matrices:

Step 1: Identify the Physical Bottleneck (D)
Before setting parameters, the engineer should measure the tightest physical clearance the robot must navigate. In this study, this was the distance between the tool and the table at the approach target (denoted as D). The min_distance_to_obstacles parameter must never exceed D, or the planner will return an ERR_START_INVALID error.

Step 2: Choose the Production Priority
The user selects a row in the Decision Matrix based on the goal of their specific cell (e.g., Speed-Oriented for online planning, Quality-Oriented for offline mass production).

Step 3: Apply the Parameter Ranges
Once a priority is selected, the engineer inputs values within the mathematically validated ranges (shown directly under the profile name in the matrix) into the APP configuration settings.

5.4 Discussion and Industrial implications

The results of this study directly answer the research questions regarding how commercial planning tools respond to configuration changes and geometric constraints. The empirical data confirms that parameter tuning is not a trivial optimization step, but rather a critical requirement for successful industrial deployment.

Generalization and Problem-Dependency
As confirmed through consultation with ABB path-planning experts, the exact influence of APP parameters is highly problem-dependent. Because universal configuration numbers (e.g., a "perfect" minimum distance) cannot be established for all industrial cells, the methodology developed in this thesis proves essential. Rather than relying on trial-and-error, engineers can use the established OFAT profiling methodology to systematically map the geometric boundaries of their specific cell and generate application-specific decision matrices.

Advanced Configuration Recommendations (The D-Value Override)
The OFAT results demonstrated that global safety margins (min_distance_to_obstacles) are strictly limited by the distance (D) between the tool and the pick/place surfaces. While the baseline methodology correctly bounded the parameter below D to prevent ERR_START_INVALID failures, this forces the entire robot to operate with a minimal safety margin just to accommodate the final pick location. To mitigate this in industrial practice, it is highly recommended that integrators utilize object-specific parameter overrides. By using the min_distance_at_touch setting exclusively for the table or target object, an engineer can enforce a larger, safer global clearance (e.g., 20 mm) for the entire cell, while allowing the robot to safely penetrate that margin solely at the required interaction point.

Addressing the Literature Gap
The findings support the theoretical framework established by Radhakrishnan and Gueaieb[26],[27] , which argued that model-based planners are fundamentally sensitive to configuration space restrictions. However, this thesis advances the literature by providing the missing empirical quantification of a commercial tool. The study proves that while algorithms can be highly capable in abstract scenarios, their real-world industrial implementation requires structured, geometry-aware parameter configuration to balance efficiency, safety, and cycle time effectively.

6 Conclusion

The objective of this study was to evaluate how configuration parameters and geometric constraints affect the reliability and efficiency of ABB's Automatic Path Planning (APP) system. Based on the simulation experiments and the resulting data, all aims and research questions formulated in the introduction have been answered.

Regarding the first research question, the data clearly identifies which settings have the greatest influence. The min_distance_to_obstacles parameter has the greatest influence on stability, and if it is set incorrectly relative to the cell geometry, it causes a 100% planning failure. Also, the optimality_effort parameter has the greatest influence on performance. Pushing the optimality effort beyond 0.5 tripled the computation time without reducing the path length. Additionally, leaving the goal target's kinematic orientation unconstrained (cfg = None), allowing the inverse kinematics solver to select the optimal arm quadrant rather than enforcing a rigid tool posture along the entire path was necessary to find a valid collision free solution.

Regarding the second research question (the effect of geometric constraints), the results prove that planner performance is heavily bounded by the physical layout of the cell. The maximum allowable safety margin (min_distance) is strictly limited by the distance between the tool and the resting surface at the pick/drop targets. Furthermore, the computation time for the exact same parameter profile scaled from 44 ms in an open environment to 144 ms in a highly constrained environment, quantifying how collision-checking efforts increase with obstacle density.

Ultimately, because APP behaviour is highly problem-dependent, there are no universally optimal settings. Instead of relying on default values or trial-and-error, engineers can use the systematic OFAT methodology and the resulting Decision Matrices developed in this thesis to make informed, data-driven configuration choices during commissioning.

6.1 Future Work and Research

This thesis focused entirely on nominal simulation environments to establish a baseline for parameter configuration. It also isolated internal software parameters, excluding external deployment errors to establish a clean, controlled simulation baseline before introducing physical hardware uncertainties. The natural next step for future research is to investigate how these optimized configurations perform when transferred to physical hardware. Future studies should introduce controlled deployment errors into the simulation, such as robot base shifts, TCP calibration inaccuracies, and fixture tolerances, to measure how much real-world geometric deviation a planned path can tolerate before a physical collision occurs. Additionally, evaluating the APP system in dynamic environments or multi-robot setups would provide further insight into the software's industrial limits.

Also, investigating how different mesh decomposition levels and voxel grid sizes interact with the optimality_effort and min_distance thresholds will provide a more comprehensive guide for real-world automated planning. Additionally, evaluating the APP system in dynamic environments or multi-robot setups would provide further insight into the software's industrial limits.

6.2 Critical Discussion

During the initial phase of this project, the goal was to test physical robustness and deployment errors. However, as the work progressed, it became clear that the baseline behaviour of the APP parameters was not systematically documented in the existing literature. It was impossible to test how real-world errors affected the planner without first understanding how the planner behaved under ideal conditions. Fixing the focus to configuration sensitivity and developing a parameter tuning methodology was a necessary step.

A limitation of this work is that it relies entirely on a perfect digital twin and the assumption that the simulation geometry perfectly matches the real world is rarely true. As detailed in the ABB APP manual, the representation of obstacles, specifically the density of convex hull decompositions (meshes) and the resolution of point clouds (voxels) drastically impacts collision-checking computations. Because this study utilized a fixed geometric resolution, the interaction between parameter sensitivity and variable mesh fidelity is an unexplored limitation.

6.3 Generalization of the result

The specific numerical results obtained in this study, such as the 144 ms computation time or the 3.70 rad path length, are unique to the CRB1100 robot and the specific pick-and-place geometry used in the test cell. Therefore, the raw data cannot be directly applied to different robots or different tasks.

However, the methodology itself can be generalized. The process of using OFAT parameter sweeps to identify geometric bottlenecks, followed by the generation of a Color-Coded Decision Matrix, can be applied to any robotic cell in ABB RobotStudio. Also, this evaluation approach could be extended to other commercial path-planning tools to create standardized benchmarking practices across the robotics industry. From a practical industrial perspective, commissioning engineers face strict time constraints and cannot be expected to run automated Python sweeps to configure every new robot cell. Therefore, the ultimate value of this methodology is not for the end-user to execute directly, but for software developers and advanced integrators to establish standardized presets. By embedding validated profiles directly into graphical interfaces, end-users can select a deployment strategy based on their high-level goals, while the backend software applies the empirically proven parameter limits.

7 References

[1] S. Secil and M. Ozkan, "A collision-free path planning method for industrial robot manipulators considering safe human–robot interaction," *Intell. Serv. Robot.*, vol. 16, no. 3, pp. 323–359, Jul. 2023, doi: 10.1007/s11370-023-00465-7.

[2] P. A. Dias *et al.*, "Robot Path Planning: from Analytical to Computer Intelligence Approaches," *J. Intell. Robot. Syst.*, vol. 111, no. 4, p. 112, Oct. 2025, doi: 10.1007/s10846-025-02322-4.

[3] S. Jin, K. Ke, B. Gao, L. Fu, and X. Huang, "A Comprehensive Review of Key Technologies for Robot Motion Planning in Contact Tasks in Industrial Automation Scenarios," *Chin. J. Mech. Eng.*, vol. 38, no. 1, p. 198, Nov. 2025, doi: 10.1186/s10033-025-01356-x.

[4] M. Zhou, J. Li, C. Wang, J. Wang, L. Wang, and V. Puig, "A distributed multi-robot collaborative collision avoidance hunting method under probabilistic uncertainty framework," *Sci. China Inf. Sci.*, vol. 69, no. 2, p. 122207, Feb. 2026, doi: 10.1007/s11432-024-4534-2.

[5] J. Jin, M. Wu, P. Fan, and G. Jin, "APPA: An adaptation path planning algorithm for real-time obstacle avoidance in multi-robot systems," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 37, no. 7, p. 194, Sep. 2025, doi: 10.1007/s44443-025-00218-9.

[6] S. Kumar and A. Sikander, "A novel hybrid framework for single and multi-robot path planning in a complex industrial environment," *J. Intell. Manuf.*, vol. 35, no. 2, pp. 587–612, Feb. 2024, doi: 10.1007/s10845-022-02056-2.

[7] L. Zhang *et al.*, "A Path Planning Method Based on Optimal Arm Angle of 7-Degrees-of-Freedom Manipulator," *J. Intell. Robot. Syst.*, vol. 111, no. 2, p. 50, Apr. 2025, doi: 10.1007/s10846-025-02246-z.

[8] Heping Chen, T. Fuhlbrigge, and Xiongzi Li, "Automated industrial robot path planning for spray painting process: A review," in *2008 IEEE International Conference on Automation Science and Engineering*, Arlington, VA: IEEE, Aug. 2008, pp. 522–527. doi: 10.1109/COASE.2008.4626515.

[9] E. Milloch, M. Brisse, F. Vacha, and B. Kuhlenkötter, "Automated Path Planning for Control Cabinet Wiring with an Industrial Robot," in *2025 11th International Conference on Control, Automation and Robotics (ICCAR)*, Kyoto, Japan: IEEE, Apr. 2025, pp. 142–150. doi: 10.1109/ICCAR64901.2025.11073012.

[10] H. Zhang, Y. Wang, J. Zheng, and J. Yu, "Path Planning of Industrial Robot Based on Improved RRT Algorithm in Complex Environments," *IEEE Access*, vol. 6, pp. 53296–53306, 2018, doi: 10.1109/ACCESS.2018.2871222.

[11] A. Alexander, K. Venkatesan, J. Mounsef, and K. Ramanujam, "A Comprehensive Survey of Path Planning Algorithms for Autonomous Systems and Mobile Robots: Traditional and Modern Approaches," *IEEE Access*, vol. 13, pp. 176287–176326, 2025, doi: 10.1109/ACCESS.2025.3618837.

[12] L. Larsen, M. Kaspar, A. Schuster, M. Vistein, J. Kim, and M. Kupke, "Full automatic path planning of cooperating robots in industrial applications," in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, Xi'an: IEEE, Aug. 2017, pp. 523–530. doi: 10.1109/COASE.2017.8256157.

[13] S. Ramasamy, X. Zhang, M. Bennulf, and F. Danielsson, "Automated Path Planning for Plug & Produce in a Cutting-tool Changing Application," in *2019 24th*

IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain: IEEE, Sep. 2019, pp. 356–362. doi: 10.1109/ETFA.2019.8869398.

[14] M. Juříček, R. Parák, and J. Kůdela, "Evolutionary Computation Techniques for Path Planning Problems in Industrial Robotics: A State-of-the-Art Review," *Computation*, vol. 11, no. 12, p. 245, Dec. 2023, doi: 10.3390/computation11120245.

[15] S. Shisiali, E. A. Mathews, O. Calvins, and O. James, "A Systematic Literature Review of Path-Planning Strategies for Robot Navigation in Unknown Environment," vol. 48, no. 1.

[16] Y. Zhuang *et al.*, "Path planning of industrial robots based on the adaptive field cooperative sampling algorithm," *Front. Neurorobotics*, vol. 19, p. 1574044, Nov. 2025, doi: 10.3389/fnbot.2025.1574044.

[17] Q. Yu, J. Zhou, and Z. Xue, "Path planning of manipulator considering obstacle avoidance based on improved RRT algorithm," *J. Braz. Soc. Mech. Sci. Eng.*, vol. 47, no. 10, p. 505, Oct. 2025, doi: 10.1007/s40430-025-05820-6.

[18] V. B. Shinde and P. J. Pawar, "Motion planning of multi degree serial manipulator with dynamic constraints using fuzzy particle swarm optimization," *Int. J. Interact. Des. Manuf. IJIDeM*, vol. 20, no. 3, pp. 1709–1722, Mar. 2026, doi: 10.1007/s12008-025-02482-5.

[19] C.-H. Wang, K. Hu, and X. Wu, "Multi-robot path planning in online dynamic obstacle environments based on parallel cooperative strategy optimization algorithm," *Discov. Comput.*, vol. 28, no. 1, p. 132, Jul. 2025, doi: 10.1007/s10791-025-09664-5.

[20] S. Ramasamy, K. Eriksson, S. Peralippatt, B. Perumal, and F. Danielsson, "Optimized Online Path Planning Algorithms Considering Energy," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vasteras, Sweden: IEEE, Sep. 2021, pp. 01–08. doi: 10.1109/ETFA45728.2021.9613457.

[21] S. Ramasamy, K. Eriksson, B. Perumal, S. Peralippatt, and F. Danielsson, "Optimized Path Planning by Adaptive RRT* Algorithm for Constrained Environments Considering Energy," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vasteras, Sweden: IEEE, Sep. 2021, pp. 1–8. doi: 10.1109/ETFA45728.2021.9613699.

[22] M. Zhang, S. Liu, Q. Zhou, and X. Han, "A novel path planning scheme based on Fast-IBi-RRT* algorithm for industrial robots," *Appl. Intell.*, vol. 55, no. 11, p. 785, Jul. 2025, doi: 10.1007/s10489-025-06694-w.

[23] B. Fu *et al.*, "An improved A* algorithm for the industrial robot path planning with high success rate and short length," *Robot. Auton. Syst.*, vol. 106, pp. 26–37, Aug. 2018, doi: 10.1016/j.robot.2018.04.007.

[24] L.-X. Zhang, X.-J. Meng, and Z.-J. Ding, "Collision avoidance strategy based on virtual body deformation for path planning of serial industrial robot," *J. Mech. Sci. Technol.*, vol. 38, no. 6, pp. 3113–3129, Jun. 2024, doi: 10.1007/s12206-024-0530-1.

[25] J. Liu, Y. Li, X. Jia, Y. Wu, and T. Li, "Path planning method of welding robot based on point cloud scene understanding," *Int. J. Adv. Manuf. Technol.*, vol. 141, no. 1–2, pp. 1049–1060, Nov. 2025, doi: 10.1007/s00170-025-16662-w.

[26] S. Radhakrishnan and W. Gueaieb, "A state-of-the-art review on topology and differential geometry-based robotic path planning—part I: planning under static

constraints," *Int. J. Intell. Robot. Appl.*, vol. 8, no. 2, pp. 435–454, Jun. 2024, doi: 10.1007/s41315-024-00330-5.

[27] S. Radhakrishnan and W. Gueaieb, "A state-of-the-art review on topology and differential geometry-based robotic path planning—part II: planning under dynamic constraints," *Int. J. Intell. Robot. Appl.*, vol. 8, no. 2, pp. 455–479, Jun. 2024, doi: 10.1007/s41315-024-00331-4.

A. Appendix

```
1:          #The following Python script illustrates the core
#experimental pipeline used to interface with the ABB CFree server. It
#demonstrates how the Configuration Profiles were defined, how the goal
#kinematics were freed to allow optimization, and how the resulting Key
#Performance Indicators (KPIs) were extracted and logged.
2:
3:
4:
5:          import cfree_py as cf
6:          import csv
7:          import pathlib
8:
9:          # 1. Define the Configuration Profiles
10:         PROFILES = {
11:             "Speed-Oriented": {"min_dist": 0.002, "opt_eff": 0.0,
"timeout": 100},
12:             "Balanced":         {"min_dist": 0.002, "opt_eff": 0.1,
"timeout": 1000},
13:             "Quality-Oriented": {"min_dist": 0.005, "opt_eff": 0.5,
"timeout": 3000}
14:         }
15:
16:         N_RUNS = 20
17:         STATION_PATH   =   pathlib.Path.home()   /   "Documents"   /
"Station_B_Moderate.ppd"
18:         OUT_FILE = "profile_results_log.csv"
19:
20:         # Initialize CSV log file
21:         with open(OUT_FILE, "w", newline="") as f:
22:             csv.writer(f).writerow(["Profile",   "Run",   "Status",
"Time_ms", "Path_Length_rad"])
23:
24:         # 2. Connect to the CFree Server and Load Station
25:         with cf.CFreePyClient(run_server_process=False) as cfree:
26:             cfree.import_station(STATION_PATH)
27:
28:             # 3. Define Targets using RAPID strings
29:             start_rapid         =         "[[501.24,         23.33,
310.63],[0.0,0,1,0],[0,0,-1,1],[9E9,9E9,9E9,9E9,9E9,9E9]]"
30:             goal_rapid         =         "[[-439.83,         50.54,
452.47],[0,1,0,0],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]]"
31:
32:             start = cf.RobTarget.from_rapid_string(start_rapid)
33:             goal = cf.RobTarget.from_rapid_string(goal_rapid)
34:
35:             # Free the goal configuration to allow IK optimization
36:             goal.cfg = None
37:
38:             # 4. Execute Planning Queries
39:             for profile_name, settings in PROFILES.items():
40:                 for i in range(N_RUNS):
```

```
41:
42:                     # Apply the specific profile parameters
43:                     params = cf.PlanningParameters(
44:
min_distance_to_obstacles=settings["min_dist"],
45:                         optimality_effort=settings["opt_eff"],
46:                         timeout=settings["timeout"]
47:                     )
48:
49:                     # Send query to ABB APP solver
50:                     query = cf.PlanningQuery(start=start, goal=goal,
params=params)
51:                     result = cfree.plan_path(query)
52:
53:                     # 5. Extract KPIs and Log Data
54:                     status = result.status.name
55:                     if status == "OK":
56:                         time_ms = result.metrics.t_total
57:                         path_len = result.metrics.path_length
58:                     else:
59:                         time_ms, path_len = 0.0, 0.0
60:
61:                     with open(OUT_FILE, "a", newline="") as f:
62:                         csv.writer(f).writerow([profile_name,     i,
status, time_ms, path_len])
```