POLYPUBLIE
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Title:	A Collaborative Robotic Framework for Aerospace Drilling: the Design of a Smart Drilling End-Effector and an Adaptive Drilling System
Auteur: Author:	Aghiles Yousfi
Date:	2026
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Yousfi, A. (2026). A Collaborative Robotic Framework for Aerospace Drilling: the Design of a Smart Drilling End-Effector and an Adaptive Drilling System [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/73600/

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/73600/
Directeurs de recherche: Advisors:	Sofiane Achiche, Abolfazl Mohebbi, & Amir Hajzargarbashi
Programme: Program:	Génie mécanique

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

A Collaborative Robotic Framework for Aerospace Drilling: the Design of a Smart Drilling End-Effector and an Adaptive Drilling System

AGHILES YOUSFI
Département de génie mécanique

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie mécanique

Janvier 2026

© Aghiles Yousfi, 2026.

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

A Collaborative Robotic Framework for Aerospace Drilling: the Design of a Smart Drilling End-Effector and an Adaptive Drilling System

présenté par **Aghiles YOUSFI**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Luc BARON, président
Sofiane ACHICHE, membre et directeur de recherche
Abolfazl MOHEBBI, membre et codirecteur de recherche
Amir HAJZARGARBASHI, membre et codirecteur de recherche
Oguzhan TUYSUZ, membre

DEDICATION

I dedicate this work to my parents

Je dédie ce travail à mes parents

ACKNOWLEDGEMENTS

First and foremost, my heartfelt appreciation goes to my research advisors, Amir Hajzargarbashi, Sofiane Achiche and Abolfazl Mohebbi. I am grateful for their invaluable guidance, support and encouragement during the project. I am also grateful for the financial support they provided through funds from the NRC and CRSNG. A special thanks to Amir for the trust he placed in me to work on this project and for how he welcomed me at the AMTC. I am proud to be a researcher learning and growing under their supervision.

I would also like to thank my colleagues at the lab for their good spirit and the enjoyable work environment they have established, as well as for the great times spent together. I cannot not mention those unfinishable *UNO* games we had at each dinner.

My gratitude also extends to the NRC technicians, machinists, and researchers who contributed to what was achieved in this research project. A special thanks to Mahdi Kazemiesfahani, Erfan Dilfanian and Hamed Hosseini for their generous assistance offered whenever it was needed.

I would also like to thank professors Luc Baron and Oguzhan Tuysuz for agreeing to be part of the jury for this master thesis.

The writing experience would not have been the same without the company provided by my two cats, Nala and Lua.

Finally, I am forever grateful to my parents, without whom I would not be where I am now.

RÉSUMÉ

Le Conseil National de Recherche Canadien (CNRC) a exploré l'idée d'utiliser des robots collaboratifs pour des opérations de perçage dans le domaine aérospatial. Celles-ci sont souvent réservées à de grands robots industriels, car les robots collaboratifs n'ont pas la rigidité nécessaire pour ces tâches qui impliquent l'application d'une force de compression sur la pièce qui dépasse leurs capacités structurelles. Le CNRC a développé un exosquelette qui protège les joints de la base du robot des efforts excessifs qui leur sont transmis lors du perçage, contribuant à une augmentation de la rigidité du cobot. Cependant, cette solution n'est qu'un pas vers un cadre complet et déployable de perçage cobotique.

La question de recherche derrière ce mémoire est de déterminer à quelle mesure un cobot muni d'une solution de rigidification peut percer des trous de qualité, et dans quelle mesure ces trous peuvent être surveillés avec précision. Le but de ces travaux étant de développer un cadre pour le perçage cobotique, cet objectif principal peut être atteint à travers la conception d'un effecteur intelligent pour le perçage cobotique ainsi que la conception d'un système de perçage adaptatif pour les métriques de qualité de trou (diamètre, cylindricité, concentricité et *runout*).

L'effecteur multi-fonctions présenté dans ce document permet d'appliquer la force de compression nécessaire de 600 N pour le perçage aérospatial, ainsi que le contrôle et la mesure de la force de compression et de la vitesse d'avance de l'outil de perçage. Un système de vision pour le perçage cobotique a également été conçu et intégré à l'effecteur afin de détecter les trous pilotes avec une précision de 3.84 mm sur la pièce, ainsi que l'orientation de la normale de surface avec une erreur d'orientation moyenne de 0.75 °.

L'autre partie de ce mémoire présente comment l'apprentissage machine est utilisé pour concevoir un système de surveillance pour le perçage cobotique. Cela impliquait l'acquisition de donnés de perçage sur plus de 300 trous, l'extraction et la sélection de marqueurs ainsi que l'entraînement d'un modèle de régression permettant de prédire les métriques géométriques de qualité de trou à partir de signaux opérationnels de force & vibration, avec des valeurs de coefficient de détermination R^2 sur l'ensemble de test de 0.75, 0.62, 0.36 et 0.1. Un classificateur de paramètres de procédés a également été entraîné afin de reconstruire les paramètres de procédés utilisés pour un trou afin de détecter la présence d'anomalies survenues durant le perçage, avec une précision de 84% pour la force de compression et 99% pour la vitesse d'avance. Ces deux modèles ont été intégrés dans un système de perçage adaptatif pouvant modéliser les paramètres futurs de procédé sachant les paramètres présents et les donnés de

perçage, en utilisant une table de recherche et un modèle statistique.

Les résultats obtenus nous permettent de conclure que seule une minorité de 37% des trous percés sont comparables en termes de qualité à ceux obtenus à partir d'un système rigide de perçage similaire et que seules les opérations de classification permettent de surveiller avec précision les trous percés. Le premier sous-objectif concernant la conception de l'effecteur à été atteint, mais le second concernant le système de perçage adaptatif reste incomplet vu les faibles performances du modèle de régression obtenues.

ABSTRACT

The National Research Council of Canada (NRC) has been investigating the possibility of adopting collaborative robots for aerospace drilling applications. This task is often reserved for heavy industrial robots, because collaborative robots lack the rigidity for such tasks: the clamping force required to be applied to the workpiece exceeds the cobot's capabilities. The NRC has built a collaborative robot exoskeleton to protect the base cobot joints from the extensive load caused by drilling, increasing the cobot's rigidity. However, this solution is only a step towards a complete and deployable cobotic drilling framework, as the current cobotic drilling system lacks elements such as a tailored smart drilling end-effector and an adaptive drilling system.

This master thesis investigates how effectively a cobot equipped with a rigidity-enhancement solution can drill holes and how accurately these holes can be monitored. The main objective is to design the foundational elements for a cobotic drilling framework. The foundational elements presented are related to the design of a smart cobotic drilling end-effector and an adaptive drilling system.

This document presents a multifunctional smart end-effector design capable of applying the required 600N clamping force for aerospace drilling tasks and includes the sensing and control of the clamping force and feed speed. A vision system for cobotic drilling was also integrated into the end-effector to detect pilot holes with an accuracy of 3.84 mm on a workpiece as well as the surface's normal orientation at the drilling site with an average orientation error of 0.75°.

The other part of this master thesis presents how machine learning was utilized to design a monitoring system for cobotic drilling. This involved the operational drilling data acquisition of over 300 holes, feature extraction and selection, and training a regression model to predict some of the hole's geometric quality metrics from operational force and vibration signals, with determination coefficient R^2 values on the test set of 0.75, 0.62, 0.36 and 0.1. A process parameters classifier was also trained to reconstruct the process parameters used for a hole to detect whether an anomaly occurred during the drilling process, achieving an accuracy of 84% for clamping force and 99% for feed speed. These two models were then integrated into an adaptive drilling system that could model subsequent drilling parameters given prior parameters and drilling data, based on a lookup table and a statistical model.

The results obtained lead to the conclusion that only 37% of holes drilled by the cobotic system have quality similar to those drilled using an equivalent rigid drilling system, and only

the process monitoring classifier can accurately monitor the drilled holes. The first secondary objective concerning the end-effector design has been achieved, but the second regarding the adaptive drilling system could not be fully completed due to the limited performance of the regression model.

TABLE OF CONTENTS

DEDICATION iii

ACKNOWLEDGEMENTS iv

RÉSUMÉ v

ABSTRACT vii

LIST OF TABLES xi

LIST OF FIGURES xii

LIST OF SYMBOLS AND ACRONYMS xv

LIST OF APPENDICES xvii

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 LITERATURE REVIEW 3
- 2.1 Introduction 3
- 2.2 Non cobotic drilling in the industry 3
- 2.3 Cobots in manufacturing 4
  - 2.3.1 Collaborative robots 4
  - 2.3.2 Payload and stiffness limitations 4
  - 2.3.3 Cobotic drilling in the industry 5
- 2.4 Drilling process control 5
- 2.5 Dynamic analysis and modeling 7
  - 2.5.1 Modal analysis 7
  - 2.5.2 Hole quality prediction 7
- 2.6 Summary and positioning of this work 10

CHAPTER 3 OBJECTIVES AND RESEARCH QUESTION 13
- 3.0.1 Research Question 13
- 3.0.2 Research assumptions 13
- 3.0.3 Main Objective and Specific Objectives 13

CHAPTER 4 METHODOLOGY 15
- 4.1 Methodology Overview 15

4.2 Experimental Setup . . . 15
4.2.1 Exobrake presentation . . . 15
4.2.2 Drilling Test Bed . . . 17
4.2.3 Data Acquisition Material . . . 17
4.3 End-Effector Design and Prototyping . . . 19
4.3.1 TCP Positioning . . . 19
4.3.2 Overall Structure . . . 26
4.3.3 Sensors and actuators . . . 28
4.3.4 Vision System . . . 30
4.4 Adaptive Drilling . . . 38
4.4.1 Design of Experiment and Drilling Tests . . . 39
4.4.2 Data Pre-processing and Features Extraction . . . 43
4.4.3 Features Selection and Model Training . . . 45
4.4.4 Adaptive Drilling Algorithm . . . 52

CHAPTER 5 DEVELOPMENTS AND RESULTS . . . 57
5.1 Experimental Setup Validation . . . 57
5.2 End-Effector Developments . . . 57
5.2.1 TCP Positioning . . . 57
5.2.2 Structural Performances . . . 58
5.2.3 Sensor and Actuators . . . 61
5.2.4 Vision System Results . . . 64
5.3 Adaptive Drilling Results . . . 68
5.3.1 Design of Experiment and Drilling Tests . . . 68
5.3.2 Data Pre-processing and Features extraction . . . 74
5.3.3 Feature Selection and Model Training . . . 77
5.3.4 Adaptive Drilling algorithm . . . 84

CHAPTER 6 CONCLUSION . . . 87
6.1 Summary of works . . . 87
6.2 Limitations . . . 88
6.3 Future research . . . 90

REFERENCES . . . 91

APPENDICES . . . 95

LIST OF TABLES

Table 2.1	Summary of machine learning-based approaches for prediction and monitoring in drilling and machining tasks	11
Table 4.1	Joint torque limits of the UR10e	23
Table 4.2	Measurement requirements and their adapted sensor	29
Table 4.3	Drilling sections coordinates on the drilling test bench	40
Table 4.4	Description of the extracted signal features	44
Table 4.5	Process lookup table containing each clamping (C) and feed speed (D) groups from their 3 levels (low, mid, high)	55
Table 5.1	ALCCE parts identification from Figure 5.1	59
Table 5.2	Normality detection results. The orientation error $\Delta\alpha$ and the number of candidate points used N are presented for each hole	66
Table 5.3	Per axis accuracy and precision results of the different hand-eye calibration methods	68
Table 5.4	Process parameters values corresponding to each level	68
Table 5.5	Relative slippage results of the 5 best robot configurations	72
Table 5.6	Most significant features based on p-value of the diameter metric used as initial feature set in the filter based method	78
Table 5.7	Average R^2 prediction performances when selecting the features based on their p-value	78
Table 5.8	Average R^2 performances and final features number of the wrapper methods during feature selection, on the full and reduced features sets	80
Table 5.9	R^2 hole quality prediction performances all 4 metrics using the backward feature selection v1 method	81
Table 5.10	Average accuracy results of the best classification models for the Clamping, Feed speed and Multiclass scenarios, considering both the full and reduced feature set	83
Table 5.11	Specification Limits used for the adaptive drilling system	84
Table 5.12	Per-metric and overall success probability given the process parameters recipes	85
Table B.1	Complete slippage results with robot configurations	96

LIST OF FIGURES

Figure 2.1	ALEMA end-effector [1]	3
Figure 2.2	Three widely used commercial collaborative robots: (a) Universal Robot UR10e, (b) Fanuc CRX-10iA, (c) Kinova Gen3	4
Figure 2.3	Crawling robot held by a cobot for drilling application [2]	5
Figure 2.4	KUKA LBR iiwa with its drilling end effector in front of a drill jig [3]	6
Figure 2.5	Cobot helping the operator to handle the drilling tool [4]	6
Figure 2.6	Delamination model for drilling, with the backup plate from [5]	8
Figure 4.1	Exobrake CAD representation with its joints labelled	16
Figure 4.2	Drilling Test Bed	18
Figure 4.3	Sensors used for data acquisition: (a) Kistler dynamometer (b) PCB accelerometer	19
Figure 4.4	Data acquisition system (a) Data acquisition elements (b) Wiring schematic	20
Figure 4.5	Laser tracking material (a) Polaris VegaXT optical tracking system (b) Detectable target, from the datasheet	20
Figure 4.6	Transformation between the flange frame and the TCP frame	21
Figure 4.7	Double curvature drilling pattern	22
Figure 4.8	Spherical Coordinate system used for TCP orientation	22
Figure 4.9	Effect of ψ on the robot pose, from *Simulink*'s visualizer	22
Figure 4.10	UR10e flange connection drawing, from the robot's datasheet [6]	26
Figure 4.11	Drilling unit's integrated Hydraulic damper and Linear potentiometer for manual feed speed control and feed measurements	27
Figure 4.12	UR10e tool payload curve, from the robot's datasheet [6]	28
Figure 4.13	Workpiece test image used for the surface normality detection. The marks have later been removed on *Paint*	31
Figure 4.14	Example of a robot pose used to capture an image of the calibration grid during hand-eye calibration	34
Figure 4.15	Example of calibration grid pictures taken by the camera, used for the hand-eye calibration procedure (a) Image 1 (b) Image 6	35
Figure 4.16	Example of detected dots pattern on the calibration grid	36
Figure 4.17	Rigid drilling setup used to record benchmark drilling data	40

Figure 4.18 Example of candidate robot configurations for slippage minimization in RDK with labelled test bench sections. (a) Config. 1 (elbow and wrist down) on section 1 (b) Config. 32 (elbow down, wrist up) on section 6 41
Figure 4.19 Optical tracking system's targets emplacements for slippage measurements during clamping tests, on robot Config. 1 and the reference frame used 42
Figure 4.20 Filter-based method ML pipeline 46
Figure 4.21 Representation of the dataset subdivision for cross-validation using 5 folds 47
Figure 4.22 Adaptive drilling algorithm architecture 53
Figure 5.1 The designed ALCCE end-effector's exploded view. The parts are labelled and identified in Table 5.1. 59
Figure 5.2 End-effector parts designed with lightening holes: (a) Main frame (2), (b) Loadcell holder (4) 60
Figure 5.3 Joints affected by bending efforts on versions (a) 2.0 and (b) 3.0 of the end-effector 61
Figure 5.4 Example of the laser sensor's square signal used for rotation speed measurement. $t_{\uparrow}$ is the period, measured between two rising edges . . 61
Figure 5.5 Example of tool rotation frequency measurement through FFT analysis 62
Figure 5.6 End-effector cut view 62
Figure 5.7 *Ecolubric* Portable MQL system used for drilling 63
Figure 5.8 Stereo camera on the end-effector 64
Figure 5.9 Configuration change for camera capture, from the *RoboDK* model (a) Drilling configuration (b) Camera capture configuration 65
Figure 5.10 Holes detection on the test image. (a) Bounding box set by the user. (b) Detected holes. In green the nominal hole and in blue the augmented hole. 65
Figure 5.11 Pointcloud of the test workpiece. The black lines are the normal vectors and the smaller frames around them are the candidate points used for normality detection 66
Figure 5.12 Calibration grid 3D projection. (a) Dots pattern projected on the fitted plane with the camera frame and the grid's frame. (b) Projected dots pattern along with the original pointcloud 67
Figure 5.13 Camera frame position on the EE after hand-eye calibration. 67

Figure 5.14 Low feed speed level position and speed profiles during a drilling test, measured by the linear potentiometer 69
Figure 5.15 Raw and filtered speed profiles at low feed speed level 70
Figure 5.16 Axial force measured by the dynamometer and the load cell during a high clamping force drilling test of Hole 38 in Plate 1. 70
Figure 5.17 Per-axis target's position during clamping test #26, recorded by the Polaris VegaXT tracker 71
Figure 5.18 Peak feed dots drilled for absolute slippage evaluation. (a) Optimal configuration (b) Non-Optimal configuration. 73
Figure 5.19 Workpiece #3 top sheet, fully drilled by Exobreak with drill bit #2. Each hole is labelled for CMM measurements 73
Figure 5.20 Tool wear seen on a microscope and measurements. (a) Unused tool (b) after 100 holes: 18.1μ m (c) after 200 holes: 59.4μ m 74
Figure 5.21 Tool wear manually measured on Figure 5.20b 74
Figure 5.22 Time domain raw axial force recorded by the dynamometer during hole 38 in plate 1 75
Figure 5.23 Frequency domain raw axial force recorded by the dynamometer during hole 38 in plate 1 75
Figure 5.24 Time domain filtered axial force recorded by the dynamometer during hole 38 in plate 1 76
Figure 5.25 Frequency domain filtered axial force recorded by the dynamometer during hole 38 in plate 1 77
Figure 5.26 Training and CV performances progression of the version 1 forward feature selection algorithm, during run #6 79
Figure 5.27 Training and CV performances progression of the version 2 forward feature selection algorithm, during run #6 80
Figure 5.28 Training and CV performances progression of the version 2 backward feature selection algorithm, during run #6 81
Figure 5.29 Training and CV performances progression of the version 1 backward feature selection algorithm, during run #6 82
Figure 5.30 Training and CV performances progression of the version 1 JMI forward feature selection algorithm, during run #6 83
Figure 5.31 Example of a data point fed into the adaptive drilling system. We see the computed prediction interval, the predicted and true metric value and the specification limit for each hole quality metrics 85
Figure A.1 FEA analysis of the drilling structure with 600N of applied force . . . 95

LIST OF SYMBOLS AND ACRONYMS

NRC	National Research Council of Canada
FRF	Frequency Response Function
ML	Machine Learning
ALCCE	Aerospace Lightweight Collaborative Clamping End-Effector
CNC	Computer Numerical Control
CFRP	Carbon Fiber Reinforced Polymer
TCP	Tool Center Point
TCP/IP	Transmission Control Protocol/Internet Protocol
EE	End-Effector
MR	Magnetorheological
CoG	Center of Gravity
FFT	Fast Fourier Transform
MQL	Minimum Quantity of Lubricant
RANSAC	RANdom SAmple Consensus
CAD	Computer Assisted Drawing
PCD	PointCloud Data
PnP	Perspective-n-Point
RMS	Root Mean Square
DOF	Degrees of Freedom
API	Application Programming Interface
CMM	Coordinates Measuring Machine
RDK	RoboDK
TPF	Tooth Pass Frequency
PSD	Power Spectral Density
ANOVA	Analysis of variance
ANCOVA	Analysis of covariance
SVM	Support Vector Machine
RF	Random Forest
MLP	Multi-Layers Perceptron
LR	Logistic Regression
DT	Decision Tree
ET	Extended Trees
GB	Gradient Boost

LDA	Linear Discriminant Analysis
AMTC	Aerospace Manufacturing Technologies Center
DOE	Design Of Experiments
CV	Cross Validation

LIST OF APPENDICES

Appendix A Drilling structure FEA 95
Appendix B Drilling tests 96
Appendix C Wrapper-based feature selection algorithm pseudocodes 97

CHAPTER 1 INTRODUCTION

Drilling is one of the most important operation in aerospace manufacturing because each aircraft requires thousand of holes that must meet strict requirements in terms of geometric quality, clamping conditions, position and surface normality. Some of these holes are still produced manually or with automated equipment. Manual drilling is flexible but slow, physically demanding and prone to human error. On the other side, traditional automated drilling solutions based on CNC machines and large industrial robots deliver excellent quality, but require a considerable capital investment, dedicated tooling and fixtures and strict safety measures. These systems are optimized for high-volume tasks and are difficult to redeploy when the application scope changes.

Between these two solutions, intermediate solutions have been developed. Some approaches rely on portable drilling units guided by external jigs, structures held by operators or crawling mechanisms. Recent solutions involving cobots improve flexibility and safety, but require external jigs or constant operator supervision. The current landscape offers either rigid and expensive systems for high volume operations, or flexible labour intensive approaches that do not always guarantee consistent aerospace drilling quality.

Cobots are attractive in this context because they are an easier to deploy alternative, can safely share workspace with human operators and are well suited for tasks that require flexibility and frequent changeovers. However, their limited payload restrict their ability to perform structurally demanding tasks such as multi-stack aerospace drilling. To mitigate this limitation, the NRC developed Exobrake, which is a rigidity enhancement exoskeleton for a UR10e cobot. Exobrake redirects the load sensed by the base joints through its structure by using pneumatically activated hydraulic brakes, reducing the load transmitted to the protected joints and increasing the cobot's payload. The exoskeleton improves payload capabilities, but by itself, it does not provide all the elements needed for aerospace drilling: a dedicated drilling end-effector, including sensing and vision, and an adaptive drilling system suited for the cobot are still missing.

The problem addressed in this master thesis is the lack of an integrated cobotic framework that combines rigidity enhancement, a smart drilling end-effector and an adaptive monitoring and decision system for aerospace drilling. Existing work generally focus on one element at a time (mechanical rigidity, end-effector design or process monitoring) without presenting a coherent solution that can be integrated on a cobot for realistic drilling scenarios. This fragmentation makes it difficult to exploit the potential of cobots for drilling tasks that require

high quality and flexibility.

In this context, the overall goal of this master's thesis is to design and validate key building blocks of a collaborative robotic drilling framework for aerospace drilling, based on a UR10e cobot equipped with a rigidity enhancement exoskeleton. This work aims to develop a lightweight clamping end-effector with necessary sensing, clamping and actuation capabilities, and to design an adaptive drilling system that uses multi sensor data to monitor hole quality and suggest process parameter adjustments.

This master's thesis is organized as follows. Chapter 2 reviews the literature on automated and robotic drilling for aerospace applications, the use of cobots in general drilling tasks, solutions for rigidity enhancement, and machine learning approaches for manufacturing process monitoring, and finally positions the current work within this context. Chapter 3 formulates the research question and the main and specific objectives, along with the assumptions used to delimit the scope of the research. Chapter 4 presents the methodology, including the experimental setup, the design and prototyping of the smart cobotic drilling end-effector, the vision system and the pipeline used to develop the adaptive drilling system. Chapter 5 presents the developments and results, from the validation of the test bench and end-effector to the evaluation of the vision system and the adaptive drilling system. Finally, Chapter 6 summarizes the main contributions, discusses the limitations of the proposed framework, and presents directions for future work on cobotic drilling for aerospace manufacturing.

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction

In aerospace manufacturing, automated drilling is a key process because it is the main process used to assemble many panels and components to build an aircraft. What distinguishes the drilling task in aerospace can be summarized as follows: Multilayered curved surfaces requiring tight tolerances. Advances in robotics and mechatronics address the issues related to the task. However, the use of cobots for this specific task is still limited. The following literature review explores advances across domains like the use of cobots for manufacturing processes, present automated drilling solutions, dynamic analysis and process modeling and the use of machine learning in drilling tasks. Finally, a summary will wrap the literature review by addressing the gaps and the positioning of this master thesis.

2.2 Non cobotic drilling in the industry

Traditional automated drilling uses industrial robots or CNC machines. ALEMA [7] uses a Kuka robot with a multifunctionnal end-effector consisting of 5 modules: drill, fastener, hammer, vision and hole quality measurements (diameter). The vision module aligns the end-effector to the drilling point axis and creates a 3D map using the 2D image and the laser telemeter. A slipping correction system allows the robot to compensate for slippage that occurs during clamping force application.

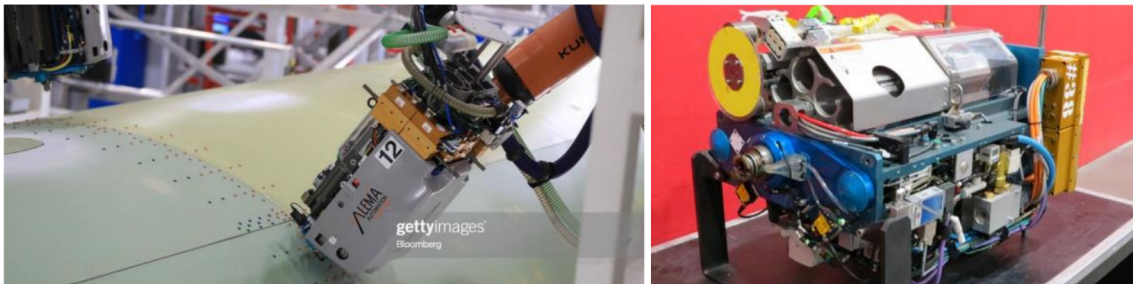


Figure 2.1 ALEMA end-effector [1]

Another example of large scale robotic drilling solution is the patent WO2021036588A1 [8]. It demonstrates a multifunctional drilling-riveting end-effector for industrial robots comprising a drilling/countersinking, riveting, inspection and clamping modules. Clamping is first applied on the surface, followed by drilling and countersinking. Then a swapping unit engages the

riveting module to insert and press the rivet in the drilled hole. The process is monitored by the inspection module.

The previously stated examples of non-cobotic drilling solutions that involve heavy industrial robots are expensive, harder to deploy because of their size and require extensive logistic needs (like specialized operators) and strict safety measures, as stated earlier.

2.3 Cobots in manufacturing

2.3.1 Collaborative robots

Collaborative robots are a type of industrial manipulators designed to share the workspace with humans. Unlike traditional robots, which operate at high speeds and forces, cobots are built with safety features, such as torque and force sensors and limited payloads, that allow close human-robot interactions when used according to safety standards (ISO 10218, ISO 15066 etc.). Cobots are optimized for flexibility, ease of use and safe interactions. They often offer intuitive programming interfaces and are easier to redeploy on new tasks. Figure 2.2 shows examples of commercially available cobots.



(a) (b) (c)

Figure 2.2 Three widely used commercial collaborative robots: (a) Universal Robot UR10e, (b) Fanuc CRX-10iA, (c) Kinova Gen3

2.3.2 Payload and stiffness limitations

Designed for safe human-robot interactions, they allow a better workflow with human workers, however their limited payload and high compliance can be a problem for some applications. For example, a Universal Robot's UR20 has a payload of around 20kg and cobots in general have a stiffness within $3 - 4.5 \times 10^4$ N/m [9], which is much lower than industrial robots.

Doiron and Gallant [10] and Arakelian [11] use trajectory optimization and the addition of a balancer to increase cobot's payload without significantly increasing its mass. Other solutions such as flexible link manipulators and variable stiffness mechanisms have also been investigated to improve payload while keeping the same precision as shown in [12], [13] .

2.3.3 Cobotic drilling in the industry

Some cobotic applications in the industry involve the use of an end-of-arm crawling system as shown by ARM Robotics [2]. The crawling robot uses suction cups to attach to the surface and drill at different positions.

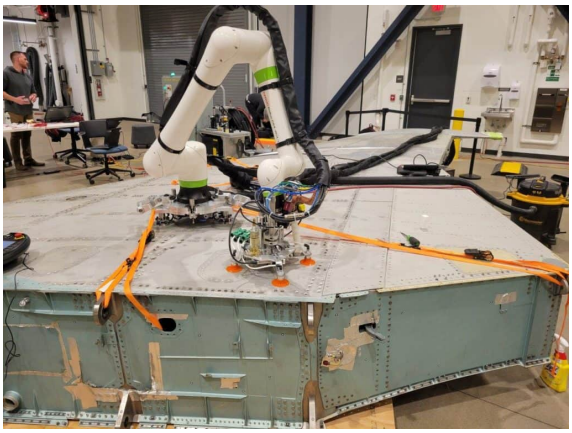


Figure 2.3 Crawling robot held by a cobot for drilling application [2]

Cobotic drilling solution from Fokker involves the use of a drill jig to guide a drilling end-effector mounted on a cobot [3].

Safran's approach to cobotic drilling is based on the direct collaboration between a human operator and the cobot: The worker would guide the cobot's drilling end-effector to the desired position of drilling [4].

2.4 Drilling process control

Advances in automatic drilling have led to an increase in process quality and productivity. Control of drilling parameters is a key component to guarantee hole quality (such as diameter, cylindricity, concentricity, run-out, etc), especially when working with multi-material (such as aluminum, titanium and composite) stacks. From force-deformation models , work in [14] [15] [16] [17] shows that clamping force has an effect on hole quality, since it helps reduce interface damage and inter-layer chip formation. Depending on the material's hardness,

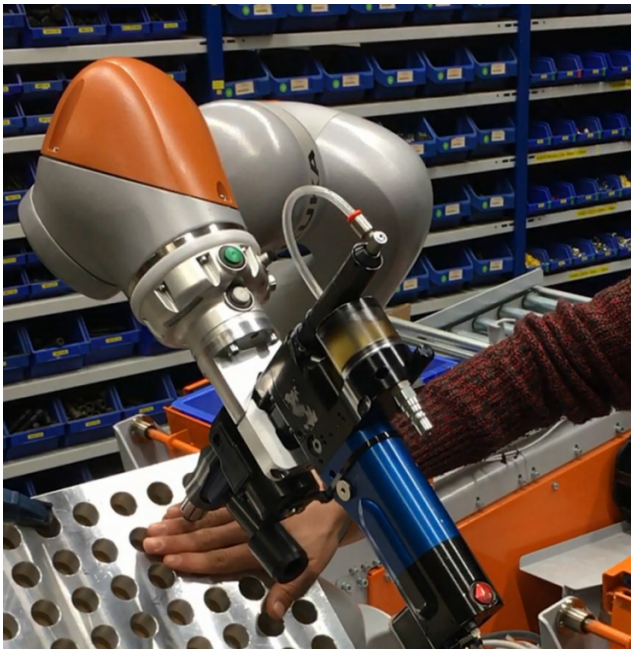


Figure 2.4 KUKA LBR iiwa with its drilling end effector in front of a drill jig [3]



Figure 2.5 Cobot helping the operator to handle the drilling tool [4]

clamping force should be between 500N and 1500N [15] , hence the choice of 600N in this project. For example, titanium needs greater clamping force than aluminum and composite since it is a harder material. Right control over drilling forces and cooling parameters can increase hole quality and reduce tool wear in multi-material stacks, as demonstrated in [18]. Control over chip size and evacuation can also have an impact on the hole quality. [7] shows that vibration-assisted drilling can effectively reduce chip size and make chip evacuation easier, resulting in smoother holes and a potential increase in productivity by reducing the need for further finishing operations.

2.5 Dynamic analysis and modeling

Dynamic analysis and modeling of the drilling process is essential to predict the system response and performance of the robot as well as the out-come quality of the drilled hole.

2.5.1 Modal analysis

An approach to maximise hole quality is the use of modal analysis. Robots being pose dependant systems, their stiffness properties vary accordingly and having a pose dependant model is required. Modal analysis gives information about the system's peak frequencies that would cause a resonance during the process and deteriorate manufacturing quality. The different approaches to analyze a robotic system for manufacturing are separate (robot and tool as distinct bodies) or joined (robot + tool). Chen et al. [19] used a separate approach to predict the pose dependant FRF by coupling a milling tool to an industrial robot. The identification of modal parameters of the robot was made using experimental tap tests and mathematical derivation and the tool's FRF was modeled using a FEA. Chen & Ahmadi [20] used a joined approach to predict the pose dependant FRF of a milling robot. They combined a multibody model of the robot with a data-driven machine learning model (Gaussian Process Regression) obtained from tap tests.

2.5.2 Hole quality prediction

On a more direct approach, direct prediction of hole quality metrics can improve the drilling process by linking hole quality metrics with process parameters (such as cutting speed, feed and drill bit diameter). Such models open the door for multi-objective optimization, fast monitoring and online adaptive drilling implementations. Models revolve around mechanistic, empirical or machine learning. A mechanistic model is an analytical model based on the physics behind the manufacturing process. In contrast, an empirical model is a model strictly built from data using simple regression techniques. We distinguish empirical models from ML models by the fact that ML models use more complex algorithm beyond linear regressions such as Neural Networks or Decision Trees for example.

Mechanistic models

An example of a mechanistic model can be reported in Flachs et al. [21]. The authors used the drill bit geometry to derive the thrust force prediction and cutting forces data to calibrate the model's parameters. Tsao and Hocheng [5] proposed analytical models to predict the

critical thrust force causing delamination during drilling and to compare the efficiency of the industry's most common solution: using a backup plate after the workpiece, as pictured in Figure 2.6.

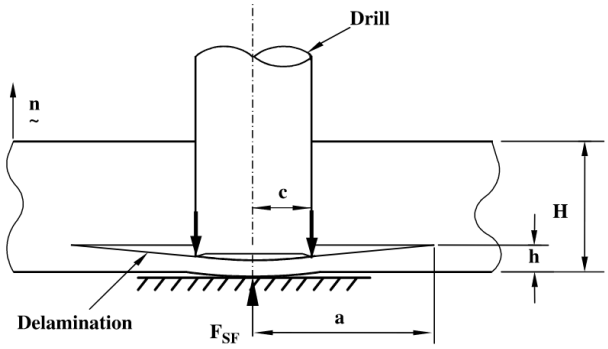


Figure 2.6 Delamination model for drilling, with the backup plate from [5]

Empirical models

Kumar and Packiaraj [22] used a linear regression to describe surface roughness, tool wear, material removal rate and hole diameter using process parameters. Similarly, Babu and Dhanasekaran [23] studied thrust force, surface roughness, and roundness error to develop prediction models from drilling parameters like feed, cutting speed and drill bit point angle. One of those models was a linear regression including quadratic terms. Alvarez-Alcon et al. [24] studied how cutting speed and feed rate affected hole quality during drilling of carbon fiber sheets. They evaluated hole quality metrics such as surface roughness, cylindricity and diameter and also used information given by thrust force, torque and electric spindle power signals to monitor and predict hole quality in real time.

Machine learning models

Along a linear regression, Babu and Dhanasekaran [23] also trained an Artificial Neural Network (ANN). The ANN takes in his input layers the process parameters (point angle, spindle speed, feed rate) and as output the predicted metrics (thrust force, surface roughness, roundness error) with a single hidden layer.
Despite showing good accuracy, the model lacked validation on unseen data or robustness analysis, which limits its reliability in real production settings. Also, the model's performances were not evaluated on different tool geometries or materials.

Shaban et al. [25] used a Logical Analysis of Data (LAD) to classify hole conformity from their

associated process parameters. LAD is a ML classification technique that identifies Boolean patterns in input to distinguish between different classes. The monitored variables used as inputs were the thrust force, cutting force and torque. The outputs were a classification of the surface delamination, hole circularity, hole diameter and surface roughness into two classes: conform or non conform.
LAD's rule based interpretability is powerful in quality-critical applications. However, the study is limited to binary classification and does not explore regression or multi-class outputs, potentially restricting its usefulness in nuanced process monitoring scenarios.

Dominguez-Monferrer et al. [26] used machine learning to predict tool wear during CFRP drilling. Prediction algorithms such as linear regression, k-Nearest Neighbors (kNN) and Random Forest (RF) were explored. The prediction model would take as input the spindle's power consumption information, cumulative cutting time and hole thickness and would predict the tool wear into 3 levels, each defined by a wear depth value.
Their non-intrusive sensor setup and multi-class output are practical for industrial deployment. However, the study bases itself on a limited number of drills and does not explore the possible impacts of variable environmental or stacking conditions, which may affect scalability.

Dekena et al. [27] worked on materials identification using machine learning for bi-material workpieces during turning operations. Cutting and feed forces, accelerations, spindle torque and motor current are used as inputs of the model. The classification ML model would detect which material is being machined in real time.
The real-time classification approach shows excellent accuracy for different materials. However, the model's performances degrade for materials with similar mechanical properties and adaptive control strategies based on misclassifications are not discussed.

Susai Mary et al. [28] investigated the use of machine learning for hole cylindricity prediction and monitoring during drilling. The prediction model would use vibration signals and feed and rotation speeds as input to predict the hole's cylindricity through an Adaptive Neuro Fuzzy Inference System(ANFIS).
The use of ANFIS effectively captures non linearities between process dynamics and form errors. However, the model was trained on a limited dataset without external validation and it did not consider other influencing factors like tool wear or material heterogeneity.

Dornfeld and DeVries [29] developed a tool wear monitoring system for turning operations using an ANN and sensor fusion. Accoustic emission, cutting forces and spindle motor current were used as inputs to classify the tool wear into used or new.
This work introduced sensor fusion for tool wear monitor. However, the classifier was binary

and the network simplistic compared to to modern standards, lacking generalization tests or comparisons with alternative models.

Qian et al. [9] explored a solution to monitor hole inclination angle during robotic drilling based on deep learning. Using operational vibration signals and vision based inclination measurements, a ResNet classifier is used to classify the validity of a hole based on its inclination, with a threshold of 0.8°.
The method showed excellent real-time performance and accuracy. However, the model is limited to binary classification and does not account for intermediate inclination levels. Additionally, the approach assumes consistent material and tool conditions, which may give different results in more variable scenarios.

Lee et al. [30] proposed a solution for in-situ evaluation of both hole quality and tool wear in robotic drilling of composite materials by using machine learning. An Ensemble Neural Network (ENN) was trained by taking as inputs the feed speed, rotation speed, feed torque and rotation torque to classify holes into 3 levels of hole delamination, the tool wear was indirectly inferred from torque measurements.
The monitoring and classification of both hole quality and tool wear is a strength that enables practical real-time implementation. However, relying on a single hole quality metric (delamination factor) may overlook other hole defects. Despite tool wear being considered, it does not give a usable prediction of the tool wear since it is not labeled or quantified, which may limit interpretability for predictive maintenance.

Table 2.1 summarizes the ML applications in manufacturing.

2.6 Summary and positioning of this work

This literature review explored the current landscape of automated drilling in aerospace manufacturing, with a particular focus on the potential and limitations of cobots. While cobots have multiple advantages such as flexibility, safety and simple integration, they are limited by their high compliance and low payload to perform tasks like aerospace drilling. Although mechanical and control-based solutions exist such as balancers, flexible links and stiffness adaptation, few are validated in industrial settings like aerospace drilling.

Industry examples show that cobots have been deployed for drilling in semi-guided environments using crawling systems, drilling jigs or direct human assistance. However, these solutions are limited in autonomy as they are reliant on external fixtures or operator input. In contrast, traditional automated drilling systems use CNC machines or high-stiffness industrial robots and multi-function drilling end-effectors, but they lack the adaptability and

Table 2.1 Summary of machine learning-based approaches for prediction and monitoring in drilling and machining tasks

Reference	Application	Sensing/Input Data	Prediction Target	ML Models Used
[23]	Surface quality prediction in drilling	Process parameters (spindle speed, feed rate, point angle)	Surface roughness, thrust force, roundness error	Linear Regression, Neural Network
[25]	Hole quality classification in drilling	Thrust force, torque, cutting force	Conformity of hole geometry	Logical Analysis of Data (LAD)
[26]	Tool wear prediction in CFRP drilling	Spindle power, cutting time, hole thickness	Tool wear level (3-class classification)	Linear Regression, k-NN, Random Forest
[27]	Real-time material identification during turning	Cutting force, feed force, accelerations, spindle torque	Material type being machined	Classification (unspecified algorithm)
[28]	Cylindricity prediction in drilling	Vibration signals, feed rate, rotation speed	Hole cylindricity	Adaptive Neuro-Fuzzy Inference System (ANFIS)
[29]	Tool wear monitoring in turning	Acoustic emission, cutting forces, spindle current	Tool wear classification (used/new)	Neural Network
[9]	Inclination state classification in cobotic drilling	Vibration signals	Inclination state (normal vs. inclined)	ResNet Neural Network
[30]	In-situ hole quality classification in CFRP drilling	Servo motor torque, feed speed, rotation speed	Hole quality class (1 to 3 based on delamination)	Ensemble Neural Network (GA-ANN + Bayesian optimization)

safety cobots provide.

Advances in process modelling and control demonstrate that robot posture and optimal process parameters can improve hole quality in multi-stack drilling. However, these approaches are often detached from cobot-specific aerospace drilling applications.

While modal analysis dynamic modelling offers an indirect way of predicting hole quality by understanding the robot's vibration, a more direct alternative is to analyze the effect of process parameters on hole quality to develop prediction models leading to a monitoring

system for adaptive drilling based on drilling parameters tuning. The main issues with prediction and monitoring models seen in the literature are as follows:

1. Most models are for CNC drilling: the simplest monitoring systems were developed from drilling data or mechanistic models proper to rigid CNC machines. Robotics systems are pose dependant and compliant systems, thus making them not adapted. A direct reproduction of the employed methodologies may result in discrepancies.

2. Models for robotic systems are not suited for aerospace drilling: for the instances of robotic drilling applications [9, 30], the scope of application is not exactly replicable. Qian et al. [9] uses a cobot for inclination monitoring during drilling, but no clamping force is applied, while Lee et al. [30] application is involving an industrial robot.

Across the reviewed domain, a clear gap exists in the integration of robotic rigidity increase solutions, drilling end-effector and drilling monitoring systems. Most current solutions treat these aspects separately, limiting their scalability and adaptability. This fragmentation highlights the need for a unified framework for aerospace cobotic drilling. This master thesis will propose two of the elements constituting that unified framework: a novel lightweight cobotic smart drilling end-effector design and an adapted monitoring and an adaptive drilling system for aerospace cobotic drilling.

CHAPTER 3 OBJECTIVES AND RESEARCH QUESTION

3.0.1 Research Question

How effectively a collaborative robot equipped with a rigidity enhancement exoskeleton drills holes for aerospace manufacturing, and how accurately can those holes be monitored during the process?

3.0.2 Research assumptions

Assumption 1: Only plane drilling is considered. Drilling on a flat surface allows more consistent drilling tests as we keep the same TCP orientation and only translate the tool following the workpiece plane. Also, no surface normality detection method or adapted test bench was available at the beginning of the project, and we use a flat surface as a first step into the development of a normality detection method.

Assumption 2: Only homogeneous stacks made of three aluminum plates are considered. The stack properties such as the number of sheets, their thickness and their material are process parameters affecting the out coming hole quality. We fix these process parameters to keep the focus on the tool process parameters.

3.0.3 Main Objective and Specific Objectives

The main objective of this thesis is to design foundational elements for collaborative robotic drilling framework for aerospace to provide aerospace drilling capabilities for a UR10e collaborative robot. This framework is comprised of an adapted drilling-end effector and an adaptive drilling system. The remaining aspect of the previously defined framework, the rigidity increase solution, is not within the scope of this master thesis as it has already been addressed by another researcher at the NRC prior to the beginning of the project.

To reach that main objective, we must complete these specific objectives first:

SO1: Develop and prototype a cobotic smart drilling end-effector suitable for aerospace manufacturing. It must be able to apply 600N of clamping force during drilling, measure process parameters such as the clamping force, the feed speed and rotation speed with a sufficient resolution to be used in a monitoring system, and detect pilot holes with a position accuracy < 5mm and surface-normal with an average error $< 1°$.

SO2: Design an adaptive drilling system for geometric hole quality metrics (diameter, cylindricity, concentricity and runout). It must be able to predict hole quality metrics with a $R^2 > 0.7$ on the test set and classify process parameters with a test accuracy $> 80\%$.

CHAPTER 4 METHODOLOGY

4.1 Methodology Overview

This section outlines the methodology used in this research, starting with a description of the experimental setup. This includes a description of the UR10e cobot used in this project: Exobrake. Next, it introduces the test bed used for drilling tests and the material used for data acquisition.

Following this, we move to the methodology behind **SO1**, which covers the design and prototyping of ALCCE. This section includes the strategy behind the overall design of the EE to fit the structural constraints of the cobot. Next is the integration of the sensors and the actuators used by the end-effector (EE). We conclude this section by the integration of the vision system.

Finally, the last section will focus on **SO2** and presents the methodology leading to the adaptive drilling system. This section will first describe the design of the monitoring algorithm through the following pipeline: Design of experiment, features extraction, features selection, model selection and training, and model testing. We conclude this section with an explanation of how the monitoring algorithms lead to the adaptive drilling advisor.

4.2 Experimental Setup

4.2.1 Exobrake presentation

Exobrake is a UR10e cobot with its exoskeleton. Figure 4.1 shows a CAD representation of Exobrake. Exobrake is a 6 DOF robotic system, direct kinematics can be used to predict the end-effector cartesian coordinates using the robot joints angles values. By convention, we will denote the Base joint as Joint 0. Transformation matrices will follow this convention:

To represent a point expressed in frame A in frame B:

$$\vec{P}^B = \mathbf{T}_{A\to B}\vec{P}^A = \mathbf{T}_A^B\vec{P}^A \tag{4.1}$$

With

$$\mathbf{T}_A^B = \begin{bmatrix} \mathbf{R}_A^B & \vec{t}_A^B \\ \vec{0}_{1\times 3} & 1 \end{bmatrix} \tag{4.2}$$

$\mathbf{R}_A^B$ being the rotation matrix between frame A and B and $\vec{t}_B^A$ being the translation vector

between frame A and B.

In this document, points $\vec{P}$ are expressed in their $\mathbb{R}^4$ augmented form, unless specified otherwise:

$$\vec{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{4.3}$$

The forward kinematics of a 6 DOF serial robot with an end-effector defined by a TCP:

$$\mathbf{T}_0^{TCP}(\vec{q}) = \mathbf{T}_0^1(q_0)\mathbf{T}_1^2(q_1)\mathbf{T}_2^3(q_2)\mathbf{T}_3^4(q_3)\mathbf{T}_4^5(q_4)\mathbf{T}_5^{Flange}(q_5)\mathbf{T}_{Flange}^{TCP} \tag{4.4}$$

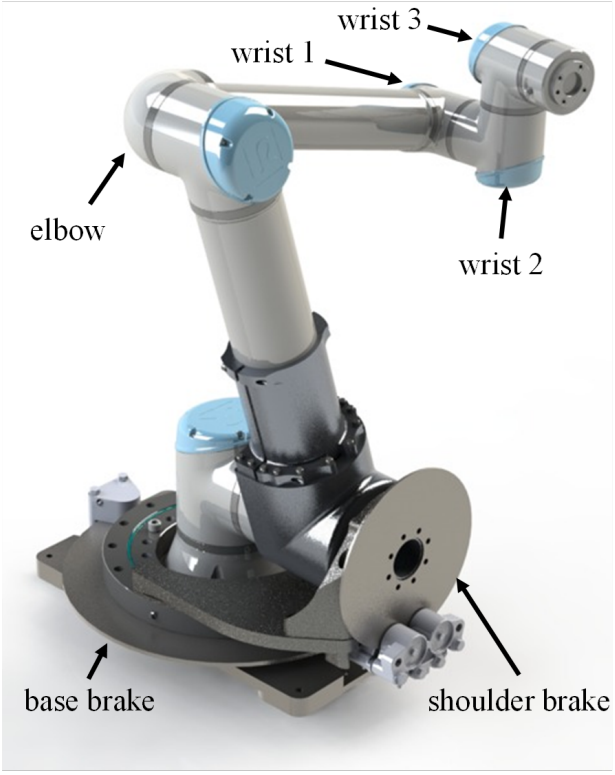


Figure 4.1 Exobrake CAD representation with its joints labelled

The UR10e control box has 8 digital inputs and outputs, and 2 analog inputs and outputs to connect external sensors and actuators.

Two programming methods were used in the project to control the robot: Direct *URscript* programming and TCP/IP remote programming. *URscript* is the programming language used by *Universal Robot* to control their robots. A *URscript* file can be uploaded to the

robot controller through a USB drive. On the other hand, TCP/IP remote programming involves socket programming through TCP/IP Ethernet protocol with the robot controller. Concretely, a *Python* program sends instructions through sockets to the robot's communication ports and can also read sockets from the robot. The instructions read by the robot are *URscript* instructions. Direct *URscript* programming is simpler as it is more straightforward: write the *URscript* program, upload it to the robot controller, and run. TCP/IP method has an added complexity layer since it requires low-level programming of the communication protocol with the robot controller. Some TCP/IP libraries such as *URX* or *URBasics* exist but are either outdated or missing key functions and often require additional programming efforts as required in this project. Despite its complexity, TCP/IP method allows for more programming flexibility as it takes advantage of *Python*'s diverse modules.

For example, TCP/IP method was used to simultaneously control the robot, acquire its joints information, and acquire serial sensors data from a micro-controller using *PySerial* module. Another possible application is to use TCP/IP to control the robot from data acquired from a camera running on the *PyRealsens* API.

4.2.2 Drilling Test Bed

To record reliable operational forces and vibrations data, the need for a rigid test bed is crucial. These are the requirements for the drilling test bed that we designed within the scope of this master project:

R1: Minimal deflection

R2: Flexible dynamometer placement

The drilling structure in Figure 4.2 is the final testbed design. The drilling structure consists of welded rectangular steel tubes forming a rigid frame. From the *Solidworks* CAD assembly, we run a FEA to predict the structural deflection where the 600N of clamping force is applied.

4.2.3 Data Acquisition Material

Force and Vibration acquisition The goal of the data acquisition system is to simultaneously record forces and vibration signals during drilling, at a sampling frequency of 20 kHz. We use a *Kistler 9272* 4 channels (Fx,Fy,Fz,Mx) dynamometer, as shown in Figure 4.3a. Similarly, we use a *PCB Triaxial ICP* 3 channels (X, Y, Z) accelerometer, as shown in Figure 4.3b.

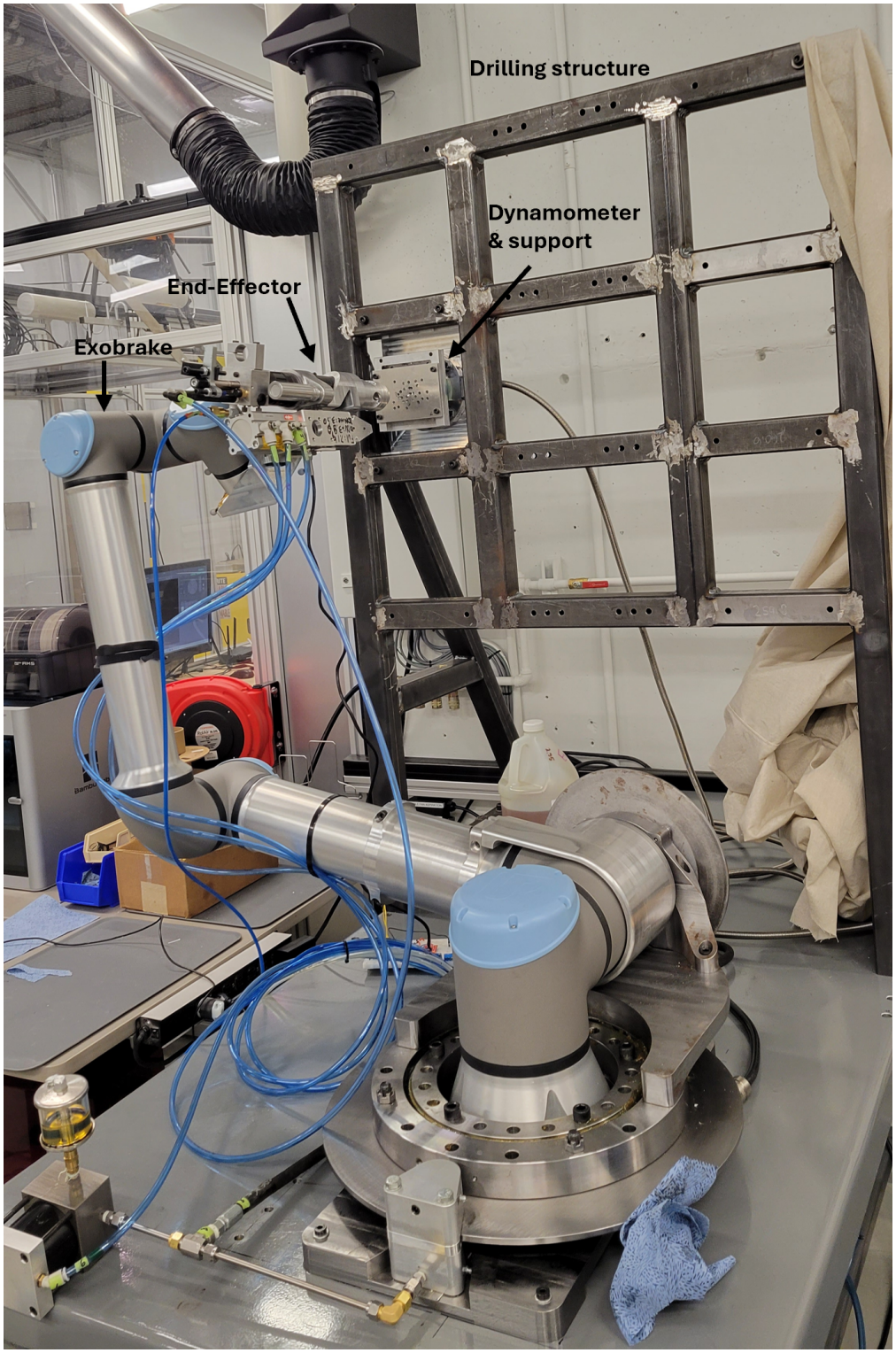
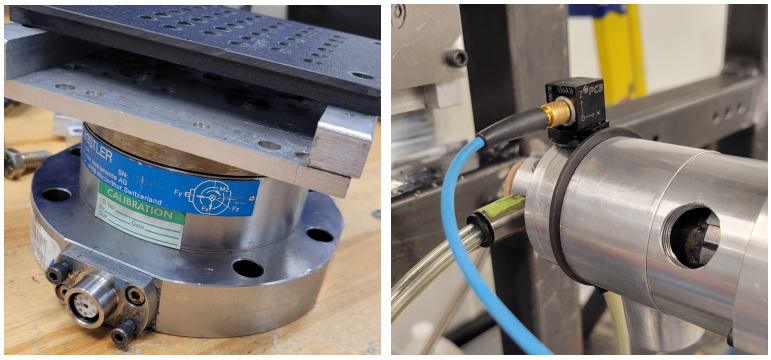


Figure 4.2 Drilling Test Bed



(a) (b)

Figure 4.3 Sensors used for data acquisition: (a) Kistler dynamometer (b) PCB accelerometer

Along with these sensors, we need signal conditioners and a data acquisition system to convert their signals into readable data for a computer. The dynamometer requires a *Kistler Multichannel Charge Amplifier* before entering an 8 channels *Siemens SCADA* data acquisition system. The accelerometer can be directly connected to the data acquisition system. The data acquisition system is then connected to a computer and the software *LMS Test Lab* is used to record and export the data. Figure 4.4 shows the data acquisition system with its wiring schematic.

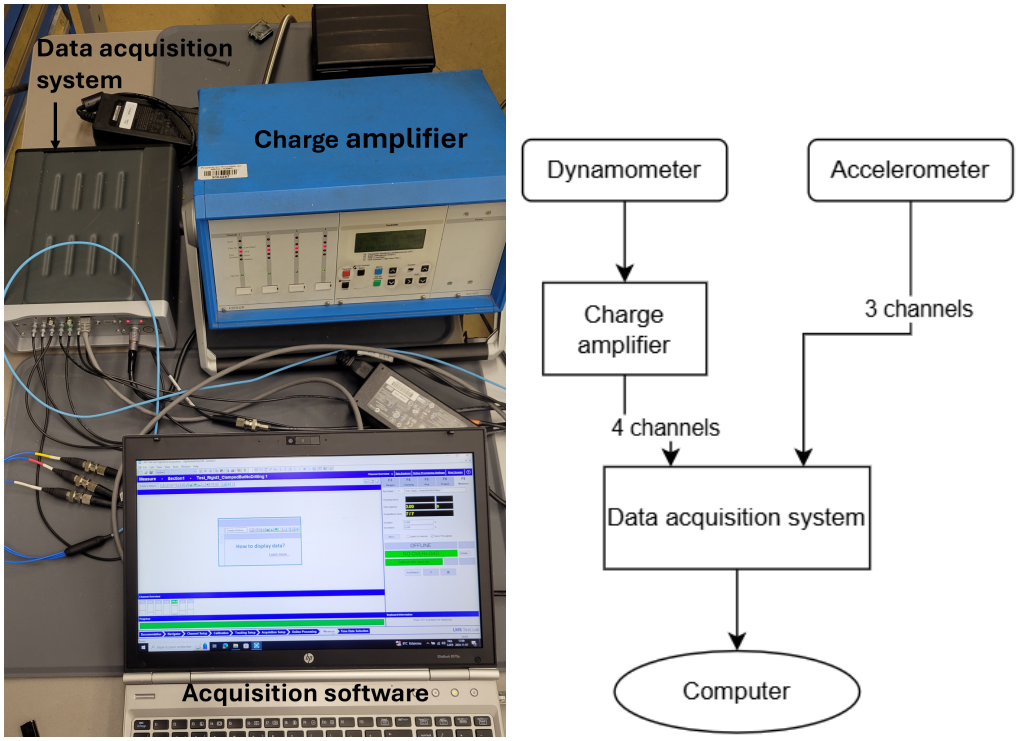
Optical Tracking System We also use a *Polaris VegaXT* optical tracking system to acquire positional data of the robot. This is done by recording the position and orientation of targets recognized by the optical tracking system. Figure 4.5 shows the optical tracking system along with its targets. The optical tracking system uses its dedicated software and outputs a set of the targets' poses over time.

4.3 End-Effector Design and Prototyping

4.3.1 TCP Positioning

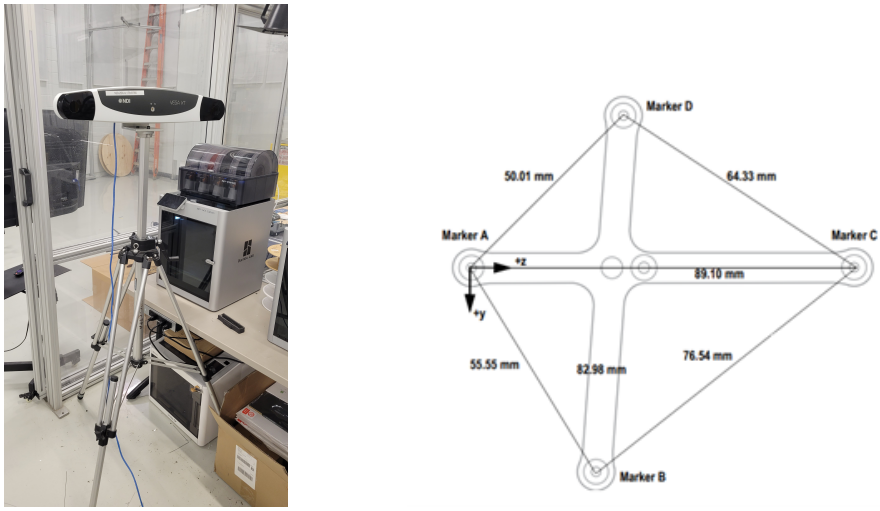
This section answers the following question: how should the robot TCP be positioned to minimize the efforts on the unprotected joints when applying the required clamping force for drilling?

From a *Matlab Simulink Multibody* model of the robot, we use surrogate optimization to build



(a) (b)

Figure 4.4 Data acquisition system (a) Data acquisition elements (b) Wiring schematic



(a) (b)

Figure 4.5 Laser tracking material (a) Polaris VegaXT optical tracking system (b) Detectable target, from the datasheet

the transformation matrix between the robot flange and the TCP:

$$\mathbf{T}_{flange \to TCP} = \begin{bmatrix} \mathbf{R} & \vec{t}^{\top} \\ \vec{0}^{\top} & 1 \end{bmatrix} \tag{4.5}$$

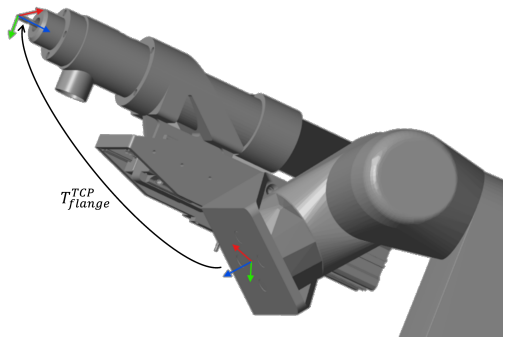


Figure 4.6 Transformation between the flange frame and the TCP frame

The optimization algorithm outputs the following solution vector:

$$\vec{x} = \begin{bmatrix} \theta & \phi & \psi & \vec{t} & \vec{P}_{cen} \end{bmatrix} \tag{4.6}$$

With:

- θ, ϕ, ψ the spherical parameters used for TCP orientation;
- $\vec{t} \in \mathbb{R}^2$ the Cartesian coordinates of the TCP, excluding y direction to simplify the design;
- $\vec{P}_{cen} \in \mathbb{R}^3$ the coordinates of the hole pattern center.

The holes pattern used in the *Simulink* model is a grid of 9 points projected on a double curvature surface defined by its width(w), height(h) and radius(r) as shown in Figure 4.7.

We used the following surface parameters: (w, h, p) = (0.1, 0.4, 1.5)m. We use a relatively small area to reduce the chances of having a non-feasible hole. A larger area may require drastic changes in the robot configuration to reach farther points in the pattern. The radius of 1.5 m was arbitrarily chosen to mimic a flat surface.

Figure 4.8 shows the spherical coordinate system used to orient the TCP Z axis (z') according to the flange. ψ allows the robot to rotate around the TCP's own Z axis for added flexibility

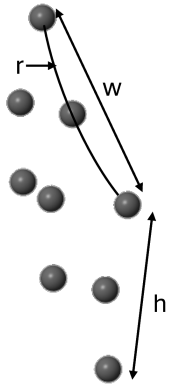


Figure 4.7 Double curvature drilling pattern

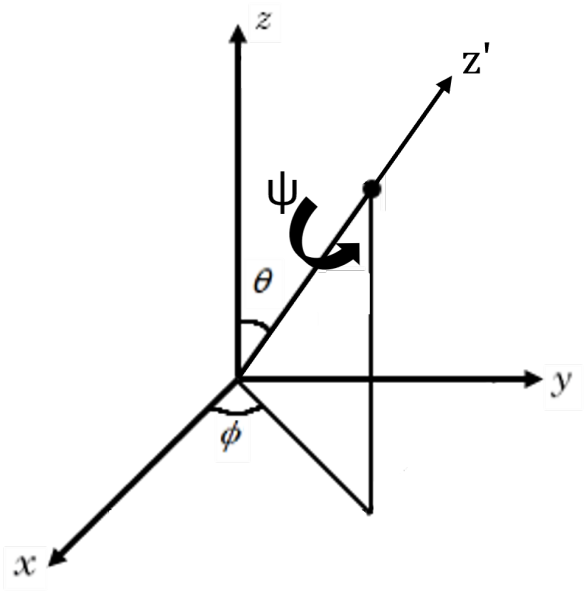
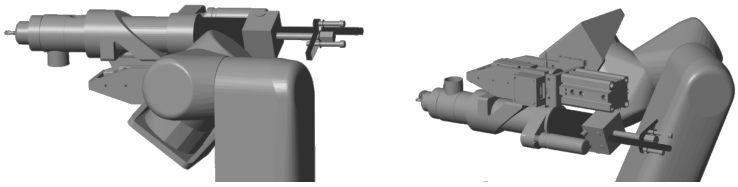


Figure 4.8 Spherical Coordinate system used for TCP orientation



(a) ψ (b) $\psi + \pi$

Figure 4.9 Effect of ψ on the robot pose, from *Simulink*'s visualizer

during the optimal parameters search. Figure 4.9 shows a representation of ψ.

The optimization problem has 2 main goals:

G1 Find the best set of geometrical parameters to define the TCP w.r.t the robot flange.

G2 Find the best drilling pattern position w.r.t the robot base.

Overall, the optimization algorithm's outputs are the optimal robot pose to adopt for drilling and the TCP geometric parameters. Only G1 is useful for the end-effector design.

The optimization algorithm aims to minimize the following cost function:

$$J = M_{z,max}(q_2) \tag{4.7}$$

Subject to inequality constraints:

$$M_{z,max}(q_i) < \tau_{max}(q_i) \quad i \in \{2, 3, 4, 5\} \tag{4.8}$$

$M_{z,max}$ is the maximum torque applied among the 9 holes evaluated in the drilling pattern. Only q_0 and q_1 are protected by the exoskeleton, so they are excluded from the problem's constraints. Table 4.1 presents the torque limits ($\vec{\tau}_{max}(q)$) of a UR10e cobot:

Table 4.1 Joint torque limits of the UR10e

Joint torque limits [Nm]					
q_0	q_1	q_2	q_3	q_4	q_5
330	330	150	56	56	56

One of the goals of that design was to keep the end-effector as compact as possible, meaning having a TCP close to the flange to avoid excessive torque on the end-effector structure while applying the clamping force. We can achieve this with the boundaries set on t_x and t_z. As for the orientation, the boundaries on θ, ϕ, ψ span the search domain over a complete sphere and a full rotation around z' for a flexible parameter search.

Those are the resulting boundaries used for the optimal parameters search:

$$0 \leq \theta \leq 2\pi \tag{4.9}$$
$$-\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2} \tag{4.10}$$
$$0 \leq \psi \leq 2\pi \tag{4.11}$$
$$0 \leq t_x \leq 0.3 \tag{4.12}$$
$$0 \leq t_z \leq 0.2 \tag{4.13}$$

We constrain the following parameters to keep the scope of the optimization problem limited to G1:

$$\vec{P}_{cen} = \begin{bmatrix} -0.37 & 2.1 & 0.89 \end{bmatrix} m$$

We use these values of $\vec{P}_{cen}$ according to the geometry and position of the drilling structure designed in the previous section. The following demonstration shows how to obtain $\mathbf{T}_{flange}^{TCP}$ from the optimization solution vector $\vec{x}$:

Following Equation 4.5, let's define

$$a = \sin\theta\cos\phi, \qquad b = \sin\theta\sin\phi, \qquad c = \cos\theta, \qquad D = a^2 + c^2.$$

$$\hat{\mathbf{z}} = \begin{bmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}. \tag{4.14}$$

We choose an X–axis orthonormal to $\hat{\mathbf{z}}$.

$$\hat{\mathbf{x}}_0 = \begin{cases} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, & a = 0, \\ \dfrac{1}{\sqrt{a^2 + c^2}} \begin{bmatrix} -c \\ 0 \\ a \end{bmatrix}, & a \neq 0\ , \end{cases} \qquad \hat{\mathbf{y}}_0 = \hat{\mathbf{z}} \times \hat{\mathbf{x}}_0. \tag{4.15}$$

With the above choice,

$$\hat{\mathbf{y}}_0 = \begin{bmatrix} \dfrac{ab}{\sqrt{D}} \\ -\sqrt{D} \\ \dfrac{bc}{\sqrt{D}} \end{bmatrix}. \tag{4.16}$$

Let

$$R_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.17}$$

Form the intermediate triad $R_0 = [\,\hat{\mathbf{x}}_0\ \hat{\mathbf{y}}_0\ \hat{\mathbf{z}}\,]$ and post–multiply:

$$R = R_0\, R_z(\psi). \tag{4.18}$$

Equivalently, the final axes are

$$\hat{\mathbf{x}} = \cos\psi\,\hat{\mathbf{x}}_0 + \sin\psi\,\hat{\mathbf{y}}_0, \qquad \hat{\mathbf{y}} = -\sin\psi\,\hat{\mathbf{x}}_0 + \cos\psi\,\hat{\mathbf{y}}_0, \qquad \hat{\mathbf{z}} \text{ unchanged.}$$

$$R = \begin{bmatrix} \dfrac{-c\cos\psi + ab\sin\psi}{\sqrt{D}} & \dfrac{c\sin\psi + ab\cos\psi}{\sqrt{D}} & a \\ -\sqrt{D}\,\sin\psi & -\sqrt{D}\,\cos\psi & b \\ \dfrac{a\cos\psi + bc\sin\psi}{\sqrt{D}} & \dfrac{-a\sin\psi + bc\cos\psi}{\sqrt{D}} & c \end{bmatrix}. \tag{4.19}$$

Singular case. :If $D = 0$ (i.e., $\hat{\mathbf{z}} = \pm\hat{\mathbf{y}}$), use the branch $a = 0$ above for $\hat{\mathbf{x}}_0$ and construct R via the axis formulas.

The given transformation matrix was adjusted during the design to simplify parts manufacturing. The adjusted TCP pose can be measured by running the TCP calibration routine in the robot teach pendant.

The optimization also computes the optimal robot configuration considering the defined drilling pattern. Among the 9 points in the drilling pattern, we chose the middle point, point 5. This robot configuration also needs to be adjusted to fit the real test bench environment.

The adjusted results were then fed back into the multibody model to confirm the feasibility of the adjusted solution.

4.3.2 Overall Structure

From the final TCP position, we build the end-effector architecture around.

The objectives behind the end-effector design were as follows:

O1 Fit the UR10e flange

O2 Fit the pneumatic drilling unit shape

O3 Respect the maximum payload of a UR10e

O4 Minimize structural deflection

O5 Allow the sensors and actuators integration

O1 is achieved simply by designing the flange bracket (1) in accordance with the UR10e flange drawings in Figure 4.10.

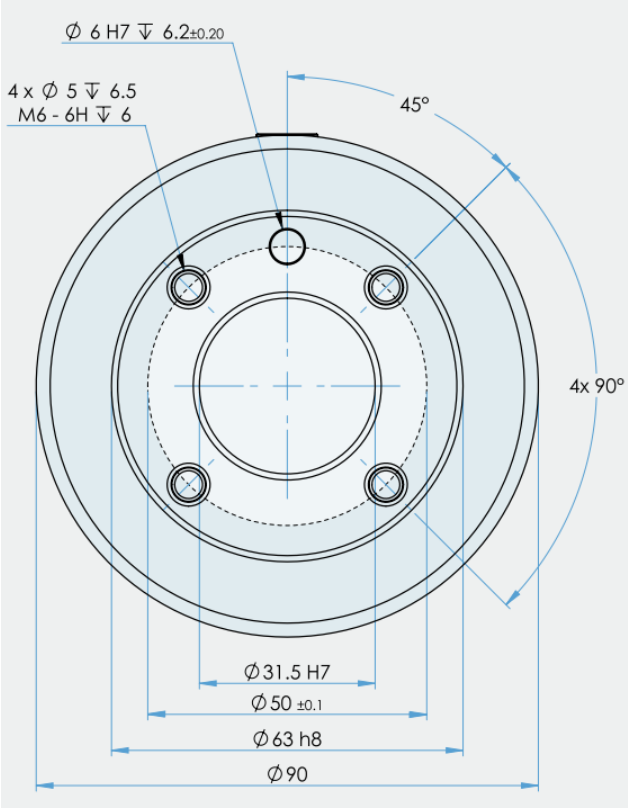


Figure 4.10 UR10e flange connection drawing, from the robot's datasheet [6]

The drilling unit has been chosen prior to the beginning of the project. Unlike electric drilling units, the pneumatic counterparts are generally lighter alternatives since they don't have any

electric motor, component that adds significant weight to electric drilling units. From the drilling unit's geometry, we dimension its supporting parts to allow a threaded connection within the end-effector, achieving **O2**.

The drilling unit uses hydraulic resistance from an included hydraulic damper to reduce its feed speed. The hydraulic damper can manually be adjusted from graduations between 0 (no resistance, maximum speed) to 30 (maximum resistance, minimal speed). Figure 4.11 shows the hydraulic check installed on the drilling unit, along with the linear potentiometer presented in the next section.

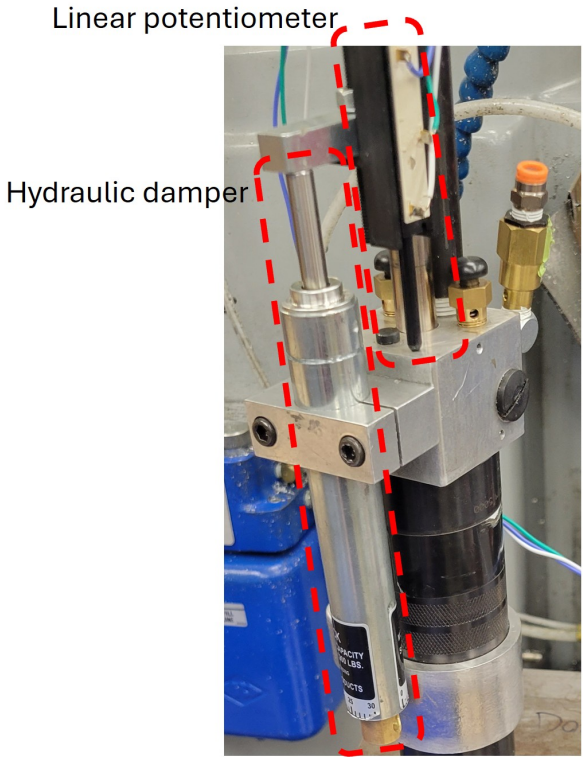


Figure 4.11 Drilling unit's integrated Hydraulic damper and Linear potentiometer for manual feed speed control and feed measurements

Figure 4.12 shows the maximum allowed end-effector mass according to its center of gravity offset from the flange. We set the mass limit of the end-effector using the prior geometrical information given by the non-adjusted TCP transformation matrix $\mathbf{T'}_{flange}^{TCP}$, whose translation part gives the farthest possible CoG location. The true CoG and mass of the final design can be measured following the assembly by the robot's payload measurement routine from the teach pendant, to verify if the payload/CoG offset combination is within the robot's capabilities.

Mass minimization of the end-effector involves material selection and selective mass reduction

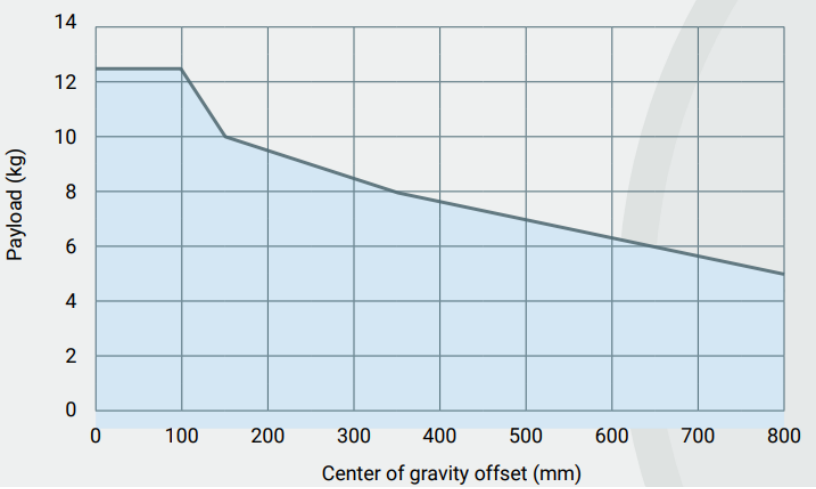


Figure 4.12 UR10e tool payload curve, from the robot's datasheet [6]

on applicable parts. We can add a lightening hole inside a fastener hole pattern on surfaces affected by shear forces by assuming fasteners carry most of the shear constraint.

As for **O4**, we started from the previous end-effector iteration (version 2.0) and improved the design to increase the overall structure rigidity for the current version (3.0). First, we minimized the cantilever distance between the robot flange and the TCP to reduce rotating efforts on the end-effector during the TCP positioning step, as mentioned earlier.

Next, we reduced as much as possible the number of manufactured parts in the assembly. We avoid unnecessary assemblies between two manufactured parts because assembled joints often lead to play when under bending efforts and thus are harmful to the overall rigidity. To compare both versions, we evaluate the number of joints between two manufactured parts in the force chain between the TCP and the robot flange that are under bending efforts.

4.3.3 Sensors and actuators

The following subsection will focus on the integration of sensors and actuators within the drilling end-effector. Unlike electric (or semi-electric) devices, fully pneumatic drilling units lack the ability to measure process variables. For example, electric drilling units can measure spindle rotation speed, feed rate, thrust force, motor currents, and many more parameters. The integration of sensors within the end-effector brings measurement capabilities to the drilling device and is the starting step for the design of a drilling monitoring system. Table 4.2 summarizes the measurement requirements and some sensing solutions that can be integrated.

The drilling unit being self-fed, the part to which the linear potentiometer is attached follows

Table 4.2 Measurement requirements and their adapted sensor

Requirements	Sensors
Feed position [mm]	Linear potentiometer
Feed speed [mm/s]	Linear potentiometer
Rotation speed [RPM]	Laser sensor, accelerometer
Clamping force [N]	Load cell
Thrust force [N]	Load cell

the linear movement of the spindle, giving the feed position as the linear potentiometer rod is pushed. Feed speed is obtained by differentiating the feed position.

Two methods were designed for rotation speed estimation: laser sensing and vibration analysis. Sticking a reflective band on the rotating part of the drilling unit, a laser inserted in the clamping nose pointing the rotating spindle would output a digital signal: 1 if it senses the reflective band, or 0 otherwise. Using the time interval between two rising edges $t_{\uparrow}$, we can compute the tool's rotation speed:

$$\omega = \frac{60}{t_{\uparrow}} \tag{4.20}$$

While laser sensing provides direct measurement, its viability in a Minimum Quantity Lubricant (MQL) is evaluated in Chapter 5.

From accelerometer vibration signals, the peak frequency observed in the FFT corresponds to a multiple of the tool's operational frequency depending on the number of edges n on the drill bit:

$$\omega = \frac{\omega_{max}}{n} \cdot 60 \tag{4.21}$$

ω_{max} being the peak frequency within the tool's operating frequency range. This method can only be used in chatter free scenarios; otherwise, the dominant peak represents the chatter frequency.

The sensors' data are acquired through an *Arduino UNO* microcontroller board using the USB protocol.

The axial force can be measured as the load cell is part of the force chain between the pneumatic cylinder and the TCP.

In parallel, the drilling end-effector described in the previous section needs added function-

alities to operate as an aerospace drilling tool. These functionalities are integrated through adding actuators and attachments that each fulfill a specific task.

Functional mapping

- **Clamping:** Pneumatic linear actuator sized for the required clamping force of 600N;
- **Lubrication:** MQL system delivered at the cutting zone, lubrication parameters are kept constant for repeatability purposes;
- **Chip evacuation:** Continuous vacuum chip extraction routed at the clamping nose to mitigate clogging caused by chips accumulation;
- **Feed control:** Passive hydraulic damping to regulate feed speed. Active or semi-active is considered as an upgrade path for closed loop control.

The actuation strategy adopts pneumatic elements with a simple control strategy.

Control strategy

- **Interface:** Proportional pneumatic actuation is driven by an analog command from the robot controller. On/Off devices are toggled by using digital I/O.
- **Setpoints:** Clamping setpoints are commanded via voltage sent to a proportional valve, lubrication is fixed, vacuum is manually enabled before drilling and kept continuous during cutting
- **Integration:** Actuator commands are time-aligned with sensing and motion states to support later feature extraction and decision-making.

4.3.4 Vision System

Along with the previously integrated sensors, a vision system is also included in the end-effector. In the context of cobotic drilling, pilot holes are manually drilled on a curved surface cockpit by human operators. From these pilot holes, a drilling pattern is virtually generated, that must be drilled by a robot. The role of the vision system is to detect these pilot holes and guide the robot toward the generated holes while also giving information about the surface normal vector locally at the drilling site to guarantee perpendicular drilling.

This leads us to define the following objectives for a initial vision system prototype:

O1: Detect pilot holes on the workpiece with an accuracy < 5mm;

O2: Find the surface normality vector with an average orientation error $< 1°$

The two stated objectives can be classified as 3D vision problems, since their completion requires depth information along with the image. Two approaches can be considered:

1. Use a 2D camera + depth sensor
2. Use a 3D camera

The approach taken in this project was approach #2. This approach is simpler as it saves us from the integration of an extra sensor and is easier to integrate in a compact drilling end-effector. For a single camera capture, a 3D camera returns a 2D image and its corresponding 3D pointcloud.

The camera chosen is an *Intel RealSens D405* stereo camera. The main reason for this choice of model is its compact design. Besides its dimensions, the camera's range of 7 - 50cm suits the needs of our application, as this distance from the workpiece is within the reachable range of the robot. The camera should be installed on the end-effector in such a way to have its field of view free and unobstructed.

The objectives of the vision system are reached by developing a *Python* computer vision program based on *OpenCV* library. The workpiece picture taken by the camera shown in Figure 4.13 is used to test the program. The program should detect the holes and return the surface normal vector at their positions.

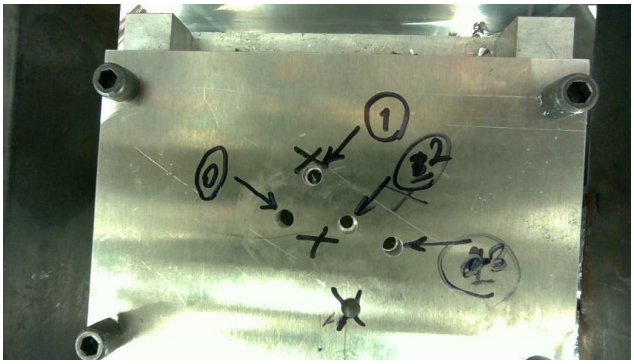


Figure 4.13 Workpiece test image used for the surface normality detection. The marks have later been removed on *Paint*

We first define a bounding box around zone of interest, in which we should detect the pilot holes. In the zone of interest, the *cv2.HoughCircle* function detects the holes, returning

their position and diameter. Experience showed that the Hough Circle algorithm, for our application, often detects a hole smaller than the actual hole because of the inside walls of the hole, as seen in Figure 4.13. For this reason, we define a larger hole around the nominally detected one by adding a buffer on the diameter. This way, we ensure that the extended hole's circumference lies on the workpiece's surface, and not inside the hole.

We now have the positions of the holes in the camera frame, and its extended circumference. Next, a simple method would be to project the 2D pixels lying on the augmented circles' circumference into the 3D pointcloud. Finding the surface normal vector expressed in the camera frame $\vec{n}_{cam}$ technically requires only 3 points. Let $\vec{p_1},\ \vec{p_2},\ \vec{p_3}$ the 3 circumference points lying on the pointcloud:

$$\vec{n}_{cam} = (\vec{p_2} - \vec{p_1}) \times (\vec{p_2} - \vec{p_3}) \tag{4.22}$$

However, this method is sensitive to measurement errors, as a single misaligned point can drastically impact the result. For this reason, we use multiple points to compute the surface normal vector using the least squares method.

The program randomly selects an initial pool of 200 points within the augmented circle's area. We use local points near the pilot hole to compute the surface's normal locally, as if this method was applied on a curved surface. Those points are mapped on the 3D pointcloud to their nearest neighbor. We use a depth threshold expressed in pixels to guarantee a selection of points within the plate surface and not inside the hole; the selected points are below this threshold. Points sinking inside a hole would have a larger depth coordinate and thus not be selected. Finally, we reject outliers based on the depth value z-score of the points previously selected according to a defined outlier threshold. This added verification layer prevents having points too far away from each other in terms of depth. The depth threshold and outlier threshold have been fine tuned; the best values are respectively 85 pixels and 0.5.

The plane equation can be written as follows:

$$ax + by + z + d = 0 \iff Ax + By + Cz + D = 0 \tag{4.23}$$

Given a point $\vec{p_i} = (x_i, y_i, z_i)_{i=1}^{N}$, a,b,d are picked to minimize the sum of square residuals:

$$\min_{a,b,d} = \sum_{i=1}^{N} (ax_i + by_i + d - z_i)^2 \tag{4.24}$$

Under matrix form:

$$\mathbf{M} = \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & 1 \end{bmatrix}, \qquad \vec{\theta} = \begin{bmatrix} a \\ b \\ d \end{bmatrix}, \qquad \vec{z} = \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix}. \tag{4.25}$$

We then solve the linear least squares problem:

$$\min_{\vec{\theta}} \left\| \mathbf{M}\vec{\theta} - \vec{z} \right\|_2^2 \tag{4.26}$$

$$(\mathbf{M}^\top \mathbf{M})\, \vec{\theta} = \mathbf{M}^\top \vec{z} \tag{4.27}$$

Whose solution can be obtained using the pseudo inverse:

$$\vec{\theta} = \left(\mathbf{M}^\top \mathbf{M}\right)^{-1} \mathbf{M}^\top \vec{z} \tag{4.28}$$

From the solution, we find the surface normal vector:

$$\hat{\vec{n}}_{cam} = \frac{1}{\sqrt{a^2 + b^2 + 1}} \begin{bmatrix} a \\ b \\ -1 \end{bmatrix}. \tag{4.29}$$

To validate the surface normality detection algorithm, we define a reference normal vector computed from points lying on the edges of the bounding box. Since we are on a flat surface, any points should have the same surface normal vector.

So far, the data provided by the vision system are only considered within the camera frame. To integrate this program into a robotic system, we need to transpose the points seen in the camera's pixel frame to the robot base frame (or world frame). With the addition of the camera, we need the information about the position of the camera w.r.t to TCP: the TCP to camera transformation matrix (or camera extrinsic matrix) $\mathbf{T}_{TCP}^{cam}$. This transformation matrix is obtained by performing hand-eye calibration [31]. In other words, a point expressed in the camera frame can be transposed onto the world frame following this series of rigid transformations:

$$\vec{P}^W = \mathbf{T}_{TCP}^{W}(\vec{q})\mathbf{T}_{cam}^{TCP}\vec{P}^{cam} \tag{4.30}$$

To perform hand-eye calibration, we first need to capture multiple images of a calibration grid at different robot poses. A total of 10 images were taken at different poses. Figure 4.14 shows the setup used for hand-eye calibration, and Figure 4.15 shows an example of pictures taken by the camera used for hand-eye calibration.

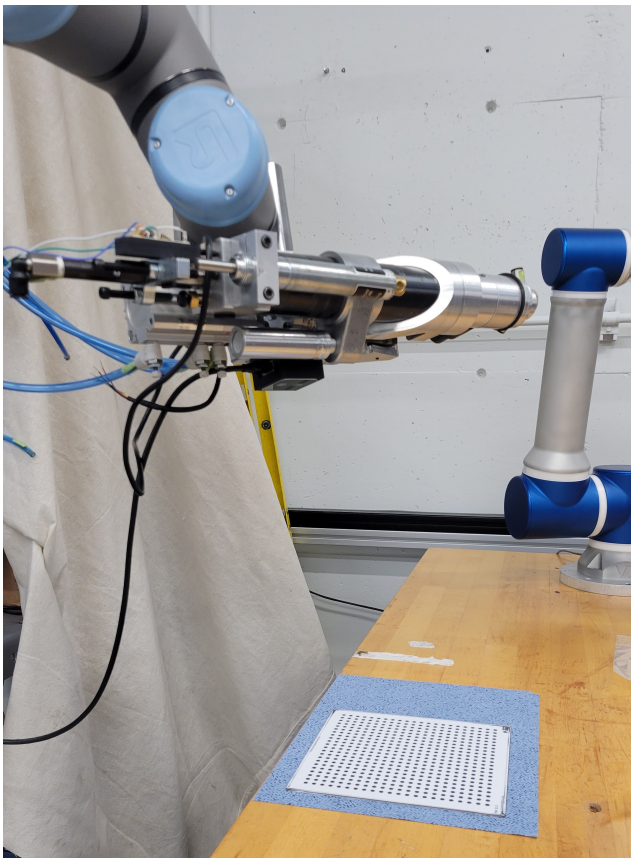


Figure 4.14 Example of a robot pose used to capture an image of the calibration grid during hand-eye calibration

The calibration grid images and their corresponding robot poses are then fed into a hand-eye calibration *Python* program based on *OpenCV* library.

OpenCV proposes two functions to perform hand-calibration: the standard version *cv2.calibrateHandEye* and the alternative version *cv2.calibrateRobotWorldHandEye*. The main difference is that *cv2.calibrateRobotWorldHandEye* also returns the transformation matrix between the robot base and the calibration grid and thus uses a different algorithm.

On the other hand, the remaining variation is about how our algorithm acquires the calibration grid pose w.r.t the camera $\mathbf{T}_{grid}^{cam}$. The solve PnP algorithm (based on *cv2.solvePnP*) can estimate the pose of the calibration grid given the spacing between each dot on the cali-



(a) (b)

Figure 4.15 Example of calibration grid pictures taken by the camera, used for the hand-eye calibration procedure (a) Image 1 (b) Image 6

bration grid; this is a fully 2D option. The PCD alternative projects the 2D dots of the grid on the pointcloud to directly find the grid pose, this is a 2D-3D option that takes advantage of the 3D information provided by the camera. This leaves us with 4 hand-eye calibration methods:

4 approaches were used to perform the hand-eye calibration:

1. Hand-eye + PnP
2. Hand-eye + PCD
3. World Hand-eye + PnP
4. World Hand-eye + PCD

Both methods start by detecting the points on the calibration grid as shown in Figure 4.16.

The function *cv2.findCirclesGrid* detects the dots given the dimensions of the dots pattern (22x22). The cross is what we define as the "origin" of the calibration grid. The solvePnP method only needs information about the dots' position. For the PCD method, we must combine the information from the 2D image from Figure 4.16 and its pointcloud.

Before projecting, we must undistort the image. Images given by cameras often come with distortion and it is a crucial parameter to consider in computer vision problems. The camera's distortion coefficients are considered, with the camera matrix **K**, as intrinsic camera parameters and normally require a different calibration procedure to obtain them. Hopefully, *Realsense* cameras are factory calibrated and the intrinsic parameters are stored in the camera's internal memory and accessible using the *get_intrinsic* Python function from the *PyRealsense* library. The distortion coefficients are then used in the *openCV undistort()*

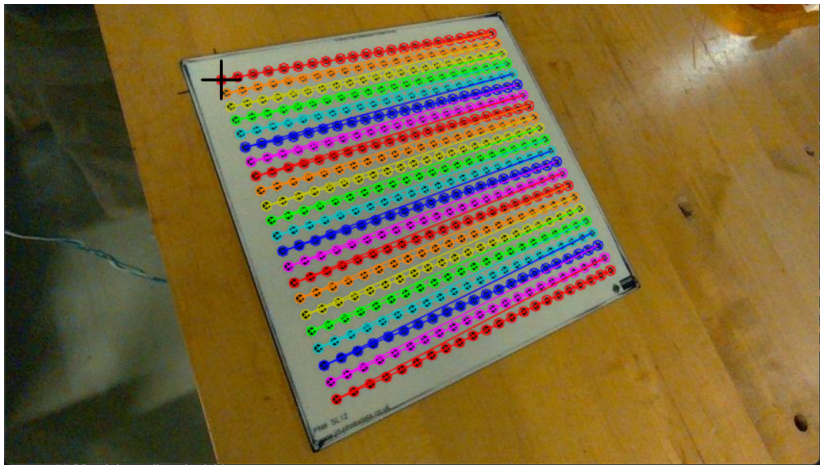


Figure 4.16 Example of detected dots pattern on the calibration grid

function that returns the undistorted image and the adjusted camera matrix to use with the undistorted image, $\mathbf{K}'$.

The idea of projecting the grid's points on a perfect plane comes from the assumption that the points lay on a perfectly flat grid and because having the points layout in a perfect plane allows us to determine the grid's pose, as shown later. We fit a plane on the calibration grid pointcloud by using the *RANSAC* algorithm. Compared to the previously presented least-squares method for plane fitting, *RANSAC* is a more robust consensus-based method that repeatedly samples three points to fit hypothesis planes, selects the one with the largest inlier respecting a distance threshold and then refines this plane using least-squares. The RANSAC method is chosen to take advantage of the larger amount of available points. Once the plane is fitted, each dot is projected to a 3D ray whose intersection with the plane results in the dot's 3D representation in the camera frame. From the dot's coordinates in the (u, v) pixel frame, we can find its 3D correspondence with the following demonstration:

Let $\mathbf{K}'$ be the new camera matrix:

$$\mathbf{K}' = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.31}$$

f_x, f_y being the camera focal length and c_x, c_y being the camera's optical center coordinates, both in pixels. Any ray coming from the camera is proportional to the camera point coordi-

nates:

$$\vec{r} \propto \begin{bmatrix} x_c \\ x_y \\ 1 \end{bmatrix} = \mathbf{K'}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{4.32}$$

A 3D point lying on that ray is

$$\vec{P}(\lambda) = \lambda \vec{r} = \begin{bmatrix} \lambda x_c \\ \lambda y_c \\ \lambda \end{bmatrix} \tag{4.33}$$

The plane equation:

$$aX + bY + cZ + d = 0 \iff \vec{n}^\top \vec{P} + d = 0 \tag{4.34}$$

The plane intersects the ray

$$\vec{n}^\top \lambda \vec{r} + d = 0 \tag{4.35}$$

$$\lambda = -\frac{d}{\vec{n}^\top \vec{r}} = -\frac{d}{ax_c + by_c + c} \tag{4.36}$$

Finally, the projected 3D point:

$$\vec{P} = -\frac{d}{\vec{n}^\top \mathbf{K'}^{-1} [u\ v\ 1]^\top} \mathbf{K'}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{4.37}$$

With the 3D representation of the dots pattern, we can get the transformation matrix between the camera and the calibration grid $\mathbf{T}_{grid}^{cam}$.

Let's organize the calibration grid dots into a matrix $\mathbf{C} \in \mathbb{R}^{22\times22}$ that stores each of the $\mathbb{R}^{3\times1}$ calibration grid points according to their position on the grid. Its origin is at (0,0) and the x and y neighbors are respectively at (1,0) and (0,1). We can define the calibration grid's x and y vectors w.r.t the camera frame:

$$\vec{x} = \mathbf{C}(1,0) - \mathbf{C}(0,0) \tag{4.38}$$

$$\vec{y} = \mathbf{C}(0,1) - \mathbf{C}(0,0) \tag{4.39}$$

$$\vec{z} = \vec{x} \times \vec{y} \tag{4.40}$$

And their normalized form:

$$\hat{\vec{x}} = \frac{\vec{x}}{||\vec{x}||} \tag{4.41}$$

$$\hat{\vec{y}} = \frac{\vec{y}}{||\vec{y}||} \tag{4.42}$$

$$\hat{\vec{z}} = \frac{\vec{z}}{||\vec{z}||} \tag{4.43}$$

The grid to camera transformation matrix:

$$\mathbf{T}_{grid}^{cam} = \begin{bmatrix} \hat{\vec{x}} & \hat{\vec{y}} & \hat{\vec{z}} & \mathbf{C}(0,0) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.44}$$

To measure the performance of the vision system, we must evaluate how precisely and accurately it can represent a point in space. The accuracy is the error between a measured point and its ground truth value. The precision defines how close within each other the measured points are. For accuracy, we detect the calibration grid's origin position with the camera, transpose it into the robot base frame by using Equation 4.30 (resulting in $\vec{P}^W$) and compare it with a reference value. The pose measured by the robot controller when probing the calibration grid origin with the TCP is used as the reference value for accuracy evaluation. For precision, the 10 robot pose/image combinations used for hand-eye calibration are used. All 4 hand-eye calibration methods are evaluated. Between the PnP and PCD methods to retrieve the calibration grid origin position, only the PCD method can be used to detect pilot holes on a workpiece since it does not need any calibration grid for position detection, unlike the PnP method. For this reason, the PCD method is used to measure the vision system's performances.

4.4 Adaptive Drilling

Following the previous sections where we instrumented the drilling end-effector, this section will explain the methodology leading to the development of the adaptive drilling algorithm. We will describe the design of experiment used to build our drilling dataset, data preprocessing, feature selection and models training and finally, its integration into a decision advisor for smart drilling.

4.4.1 Design of Experiment and Drilling Tests

The first step to train a machine learning model is to build the dataset. This dataset is built through drilling tests using different process parameters and recording the forces and vibration signals, as well as the hole quality metrics and tool wear. This section will describe the procedure used to perform drilling tests.

The only variable process parameters with the current drilling setup are the feed speed and the clamping force, respectively achieved by the hydraulic damper and the pneumatic cylinder previously presented. The feed speed is controlled by adjusting the hydraulic resistance on the hydraulic damper, and the clamping force is controlled by an analog signal sent to the proportional valve feeding the pneumatic cylinder with air. We define 3 levels for each: low, mid, high. The minimum and maximum values of the process parameters cover the complete possible range for drilling. The minimal clamping force is the lowest force applied to the workpiece during drilling without having the drill bit push back the end-effector, and the minimal feed speed is close to the highest resistance allowed by the hydraulic damper while still allowing movement. Similarly, the maximum values are respectively the maximum clamping force allowed by the pneumatic cylinder and the highest feed speed capable of drilling a thru hole in the workpiece.

The acquired force and vibration data are provided by the instruments presented in Section 4.2.3. The hole quality metrics are measured using a CMM. The tool wear was measured as the number of holes previously drilled by the drill bit.

In order to evaluate Exobrake's drilling performance, we use a "rigid" tool holder as a benchmark. A drilling system can be separated into two main bodies: the drilling tool and the tool holder. The tool is purely the drilling unit and the tool holder in the case of Exobrake is the cobot with the end-effector, excluding the drilling unit. By mounting the tool on a more rigid tool holder we can have a glimpse of the maximum performances achievable if Exobrake was as rigid as a CNC machine. The rigid tool holder is a milling machine whose arm holds the drilling unit, as shown in Figure 4.17.

Drilling data from the rigid setup follows the same procedure as that used in the base setup and will later be used as an upper limit in the adaptive drilling algorithm.

Before proceeding to drilling, we must use a robot configuration that allows us to drill following the surface's normal and with minimum slippage. Slippage occurs when applying the clamping force, causing a deviation from the nominal drilling point. To guarantee surface normality, we acquire the coordinates of 3 points on the workpiece's surface by probing them with the robot end-effector and use Equation 4.22 to find the workpiece's normal vector. The

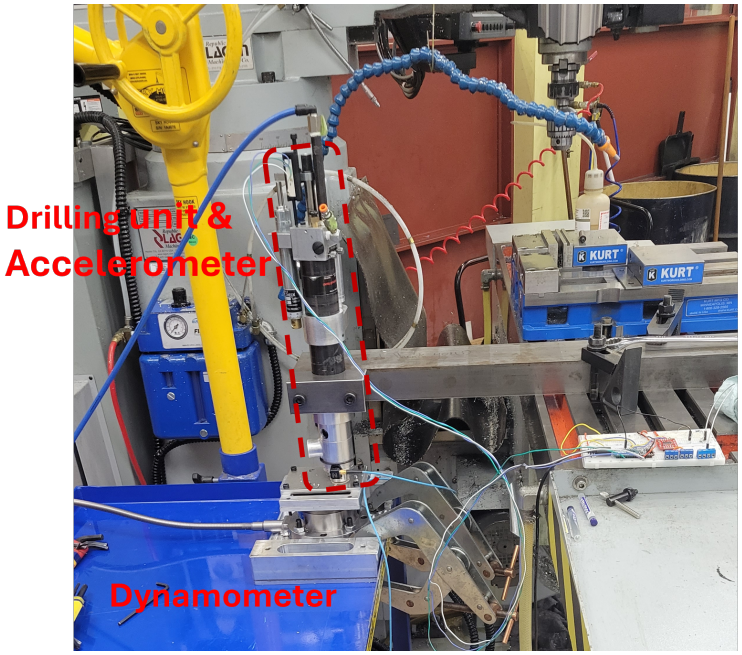


Figure 4.17 Rigid drilling setup used to record benchmark drilling data

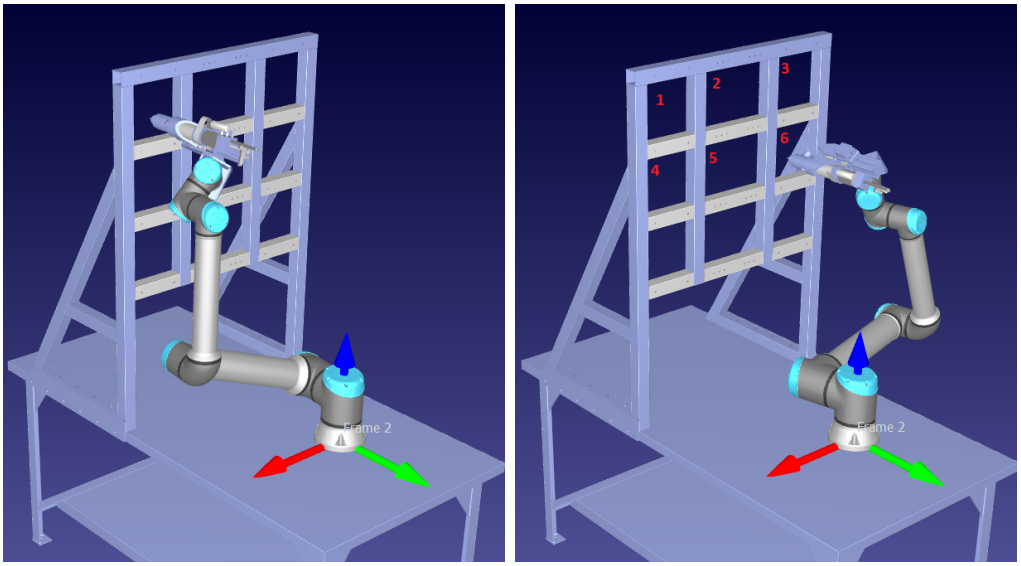
normal vector is then used to determine the orientation of the end-effector that allows for perpendicular drilling, whch will be used for the following tests. The vision-based normality detection method was not correctly implemented at that point of the project to be used. For slippage minimization, we explore 6 different sections on the drilling structure, each having 8 alternative robot configurations. The 6 sections are labelled in Figure 4.18 and their Cartesian coordinates w.r.t the robot frame are presented in Table 4.3.

Table 4.3 Drilling sections coordinates on the drilling test bench

Section	Coordinates [mm]		
	x	y	z
1	422	-588	1154
2	124	-588	1154
3	-173	-588	1154
4	422	-588	894
5	124	-588	894
6	-173	-588	894

These coordinates are based on an exact point probed on the physical test bench on which we applied offsets to get the other points, corresponding to the center of each section. We only used 6 points to discretize the test bench to have a general idea of the robot behavior on the different sections and reduce the computation and measurement time that will follow.

The robot configurations are generated by changing the wrist and elbow positions (elbow up/down and wrist up/down) and by using different ψ TCP orientations as described in Figures 4.9 and 4.8 directly on the physical robot to guarantee their feasibility. All robot configurations within a section share the same drilling position of Table 4.3. The feasibility of the initial pool of 48 configurations is verified using the robot's *Simulink* multi-body model to ensure all joints are solicited within the theoretical limits of the robot when applying 600 N of clamping force onto the workpiece; the invalid configurations are removed. Figure 4.18 shows examples of candidate robot poses.



(a) (b)

Figure 4.18 Example of candidate robot configurations for slippage minimization in RDK with labelled test bench sections. (a) Config. 1 (elbow and wrist down) on section 1 (b) Config. 32 (elbow down, wrist up) on section 6

To evaluate the slippage of each remaining configuration, we use the optical tracking system to measure the displacement of the end-effector during the application of the clamping force. The configuration with the smallest relative slippage is used for the following drilling tests. We proceed by sticking a target on the end-effector and on the drilling structure, which is used as a reference, as shown in Figure 4.19. We record the positional data of the EE target w.r.t the reference target while applying the clamping force. We then compare the before and after positions of the EE target to obtain a relative slippage measurement between all the remaining robot configurations. We only consider displacement in the workpiece's x-z plane defined by the reference target as shown in Figure 4.19. From the before and after EE

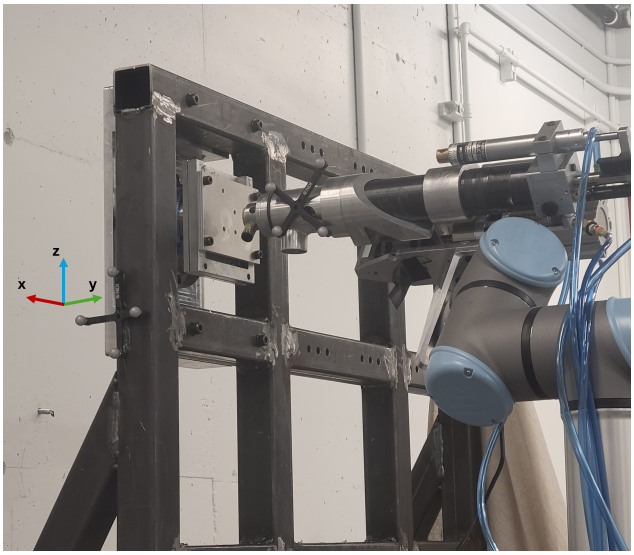


Figure 4.19 Optical tracking system's targets emplacements for slippage measurements during clamping tests, on robot Config. 1 and the reference frame used

target positions $\vec{p}_1$ and $\vec{p}_2$, the relative slippage s is calculated as follows:

$$\Delta\vec{p} = \vec{p}_2 - \vec{p}_1 \tag{4.45}$$

$$s = \sqrt{\Delta p_x^2 + \Delta p_z^2} \tag{4.46}$$

Absolute slippage can be measured to evaluate the precision of Exobrake by using peak drilling. Peak drilling is achieved by adjusting the stroke length of the drilling unit to leave only a small dot on the workpiece, instead of drilling a full thru hole. We peak the workpiece's surface before and after clamping and measure the distance between the two dots, which corresponds to the absolute deviation of the tool when applying the clamping force. For the same position on the workpiece, we compared the optimal robot configuration exhibiting the least relative slippage with a configuration that was neither adjusted for normal drilling nor for minimal slippage.

For the drilling tests, we drilled holes in 6 workpieces with Exobrake and in 1 workpiece with the rigid setup using 3 different drill bits in total. Each workpiece completed by Exobrake and the rigid setup has respectively 50 and 27 holes. With Exobrake, we change the clamping force and feed speed parameters between their 3 respective levels, resulting in 9 different process parameter combinations. On the rigid setup, we fix the feed parameter to its mid level but keep changing the clamping level. The feed speed is measured using the linear potentiometer, and only consider the speed when the drill bit is inside the workpiece. We

acquire the position data at a frequency of 20 Hz and apply a moving average filter to smooth the raw speed data. Using a window size of 5 data points showed good averaging performance while still utilizing local information as much as possible to avoid overgeneralization. The test procedure is as follows:

For a single drill bit:

1. Early life assessment: Using the rigid setup, we drill 3 holes with a brand new drill bit.
2. First workpiece: Using Exobrake, we drilled 50 holes.
3. Mid life assessment: Using the rigid setup, we drill 3 holes.
4. Second workpiece: Using Exobrake, we drill 50 holes.
5. End-life assessment: Using the rigid setup, we drill 3 holes.

We end up with 9 holes on the rigid setup and 100 holes completed by Exobrake. We only use 3 process parameters on the rigid setup because these tests are only used as assessments; the drilling dataset is built from the tests done using Exobrake. In total, with the 3 drill bits, 300 holes were drilled by Exobrake and 27 holes by the rigid setup, balanced among the different process parameters combinations.

4.4.2 Data Pre-processing and Features Extraction

Once raw operational forces and vibration signals are acquired, we need to perform data pre-processing before training a machine learning model on them. Filtering and feature extraction are the main types of pre-processing employed.

The data acquisition system only provides time domain force and acceleration signals, we use *Matlab*'s FFT functions to transform the data to the frequency domain for each channel. Next, we filter the raw data. To reduce noise, we apply on both acceleration and force signals a low-pass filter with a cutting frequency of 200Hz. We cut the signal beyond this frequency because it is much higher than the expected tool operation frequencies, so we don't lose important parts of the signal. The clamping force brings a bias on the force signals which is represented by a step corresponding to the clamping value in the time domain, and a low frequency spike at 0 hz in the frequency domain. To remove the implication of the clamping force, we apply on both force and acceleration signals a high-pass filter with a cutoff frequency of 40hz. This value is enough to eliminate all clamping force contributions while not affecting the important signal components brought by the drilling process. However, windowing can be used to ensure the robot frequencies don't get suppressed.

Once the operational signals are cleaned from noise and bias, we must extract features from them. The input data of the machine learning model are not currently in a usable format, hence the need for feature extraction. From the time and frequency domains of the data, we use *Matlab*'s signal processing library to extract the key features. A description of the extracted features is presented in Table 4.4.

Table 4.4 Description of the extracted signal features

Feature (channels)	Description
RMS_F (Fx, Fy, Fz, Mz)	RMS of force signals.
PeakValue_F (Fx, Fy, Fz, Mz)	Maximum absolute value in the time domain (forces).
Std_F (Fx, Fy, Fz, Mz)	Standard deviation of force signals.
MeanFrequency_F (Fx, Fy, Fz, Mz)	Spectral centroid of the force spectrum.
MeanFrequency_A (X, Y, Z)	Spectral centroid of the acceleration spectrum.
BandPower_F (Fx, Fy, Fz, Mz)	Total spectral power in the force spectrum.
BandPower_A (X, Y, Z)	Total spectral power in the acceleration spectrum.
PeakAmp_F (Fx, Fy, Fz, Mz)	Tallest FFT magnitude within 80–180 Hz (one-sided spectrum).
PeakAmp_A (X, Y, Z)	Tallest FFT magnitude within 80–180 Hz (one-sided spectrum).
PeakFreq_F (Fx, Fy, Fz, Mz)	Frequency (Hz) of the above peak within 80–180 Hz.
PeakFreq_A (X, Y, Z)	Frequency (Hz) of the above peak within 80–180 Hz.
SpecEnt_A (X, Y, Z)	Normalized spectral entropy of the acceleration PSD (spread/broadbandness).
TPF_Pwr_A (X, Y, Z)	Power around the TPF ($\pm 20\%$).
ChatPwr_Z (Z)	Chatter-band power in 150–600 Hz on the Z acceleration.
Fz_JerkRMS (Fz)	RMS of axial-force jerk (time derivative of F_z).
ForceRatios (FxFz, FyFz, MzFz)	Mean force ratios relative to axial force F_z.

These features provide a compact, interpretable summary of each signal's behaviour in both time and frequency. Time–domain metrics (RMS, PeakValue, Std) quantify overall intensity and variability, while spectral features (MeanFrequency, BandPower) indicate where energy is concentrated and how much of it is present. Band-limited peaks in 80–180 Hz capture the dominant periodic component relevant to cutting/chatter, and spectral entropy distinguishes narrow band, ordered content from broadband dynamics. Application-specific terms (TPF power, chatter-band power, jerk, and force ratios) connect the measurements to machining physics, improving separability for downstream machine-learning models.

Among the signal-specific features, we also include the process parameters (feed speed and clamping), the tool's age, and modified version of the process parameters involving mathematical operations (such as square roots, squared, feed speed × clamping etc), for a total of 61 features.

Finally, we filter the dataset to remove outliers. For outlier detection, we assume that data sharing the same process parameters should exhibit similar signal features. We use pairwise correlation based outlier detection: for each data point in a similar group, the Pearson correlation score is computed between its feature vector and those of all other points. The average of these correlation scores is then calculated for each data point. Data points with an average correlation below a predefined threshold of 0.8 are flagged as outliers, as they exhibit feature patterns that are less consistent with the rest of the group. This approach ensures that only data points representative of their respective process conditions are retained for further analysis.

4.4.3 Features Selection and Model Training

The hole quality regressor and the process parameters classifier use different methodologies; this section will describe each of them separately. These algorithms are dependent on a random seed; they were run 10 times using seeds from 43 to 52. This was necessary because, depending on the seed, some models would perform better than others, and model selection would have been biased if we only considered a single seed. The *sklearn* python library was used for machine learning.

The machine learning model is intended to be deployed for industrial drilling applications, which means losing access to the instrumentation material currently used to acquire the input data for the embedded sensors integrated on the end-effector. The PCB accelerometer can simply be replaced by the machining accelerometer mentioned in 4.3.3, but the dynamometer cannot completely be replaced by the load cell currently in use because it can't sense the Fx and Fy force components and the Mz moment. For this reason, the omission of the missing force components has also been investigated during feature selection. This was done by removing all features involving Fx, Fy or Mz. We then compared the performances of the models using the full feature set and the reduced feature set.

Hole Quality regressor

The first element of the adaptive drilling system is the hole quality regressor. It is meant to remove the need to inspect the workpiece in a CMM to measure the hole quality metrics.

Instead, we use machine learning to predict the hole quality of the just-drilled hole, given force and vibration signals and tool wear as inputs.

Having the maximum number of features for training is not always beneficial and can cause performance drops in a regression model. A feature selection step is required to filter the most important features that maximize the model's performance. The objectives of this section are to select the best subset of features and to find the best prediction model to use. To save computation time, all experiments were conducted using the diameter metric, assuming it provides a good representation of the other quality metrics. Only the final model is transposed onto the remaining quality metrics.

We investigated both filter-based and wrapper methods for feature selection. Filter-based methods use statistical tests to identify the importance orders of the features, followed by model training, as shown in the flowchart diagram in Figure 4.20. Wrapper methods are iterative algorithms that perform feature selection and model training simultaneously by training a model on different subsets of features.

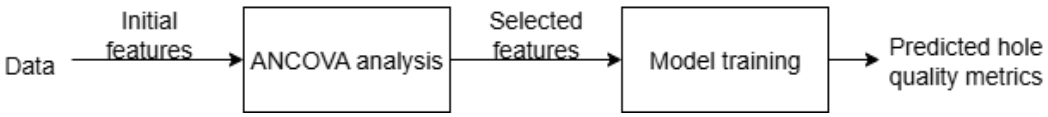


Figure 4.20 Filter-based method ML pipeline

For the filter-based feature selection, we used an ANCOVA analysis to rank the features based on their partial effects on the response. We use ANCOVA instead of ANOVA because our setup mixes categorical factors (process parameters) with continuous covariates (force and vibration features). We evaluate the unique contribution of each feature using an F-Test p-value. Concretely, the p-value represents how significant a feature is to the model's overall variance; a feature with $p < 0.05$ often means it is significant. We start with the 12 most significant features.

Even though ANCOVA assumes a linear model, we also fit non-linear models following feature selection. With the *sklearn* Python library, we use a 90/10 train/test split and employ grid search and 5-fold cross-validation to test different models and hyperparameters. For hyperparameter selection, multiple versions of the model with different hyerparameters values are evaluated on a validation set, which is a subset of the training set, in order to find the best performing hyperparameter values. On smaller datasets, subdividing the training set to include a validation set can be problematic, as it reduces the amount of data used for training once again. Cross-validation is a solution to this problem, as it divides the training set into multiple folds that are each used as the validation set at different iterations. The overall

cross-validation performance is the average of each fold's validation performance. Figure 4.21 shows the dataset subdivision for cross-validation.

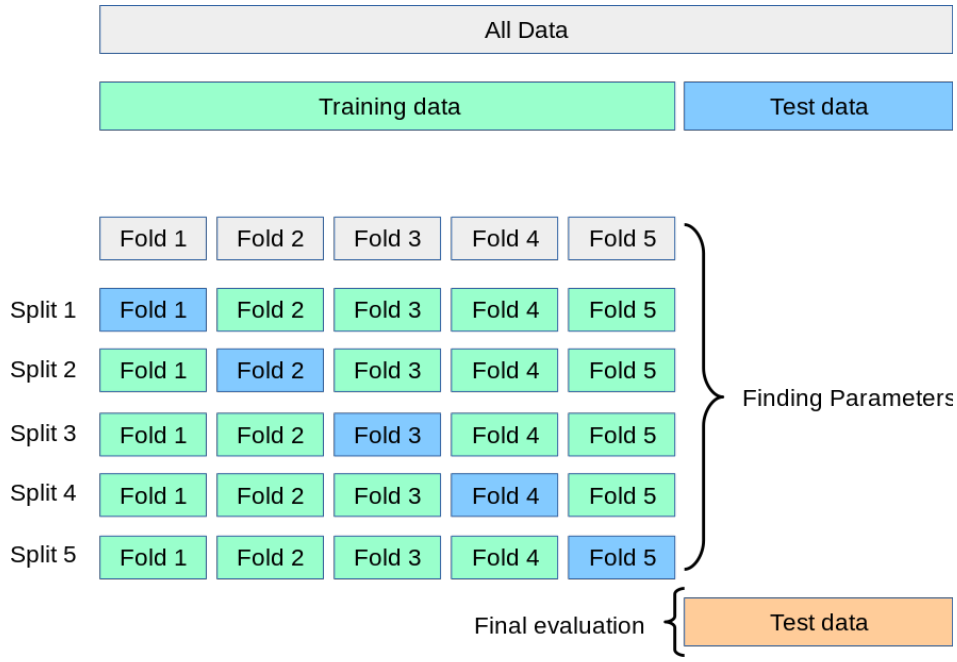


Figure 4.21 Representation of the dataset subdivision for cross-validation using 5 folds

The grid-search iterates through the following models and their respective hyperparameters:

- **Ordinary Least Squares (OLS)** — Baseline linear regression (no regularization).
  Hyperparameters: none.
- **Ridge Regression** — Linear model with L2 regularization to mitigate collinearity.
  Hyperparameters: `ridge__alpha = {0.1, 1, 10}`
- **Extended Trees (ET)** — Ensemble of extremely randomized trees (nonlinear, robust).
  Hyperparameters: `extratreesregressor__n_estimators = {200, 400}`; `extratreesregressor__max_depth = {None, 10}`
- **Random Forest (RF)** — Bagged decision trees; handles nonlinearities and interactions.
  Hyperparameters: `randomforestregressor__n_estimators = {200, 400}`; `randomforestregressor__max_depth = {None, 10}`

- **Gradient Boosting (GB)** — Stage-wise additive trees for tabular regression.
  Hyperparameters: `gradientboostingregressor__n_estimators = {200, 400}`; `gradientboostingregressor__learning_rate = {0.05, 0.1}`

- **PCA + (each base model)** — PCA inserted before the regressor to reduce dimensionality.
  Hyperparameters: `pca__n_components = {0.90, 0.95, 0.99}`
  (plus the corresponding base model grid above).

- **Stacking: ET + RF + GB → Ridge** — Two-level stack; Ridge fits out-of-fold base predictions.
  Hyperparameters: `stackingregressor__final_estimator__alpha = {0.1, 1, 10}`

- **Stacking: ET + RF → GB** — Two-level stack with GB as final estimator.
  Hyperparameters: `stackingregressor__final_estimator__n_estimators = {200, 400}`; `stackingregressor__final_estimator__learning_rate = {0.05, 0.1}`

- **Stacking: ET + GB → RF** — Two-level stack with RF as final estimator.
  Hyperparameters: `stackingregressor__final_estimator__n_estimators = {200, 400}`; `stackingregressor__final_estimator__max_depth = {None, 10}`

Also, despite PCA dimensionality reduction being a preprocessing method, we considered it as an hyperparameter that was indexed to the regression models (along with the model's own hyperparameters). Grid-search was iterating through these set of PCA hyperparameters: number of components = $\{0.9, 0.95, 0.99\}$. Since this is a regression problem, the best model selection is based on the validation set R^2, and its final performance is measured on the test set. Alongside the R^2, we also compute the RMSE and nRMSE, which correspond to the model's root mean squared error and its normalized form. The 3 evaluation metrics are calculated as follows:

$$SSE = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{4.47}$$

$$SST = \sum_{i=1}^{N} (y_i - \bar{y})^2 \tag{4.48}$$

$$R^2 = 1 - \frac{SSE}{SST} \tag{4.49}$$

$$RMSE = \sqrt{\frac{SSE}{N}} \tag{4.50}$$

$$nRMSE = \frac{RMSE}{y_{max} - y_{min}} \tag{4.51}$$

With SSE representing the sum of squared errors and SST representing the sum of squares total. R^2 represents how well a regression model fits reference points, the RMSE is the average model's error and the nRMSE is the RMSE normalized.

After the first round of grid search, we remove the least significant feature and learn the best model through grid search. This was performed 4 times, leaving 8 final features.

On the other hand, wrapper-based selection optimizes predictive performance rather than marginal significance. Unlike the filter-based strategy, the feature selection and model training are conducted simultaneously and iteratively: we train and evaluate the best model on a subset of features, modify this subset, and repeat until we can't improve the model's performance anymore. Model selection was also based on the validation set R^2, and the best model was evaluated on the test set. This method, being far more computationally expensive than the filter-based method, led us to consider only the extended tree model with its hyperparameters in the grid search, as it was the best performing model in filter-based learning. Three main strategies were investigated: forward feature selection, backward feature selection, and joint mutual information forward selection.

For forward feature selection, two versions of the algorithm were implemented. Both versions start with an empty set of features and iteratively evaluate the addition of each candidate feature using cross-validation R^2. In version 1, at each step, we only add the features that improve the model's performance over a predefined tolerance; if no feature meets this condition, the selection process stops, and the last feature set and hyperparameters are returned. In version 2, we greedily add features until all the features are added and return the best set among all.

Backward feature selection involves a similar procedure, but in the opposite order. We start with the full set of features and iteratively remove the least useful feature at each step,

following the same grid-search and cross-validation methodology as in the forward selection. Two versions of this algorithm were developed: greedy removal (v1) and early stopping (v2). In version 1, we remove the least significant feature regardless of whether the performances drop or remain the same, until only one feature remains. The feature subset yielding the best performance during the process is returned. In version 2, we remove the least significant feature only if it improves the CV performance by more than a predefined tolerance. If no feature removal yields an improvement, the process stops early and the final features subset is returned.

The last investigated wrapper method for feature selection was Joint Mutual Information (JMI) [32]. This is a greedy forward feature selection method aiming to maximize the relevance of targeted features to the target variable and minimize the redundancy among the selected features. At each iteration, the algorithm evaluates all the remaining candidate features and selects the one that maximizes the JMI criterion:

$$\text{JMI}(f) = I(f; y) - \frac{1}{|S|} \sum_{s \in S} I(f; s) \tag{4.52}$$

Where I(f; y) is the mutual information between the feature (f) and target (y) and I(f; s) is the mutual information between the feature and each selected feature (s) within the subset of the remaining features (S), each respectively representing the relevance and redundancy terms. The mutual information function I($\cdot$) is estimated with *sklearn.feature_section.mutual_info_regression* function. The described algorithms can be summarized through pseudocodes available in Appendix C.

Process Parameters Classifier

The second part of the adaptive drilling system consists of a process parameter monitor. In situations where an anomaly occurs during the drilling process affecting the drilling tool and modifying the process parameters, it is important to identify it. The role of the process parameters monitor is to provide feedback to identify whether the final drilled hole was conducted with the initially set process parameters (valid hole), or if these parameters have changed during the process (invalid hole). Machine learning was used to develop a classification model whose outputs are the predicted levels of each process parameter given the operational forces and vibrations signals. This anomaly detection system is an additional protection layer on top of direct process parameters monitoring through their own sensors and can be a solution for systems lacking proper process parameter sensors.

Unlike for the hole quality predictor, no feature selection was required, since the results were

good enough to skip this step, and PCA was enough most of the time. Data augmentation was used, it involves creating synthetic data from already existing data with added Gaussian noise; balancing follows by default to ensure that all classes have the same amount of data. It can be useful when the dataset has few data points per class. PCA reduces the dimensionality to express the dataset using the most significant features; it can be useful when there are a large number of features in the dataset.

Grid search was conducted to test different hyperparameters (including PCA), for each classifier. For this classification problem, we use the accuracy metric to select each model's hyperparameters using 5-fold cross validation. The final model performance was evaluated on a test set. Train/split ratio is 75/25. These are the evaluated models with their hyperparameters:

- **Support Vector Machine (SVM)** — Kernel-based classifier that finds optimal hyperplane for class separation.
  Hyperparameters: `clf__C = {0.1, 1, 10, 100};`
  `clf__kernel = {"linear", "rbf", "poly", "sigmoid"};`
  `clf__gamma = {"scale", "auto"};`
  `clf__degree = {2, 3, 4}`

- **Random Forest (RF)** — Ensemble of decision trees using bagging for handling non-linearities.
  Hyperparameters: `clf__n_estimators = {100, 300, 500};`
  `clf__max_depth = {None, 5, 10, 20};`
  `clf__max_features = {None, "sqrt", "log2", 0.5}`

- **Multi-layer Perceptron (MLP)** — Feed-forward neural network for classification.
  Hyperparameters: `clf__hidden_layer_sizes = {(50,), (100,), (100,50), (100,100,50)};`
  `clf__activation = {"relu", "tanh"};`
  `clf__alpha = {0.0001, 0.001, 0.01};`
  `clf__learning_rate_init = {0.001, 0.01, 0.1}`

- **Logistic Regression (LR)** — Linear classifier for probabilistic classification.
  Hyperparameters: `clf__C = {0.01, 0.1, 1, 10, 100, 1000, 10000}`

- **Decision Tree (DT)** — Tree-based classifier using recursive binary splitting.
  Hyperparameters: `clf__max_depth = {None, 3, 5, 10, 20};`
  `clf__min_samples_split = {2, 4, 6, 10};`

`clf__min_samples_leaf = {1, 2, 4};`
`clf__criterion = {"gini", "entropy", "log_loss"}`

- **Gradient Boosting (GB)** — Sequential ensemble of trees optimizing loss gradient.
  Hyperparameters: `clf__n_estimators = {100, 200, 400};`
  `clf__learning_rate = {0.03, 0.06, 0.1};`
  `clf__max_depth = {2, 3, 4}`

- **Linear Discriminant Analysis (LDA)** — Linear classifier using class-conditional densities.
  Hyperparameters: `clf__solver = {"svd", "lsqr", "eigen"};`
  `clf__shrinkage = {None, "auto", 0.1, 0.5}`

The addition of PCA iterated through these hyperparameters: number of components = 0.65, 0.7, 0.75, 0.80, 0.90, 0.95. Data augmentation would double the dataset and apply Gaussian noise with a spread coefficient of 0.02. The synthetic data are equally distributed between classes using the Synthetic Minority Over-sampling Technique (SMOTE) method.

Two approaches to this problem have been investigated: separate and joint approaches. The separate approach involves having a model for each process parameter and 3 classes, corresponding to the 3 levels used. The joint approach only requires a single multiclass model for both process parameters, with 9 different classes corresponding to the combinations between the process parameters (3^2). Training the models on a reduced feature set omitting Fx, Fy and Mz was also investigated.

Finally, the selected models are packed into .pkl pickle files that include the feature subset to use and the trained models that can be deployed by a tier program. Those files will be loaded by the adaptive drilling algorithm for our case.

4.4.4 Adaptive Drilling Algorithm

Once the hole quality predictor and the process parameters classifier are trained, we integrate them into the adaptive drilling algorithm. As its name implies, its goal is to adjust the process parameters to maintain hole quality within specifications. It uses predictions from the hole quality predictor and a decision engine to provide the set of process parameters to adopt for the future hole, given the information provided by the present hole. The overall architecture of the adaptive drilling algorithm is shown in Figure 4.22 and the remainder of this section will explain each element.

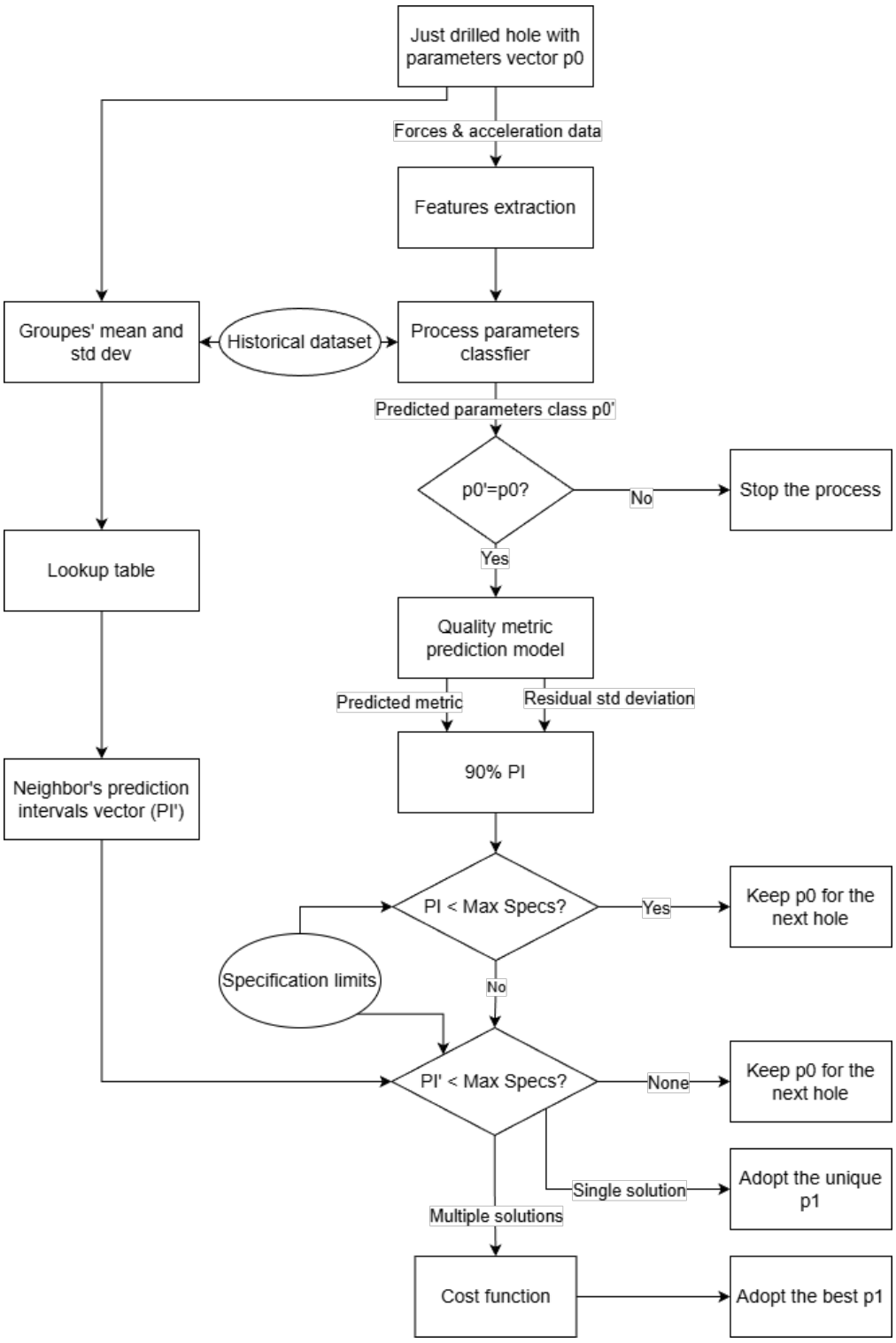


Figure 4.22 Adaptive drilling algorithm architecture

Before starting, this algorithm requires the following information: a historical drilling dataset

and hole quality specification limits. This algorithm assumes a normal distribution of the drilling data and needs a dataset that supports this distribution. The dataset used for model training can be utilized as the historical dataset, and new data can be added following their inspection in a CMM. The hole quality specification limits are predefined limits set by the user; in our case, we used the worst hole quality metrics achieved by the rigid drilling setup.

The initial hole is performed using its initial set of process parameters p0 and we acquire the operational force and vibration signals. From these data, we extract the features and feed them into the process parameters classifier. If the classifier identifies different process parameters than those initially set, we interrupt the drilling routine. Otherwise, the just-drilled data are fed into the hole quality metrics regression model. In parallel, we also feed the historical dataset into the prediction model to get the model's prediction interval. This is achieved by comparing the predicted values to their ground truth. The model's 90% prediction interval can be calculated from the following equations:

$$\vec{PI}_{model} = \hat{\vec{y}} + 1.64\vec{\sigma}_{emp} \tag{4.53}$$

$$\vec{\sigma}_{emp} = \sqrt{\frac{\sum_{i=1}^{N}(\vec{y}_i - \hat{\vec{y}}_i)^2}{N}} \tag{4.54}$$

With $\hat{\vec{y}}$ the prediction vector given by the model and $\vec{y}$ its ground truth for a dataset of size N. We only consider the upper prediction intervals because we have only an upper limit on the quality metrics.

The historical drilling dataset is also used to build the process lookup table, as shown in Table 4.5. This lookup table stores each group's mean and standard deviation, taken from the historical drilling dataset:

$$\mu_{CD} = \mathbb{E}[y|C, D] \tag{4.55}$$

$$\sigma_{CD} = std[y|C, D] \tag{4.56}$$

We can define the lookup table's 90% prediction intervals:

$$PI_{CD} = \mu_{CD} + 1.64\sigma_{CD} \tag{4.57}$$

Table 4.5 Process lookup table containing each clamping (C) and feed speed (D) groups from their 3 levels (low, mid, high)

	C_L	C_M	C_H
D_L	μ_{LL}, σ_{LL}	μ_{LM}, σ_{LM}	μ_{LH}, σ_{LH}
D_M	μ_{ML}, σ_{ML}	μ_{MM}, σ_{MM}	μ_{MH}, σ_{MH}
D_H	μ_{HL}, σ_{HL}	μ_{HM}, σ_{HM}	μ_{HH}, σ_{HL}

On the lookup table, we define a groupe's direct neighbors as the groupes resulting in a single parameter change, as opposed to diagonal neighbors requiring both parameters changes. A groupe has 4 neighbors in the center, 3 on the borders, and 2 in corners.

The first decision checkpoint is used to verify whether the current process parameters can be maintained. We compare the specification limits to the model's prediction interval of Equation 4.53 to answer the following question: "If I keep the current process parameters, can I be certain at 90% that the following hole will be within the specification limits?". If the condition is met, p0 is kept for the next hole. Otherwise, we seek possible alternatives in the neighboring process parameters stored in the lookup table. We use the same logic to compare the specification limits to the neighbor's prediction intervals of Equation 4.57. If there is no solution, we keep the same process parameters. Otherwise, we adopt the unique solution, or in the case of multiple solutions, we evaluate each of them in a cost function and keep the solution with the lowest expected hole quality metric relative to specification limits $\vec{\mu}_{max}$:

$$J(p) = \sum_{k=1}^{K} \frac{\mu_{k,p}}{\mu_{max,k}} \quad \forall\, k = \{1, ..., K\},\, p = \{1, ..., P\} \tag{4.58}$$

$$p1 = \arg\min_{p} J(p) \tag{4.59}$$

With P as the candidate solutions vector size and $K = 4$ as the number of quality metrics considered.

Alternatively, instead of using a lookup table and cost function evaluation, we can refer to a success table that stores the per-groupe success rates to select the groupe with the highest success probability. A success in this context occurs when all the specification limits are respected. The initial guess of p0 can also be based on this success table.

Finally, the adaptive drilling system was tested on random data points taken from a drilling dataset that was previously unknown to the models. 100 experiments were randomly con-

ducted to obtain the percentage of success for each metric. We define success as the algorithm either keeping an already valid process parameter set p0, or finding an adapted process parameter set p1.

CHAPTER 5 DEVELOPMENTS AND RESULTS

Following the methodology presented in the previous chapter, this section will present the upcoming developments and results.

5.1 Experimental Setup Validation

Following the FEA analysis on the rigid structure's CAD, we measure a displacement of 7.67μm at the emplacement of the clamping force, which we consider small enough to validate R1. Figure A.1 in Appendix A shows the deformation pattern of the rigid structure, the exaggerated deformation pattern illustrated is consistent with the expected behavior of the rigid structure during drilling. As of R2, this is met by adopting a grid-like shape and various mounting holes allowing for a flexible installation of the dynamometer spanning the whole surface.

From a stiffness perspective, the measured deflection of 7.67 μm under a 600N clamping load is negligible compared to the compliance of the UR10e robot itself. Reported cobot stiffness values of $3 - 4.5 \times 10^4 N/m$ [9] indicate that most of the compliance in cobotic drilling comes from the robot's structure rather than external fixtures. Ensuring a quasi-rigid test bench follows best practices in robotics manufacturing, where a stiff support is assumed to identify the robot-tool FRF [19,20]. In that sense, the rigid structure validated through FEA created similar conditions to those used in FRF identification for milling [19, 20], but for cobotic drilling.

5.2 End-Effector Developments

5.2.1 TCP Positioning

The surrogate optimization algorithm provided the optimal TCP emplacement w.r.t the robot flange through the following solution vector:

$$\vec{x} = [4.048, -0.38, 4.90, 0.293, 0.063, -0.37, 2.12, 0.89]$$

Following the demonstration starting at Equation 4.14, we retrieve the corresponding non-adjusted transformation matrix that defines the TCP pose:

$$\mathbf{T'}_{flange}^{TCP} = \begin{bmatrix} 0.34 & 0.59 & -0.73 & 0.29 \\ 0.94 & -0.18 & 0.29 & 0 \\ 0.04 & -0.79 & -0.62 & 0.0638 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The matrix was adjusted following CAD simplifications, such as the rounding of dimension and angle values, to simplify parts manufacturing. After the practical adjustments:

$$\mathbf{T}_{flange}^{TCP} = \begin{bmatrix} 0.79 & 0.26 & -0.56 & 0.277 \\ 0.58 & -0.61 & 0.54 & -0.134 \\ -0.2 & -0.75 & -0.63 & 0.074 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Exact TCP measurement on the physical robot has been conducted using the TCP calibration routine from the robot's teach pendant, based on point probing.

The optimization also computes the optimal robot poses. Among the 9 points defined in the panel, we choose the middle point, point 5:

$$\vec{q'} = [-13.6, 192.1, -288.4, 13.8, -52.5, -175.4]^\circ$$

And after applying some adjustments to the physical setup:

$$\vec{q} = [-8.19, -168.6, 74.17, 3.08, -53.02, -143.3]^\circ$$

5.2.2 Structural Performances

Figure 5.1 displays the final exploded view of the end-effector designed within the scope of this project; parentheses notation will be used to reference part numbers.

Table 5.1 identifies each of ALCCE's parts.

The machined parts of the end effector are in aluminum Al6061, which is known for its low density and often used in aerospace. The mass minimization strategy involving lightning holes is present in the design of the rail supports and sensors to drilling unit parts, as seen in Figure 5.2.

The fasteners pattern on both sides of the rail support part is meant to attach to the flange bracket part and is solicited in shear. We added the lightening hole on both sides for symmetry purposes. The remaining hole is there for the pneumatic cylinder's (5) rod. The fastener

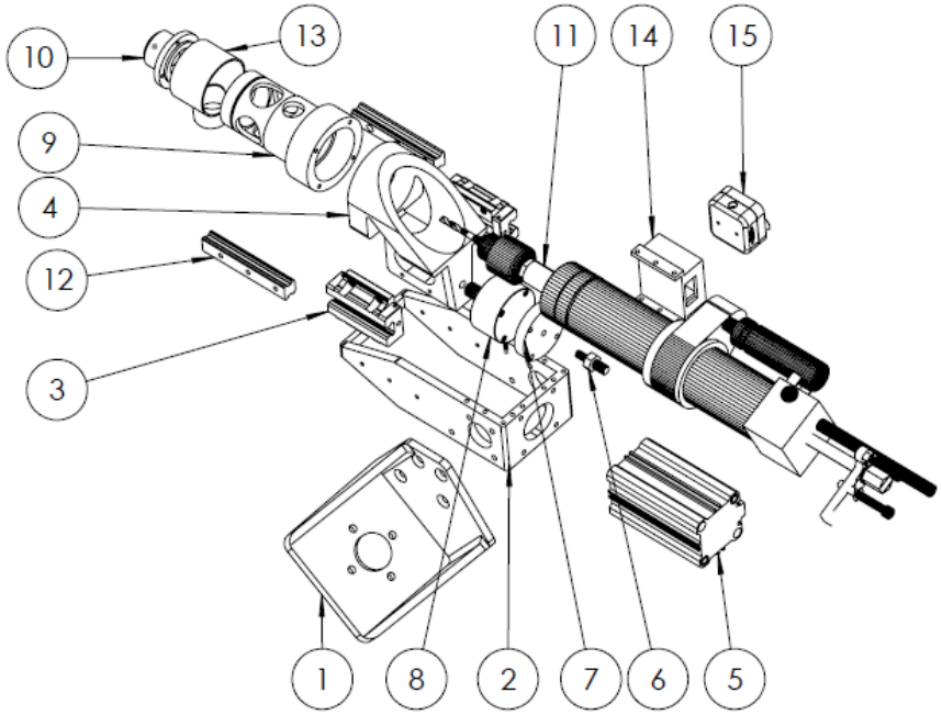
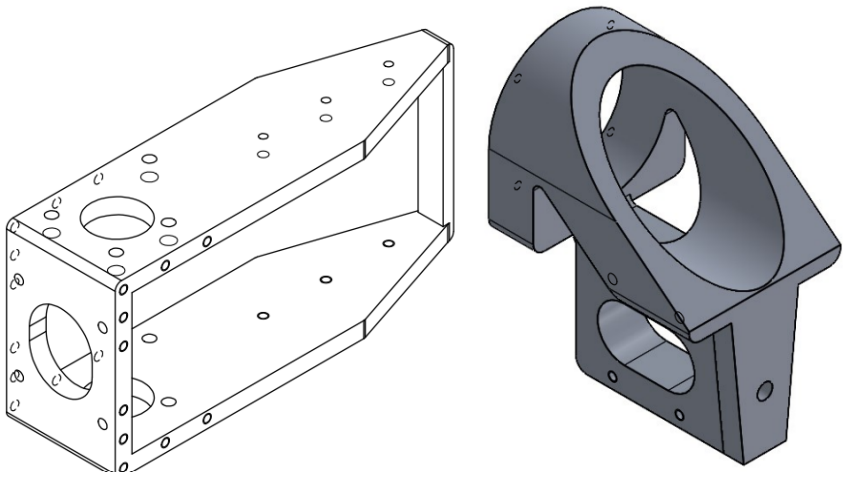


Figure 5.1 The designed ALCCE end-effector's exploded view. The parts are labelled and identified in Table 5.1.

Table 5.1 ALCCE parts identification from Figure 5.1

Part number	Description
1	Flange bracket
2	Main frame
3	Ball bearing carriage
4	Loadcell holder
5	Locking cylinder
6	M6x1 to M8x1.25 adapter
7	Loadcell adapter
8	Loadcell
9	Clamping nose
10	Clamping head
11	Drilling unit
12	Linear rail
13	Vacuum interface
14	Camera support
15	Camera



(a) (b)

Figure 5.2 End-effector parts designed with lightening holes: (a) Main frame (2), (b) Loadcell holder (4)

holes pattern on the sensor to drilling unit part is meant to attach to the linear carriages (3) so we added a lightening hole inside the pattern since this surface is also solicited in shear.

Finally, we respect **O3** by measuring a fully assembled end-effector mass of 7.11 kg, below the 8.2 kg limit set at a CoG offset of 300 mm. The final measured CoG coordinates:

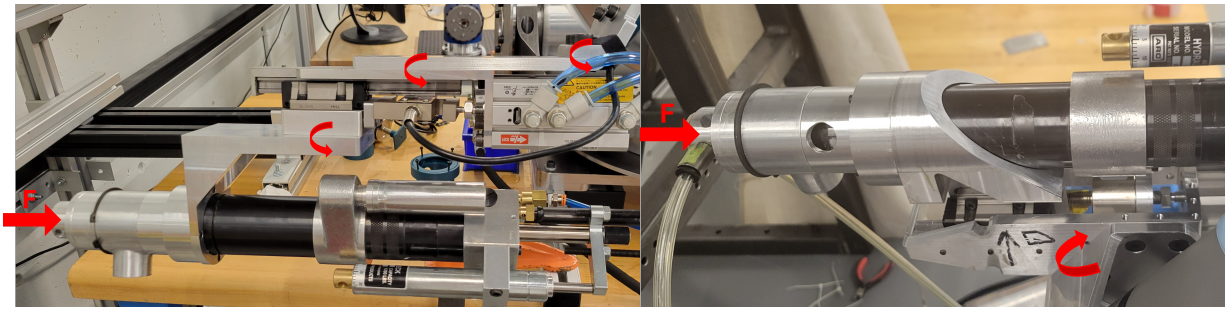
$$CoG = (107, 16, -55)mm$$

results in an absolute offset of 121 mm, corresponding to a final payload capacity of 12 kg, leaving a significant margin.

The mass of ALCCEE is significantly lower than that of industrial end-effectors such as ALEMA, FDS or GKN's riveting drilling-riveting units [1–3, 7, 8]. In these industrial examples, the robot's high stiffness and payload allow for the design of heavier end-effectors, whereas the low payload of a UR10e limits the mass of the end-effector.

Structure rigidity improvements are reflected in the design of the latest end-effector version. Comparing version 2.0 and 3.0, we reduced the number of fragile joints from 3 to 1 as shown in Figure 5.3

The final upgrade was to use the linear rail and carriage as a pair, as shown in the exploded view in Figure 5.1. By having two ball bearing carriages in parallel, we improve the end-effector rigidity by doubling their static load capacity in the transverse directions.



(a) (b)

Figure 5.3 Joints affected by bending efforts on versions (a) 2.0 and (b) 3.0 of the end-effector

5.2.3 Sensor and Actuators

Feed Sensing Feed sensing is achieved by using a linear potentiometer mounted on the moving part of the drilling unit using a 3D printed bracket, as previously shown in Figure 4.11.

Drilling rotation speed sensing A working implementation of rotation speed sensing was not achieved in the project. The first option, involving laser sensing, was tested. Figure 5.4 shows an example of the digital signal sensed by the laser sensor during dry drilling.

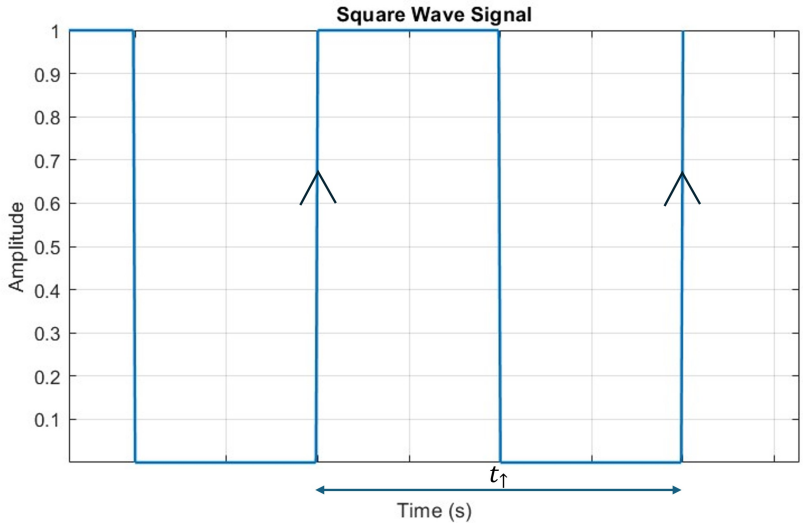


Figure 5.4 Example of the laser sensor's square signal used for rotation speed measurement. $t_\uparrow$ is the period, measured between two rising edges

However, the problem with that solution is that it only works for dry drilling scenarios, as

the reflective band does not hold when lubricant is projected onto the spindle.

While real-time feedback was not implemented, offline analysis of the accelerometer data (Figure 5.5) validates the feasibility of this method, clearly showing the rotation frequency peak at 106.36 Hz (which corresponds to twice the tool frequency since we use a drill bit with two edges), despite the use of an MQL system.

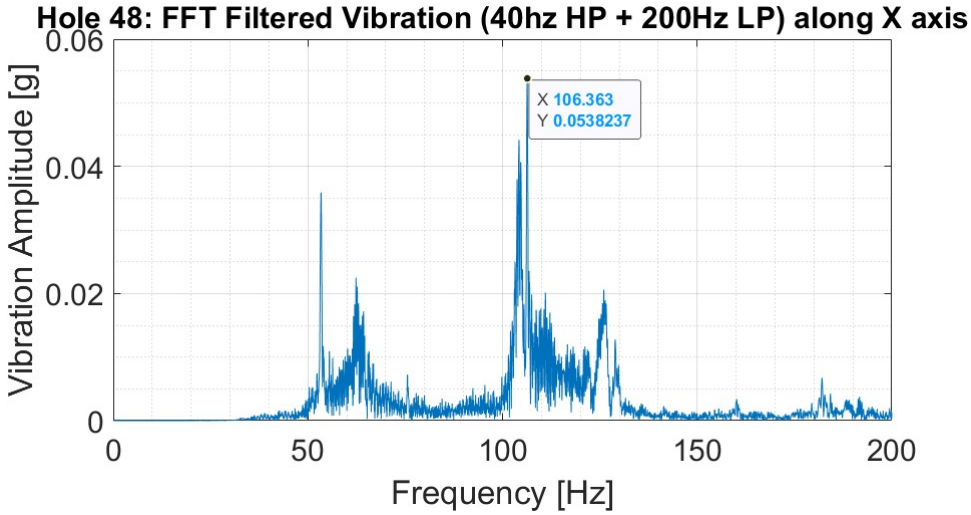


Figure 5.5 Example of tool rotation frequency measurement through FFT analysis

Integrating a machining accelerometer like the EVAL-ADXL1003Z on the end-effector and using a real-time FFT algorithm could, in theory, provide rotation speed data using the described methodology.

Force sensing The emplacement of the load cell shown in Figure 5.6 allows for clamping and thrust force measurements.

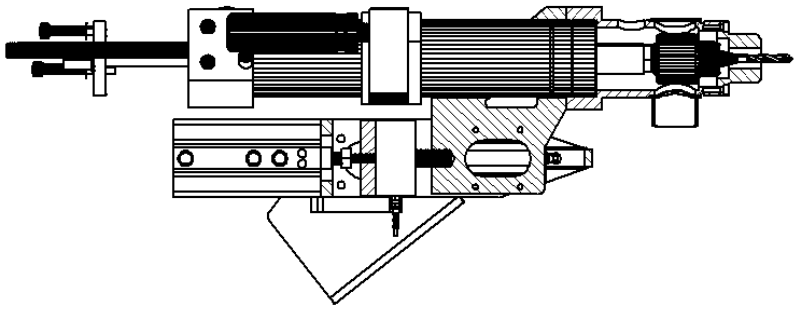


Figure 5.6 End-effector cut view

Clamping actuator The clamping force is provided by the pneumatic cylinder (5) meant to push the clamping nose against the workpiece. The pneumatic cylinder model *CDLQB40-25D-B* is used because of its ability to apply the required clamping force of 600N and locking capability to hold the clamping force. The pneumatic circuit runs on 100PSI, the cylinder's maximal air pressure is 145PSI. In theory, it can output 825N of thrust force at 73PSI and its lock can withstand a maximum of 629N of holding force. However, in practice, the maximum measured clamping force is 610N.

Lubrication and chip evacuation The portable MQL *Ecolubric MQL HSP-1 Booster* from *Accu Svenska AB* from Figure 5.7 provides the necessary lubrication for drilling. Even though the lubrication parameters have an effect on the drilling parameters, we kept this process variable constant. The lubrication parameters were arbitrarily chosen as the provided lubrication was judged to be enough for drilling. The MQL system's hoses are screwed to the drilling nose (10) part, which has carved tunnels to direct the lubricant towards the drill bit.

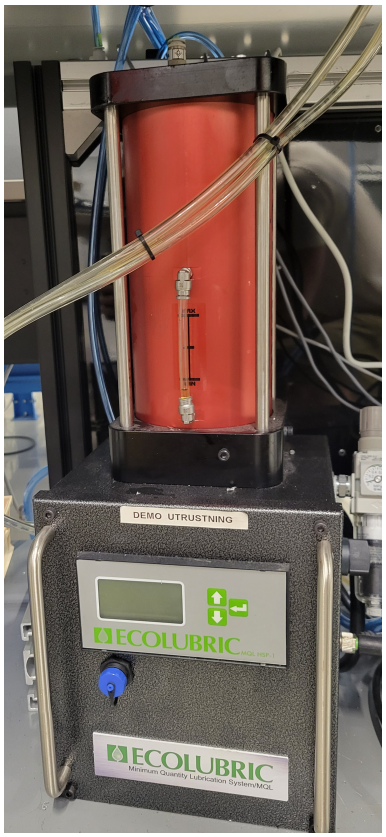


Figure 5.7 *Ecolubric* Portable MQL system used for drilling

The high chips production rate caused by drilling can be harmful to the tool as it is contained

in a small space in the drilling nose, hence the need for a vacuum system. The vacuum is a commercially available industrial vacuum and connects to the drilling nose through the vacuum connector (13) part, which rotates freely around the nose to accommodate different orientations of the end-effector.

A dynamometer, an accelerometer and process sensors used in this project are standard in multi-sensor monitoring approaches to acquire the main physical quantities commonly used for tool wear monitoring: thrust force, cutting torque, and vibration [5, 21–26, 28–30].

5.2.4 Vision System Results

The following section will present the results and developments of the vision system following the methodology previously presented.

Camera installation As shown in Figure 5.1 and Figure 5.8, the camera (15) is mounted on the side of the end-effector by using a 3D printed support (14). This sideway configuration keeps the whole field of view of the camera unobstructed and is low-profile.

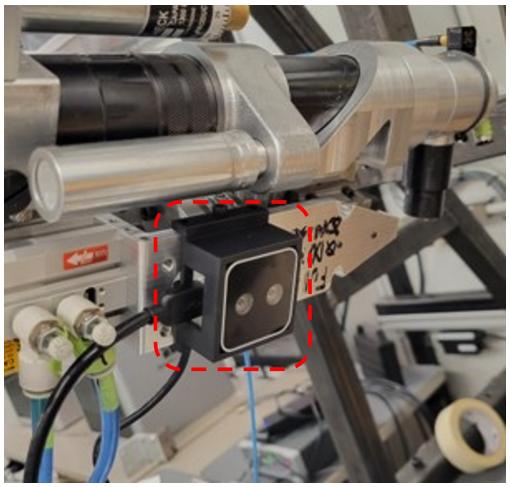
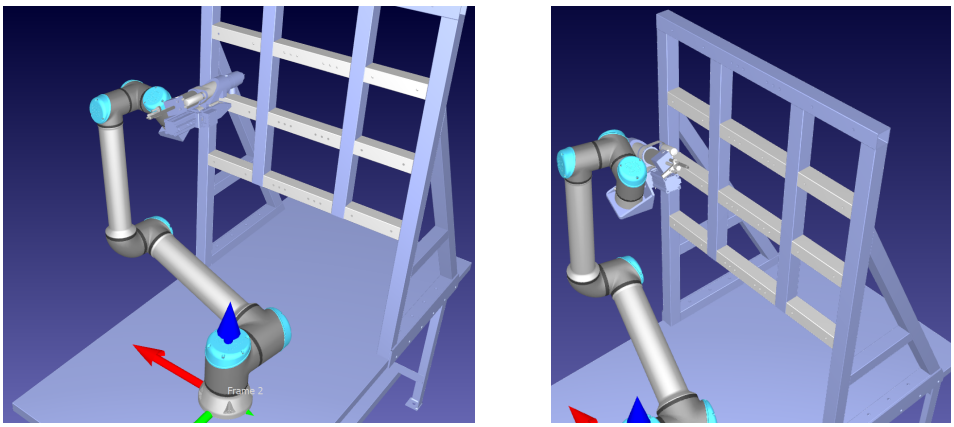


Figure 5.8 Stereo camera on the end-effector

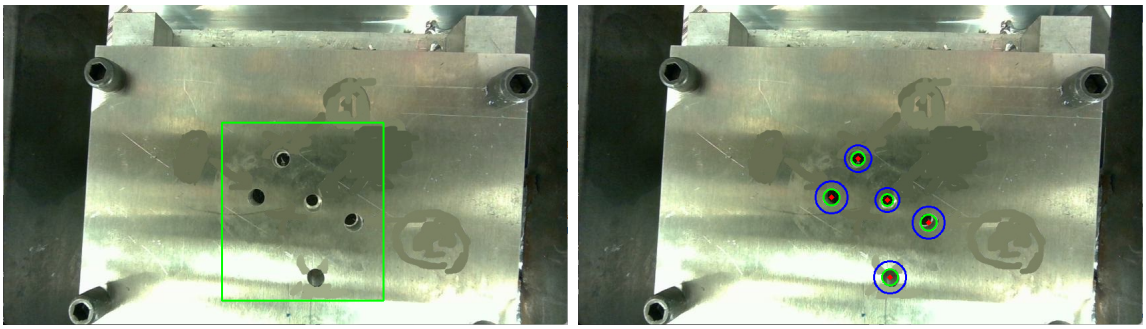
This camera emplacement requires the end-effector to rotate sideways from the direction of drilling to capture an image of the workpiece, as demonstrated in the *RoboDK* model of Figure 5.9.

Holes detection Figure 5.10 shows the zone of interest defined by the user and the resulting detected holes needed to compute the surface normality.



(a) (b)

Figure 5.9 Configuration change for camera capture, from the *RoboDK* model (a) Drilling configuration (b) Camera capture configuration



(a) (b)

Figure 5.10 Holes detection on the test image. (a) Bounding box set by the user. (b) Detected holes. In green the nominal hole and in blue the augmented hole.

Surface normality detection Figure 5.11 shows the 3D pointcloud with the circumference points of each hole and their normal vector, as well as the reference normal vector.

In Figure 5.11 we can see blank spots on the pointcloud, meaning the surface was overexposed. This problem can be fixed by controlling the local lighting conditions or by adding a matte coat of paint to the workpiece. Despite this, the program can still correctly detect the holes as seen in Figure 5.10b. Table 5.2 shows the orientation error $\Delta\alpha$ of each hole normal vector with the reference, as well as the number of final candidate points used N.

The normality detection shows an average error of 0.75° with the reference normal, which is below the limit set at the beginning of this section. Also, if we make that limit stricter and

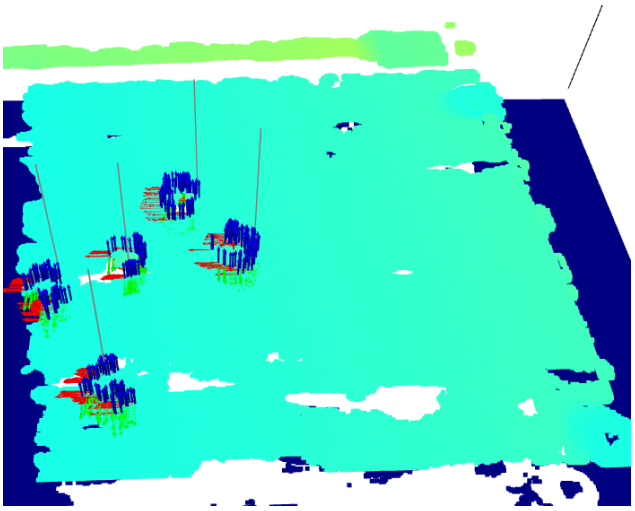


Figure 5.11 Pointcloud of the test workpiece. The black lines are the normal vectors and the smaller frames around them are the candidate points used for normality detection

Table 5.2 Normality detection results. The orientation error $\Delta\alpha$ and the number of candidate points used N are presented for each hole

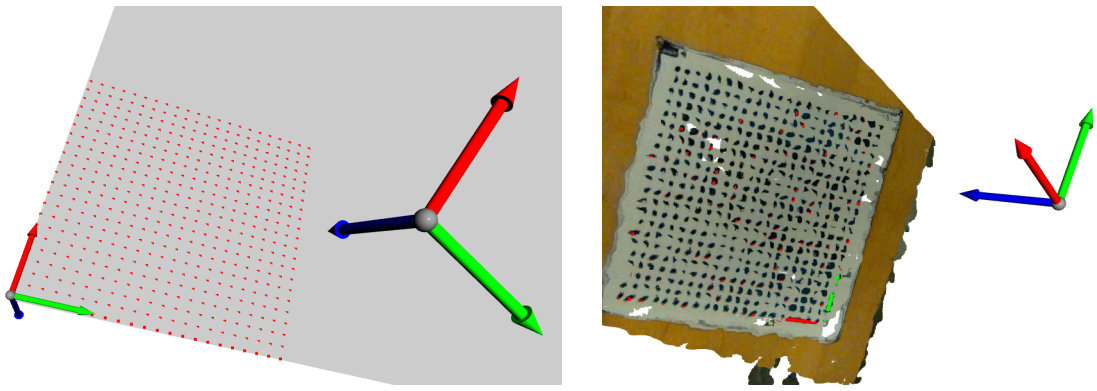
Hole #	$\Delta\alpha$ [°]	N
1	0.67	87
2	1.05	72
3	1.03	104
4	0.8	67
5	0.18	88
AVERAGE	0.75	

consider the individual value of each hole rather than the average, results show that only holes 2 and 3 exceed the 1 ° limit, but only by 5% and 3%, which is negligible. The normality detection algorithm was applied locally around holes, which is a more complicated task than on a smooth surface due to the possible points detected inside the holes. This means that the normality detection algorithm should perform better in the simpler scenario for which it was designed to be used.

Hand-eye calibration The developed PCD based hand-eye calibration methods are based on the projection of the grid points onto a fitted perfect plane. Figure 5.12 shows the projected calibration points overlayed with the original colored pointcloud.

The projected grid pattern and the fitted plane are represented in Figure 5.12.

From Figure 5.12b we see a good alignment between the projected dots and the real dots



(a) (b)

Figure 5.12 Calibration grid 3D projection. (a) Dots pattern projected on the fitted plane with the camera frame and the grid's frame. (b) Projected dots pattern along with the original pointcloud

from the colored pointcloud. Despite the lack of metrics to measure point alignment, we consider this result good enough to proceed with the next steps.

We use *RoboDK* to visualize the resulting transformation matrix provided by the different hand-eye calibration methods. An example using method 1 is displayed in Figure 5.13.

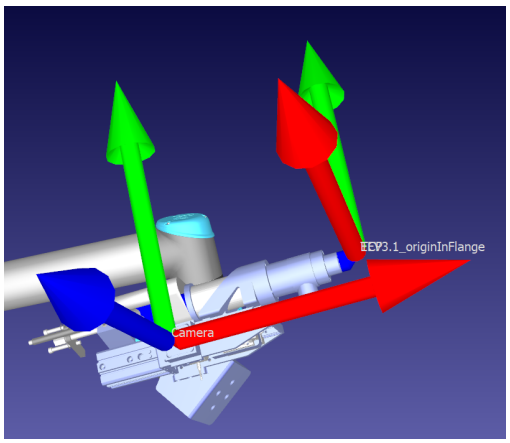


Figure 5.13 Camera frame position on the EE after hand-eye calibration.

We can notice that the position of the camera frame is aligned with the true position of the camera in the CAD. All hand-eye calibration methods give a similar representation.

Finally, Table 5.3 presents the per-axis precision and accuracy results for each hand-eye calibration method.

Results show that the hand-eye calibration method 3 has the best accuracy with an absolute

Table 5.3 Per axis accuracy and precision results of the different hand-eye calibration methods

Calibration method	Accuracy [mm]				Precision RMS [mm]			
	x	y	z	norm	x	y	z	norm
1	-1.28	5.45	-1.48	5.79	1.01	1.67	1.64	2.55
2	-7.36	5.38	4.57	10.20	1.04	0.82	1.90	2.31
3	2.15	0.82	-3.08	3.84	1.00	1.77	1.23	2.38
4	5.21	-0.95	-4.23	6.78	1.42	1.92	1.64	2.90

error of 3.84 mm and an absolute precision RMS of 2.38 mm. We can notice that method 2 has the best precision with 2.31 mm. However, a slight increase in precision comes with a large reduction in accuracy, with an error of up to 10.2 mm. Therefore, calibration method 3 is recommended. We also notice that PnP-based methods perform better than the developed PCD-based methods. This leaves room for future improvement to fully take advantage of the 3D capabilities of stereo cameras for hand-eye calibration. To conclude, the results in Table 5.3 and 5.2 validate **O1** and **O2**.

5.3 Adaptive Drilling Results

This section will showcase the results obtained through the development of the adaptive drilling system. This will cover the design of experiment and tests, data preprocessing, features selection and models training and the adaptive drilling system itself.

5.3.1 Design of Experiment and Drilling Tests

The clamping force and feed speed levels, which were discretized into 3 values each, were measured respectively using the built-in end-effector load-cell and the linear potentiometer previously presented. The measured value of each level is presented in Table 5.4.

Table 5.4 Process parameters values corresponding to each level

	Feed speed (mm/s)	Clamping force (kgf)
LOW	1	30
MID	2.6	45
HIGH	5.2	60

The choice of feed speed and clamping force as primary experimental factors is consistent with studies on multi-material stacks [14–17, 21–24]. These works show that adequate clamping (500-1500 N) and feed speed can limit interface damage and reduce dimensional errors. Our

DOE focuses on a reduced clamping range (300 - 600 N), adapted to aluminum plates and the payload limits of the UR10e, while also exploring the lower end of the recommended clamping values.

Figure 5.14 shows an example of the position and speed profiles of the drilling unit during a mid feed speed drilling test.

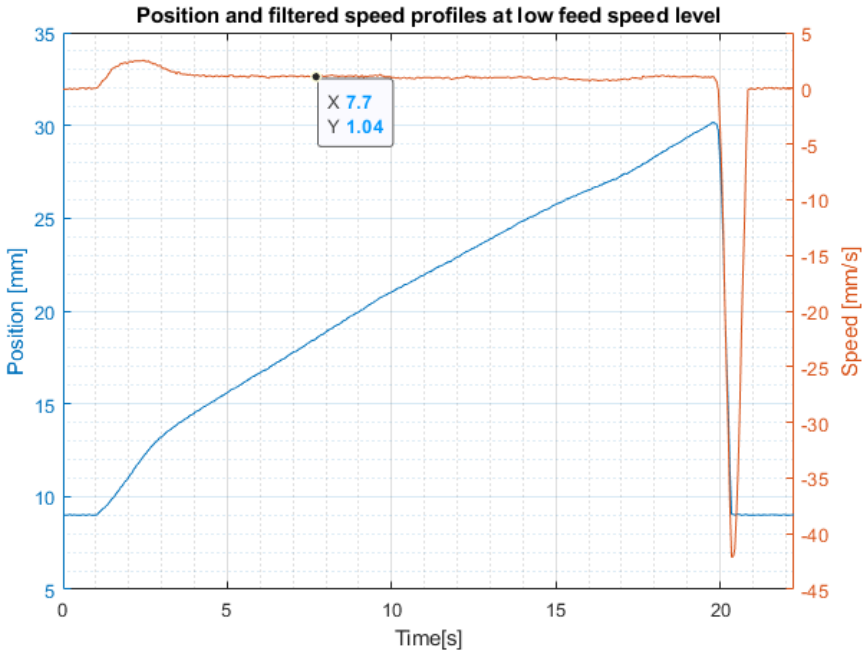


Figure 5.14 Low feed speed level position and speed profiles during a drilling test, measured by the linear potentiometer

We see that the first part of the movement (approach phase) has a higher slope in position than the following part (contact phase) because the tool is slowed down by the material as it penetrates the workpiece. Similarly, a closer look at the raw speed data is shown in Figure 5.15.

We can see that the filtered data are a smoothed version of the raw data and that the window of 5 data points preserves the overall signal information.

On the other hand, we compared the axial force sensed by the load-cell and the dynamometer in Figure 5.16.

We can see in Figure 5.16 that the clamping axial force measured by the two sensors can display different results and does not reach the targeted clamping force. This offset and the force data are consistent across the clamping force levels, meaning they will not harm the dataset later used for ML models training. The offset may be caused by the miscalibration



Figure 5.15 Raw and filtered speed profiles at low feed speed level

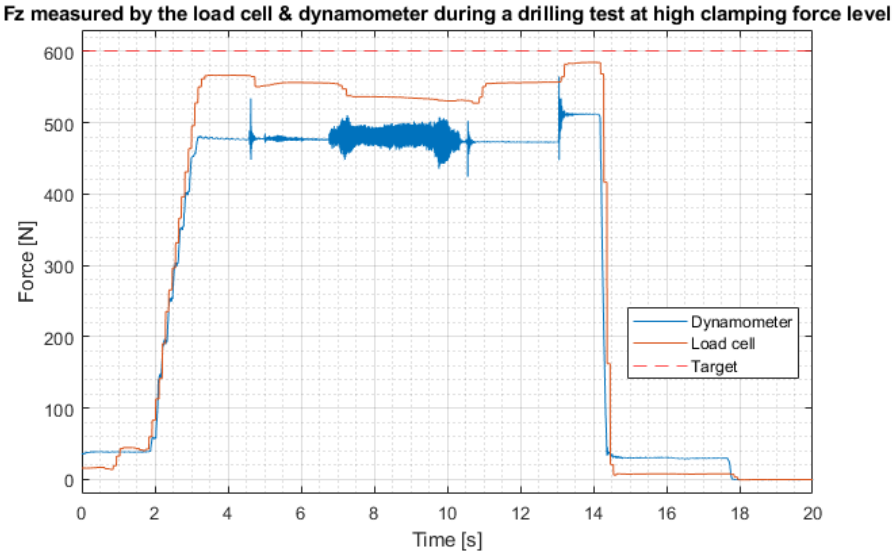


Figure 5.16 Axial force measured by the dynamometer and the load cell during a high clamping force drilling test of Hole 38 in Plate 1.

of the load cell, as it was not calibrated to match the dynamometer. Better matching can be obtained by calibrating the loadcell directly from the axial force data provided by the dynamometer, instead of using free weights and an industrial scale.

During the slippage optimization step, we first adjusted the robot pose orientation to be normal to the surface. Point probing on the workpiece resulted in the following robot poses

used to compute the surface normal vector:

$$P_1 = [0.418, -0.595, 0.916, 0.286, 2.325, 2.308]$$
$$P_2 = [0.386, -0.596, 0.884, 0.286, 2.325, 2.308]$$
$$P_3 = [0.426, -0.594, 0.876, 0.301, 2.355, 2.329]$$

The first 3 elements, being the x-y-z point coordinates, we apply Equation 4.22 to get the following normal vector:

$$\vec{n} = [-0.0435, 0.9991, 0.0012]$$

After evaluating the 48 manually generated robot configurations on the multibody model, 34 were kept for relative slippage evaluation. Figure 5.17 shows an example of the position data recorded by the laser tracker.

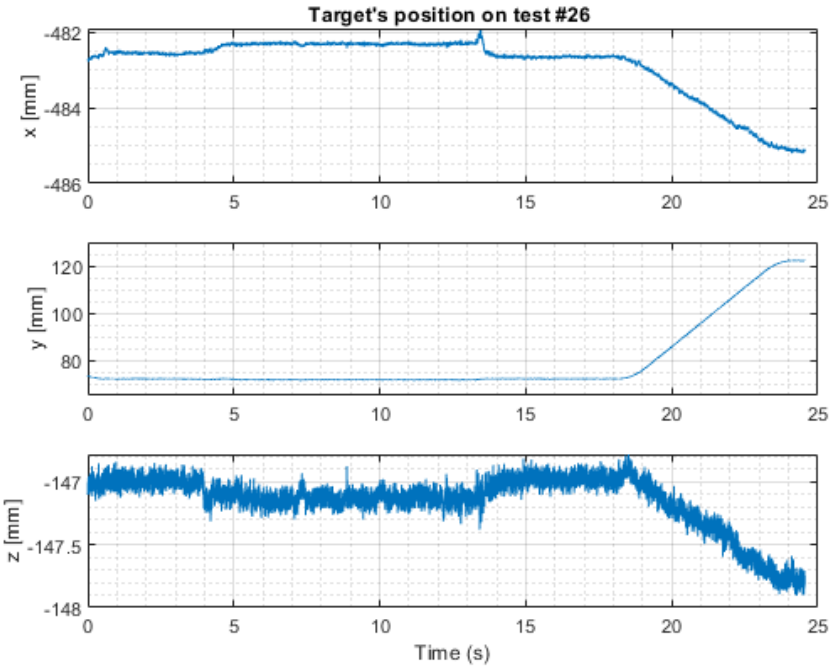


Figure 5.17 Per-axis target's position during clamping test #26, recorded by the Polaris VegaXT tracker

We can clearly see movement around $t = 18s$ when the clamping force is applied. The slippage was measured as the difference between the initial position before applying the

clamping force and the final position once the clamping force had been fully applied. Using the same example of Figure 5.17, the relative slippage of configuration 26 is calculated as follows, using Equations 4.45 and 4.46:

Let $\vec{p_1} = [-482.54, 72.01, -147.02]mm$ and $\vec{p_2} = [-482.29, 71.69, -147.09]mm$ be the before/after points, respectively taken at $t_1 = 3.3s$ and $t_2 = 13s$.

The relative slippage is the norm of the x and z components of the displacement vector:

$$\vec{\Delta p} = \vec{p_2} - \vec{p_1} = [0.254, -0.318, -0.065] \tag{5.1}$$

$$s = \sqrt{\Delta p_x^2 + \Delta p_z^2} = \sqrt{(0.254)^2 + (-0.065)^2} = 0.262 \tag{5.2}$$

$$\tag{5.3}$$

The slippage values of the most rigid robot configurations are presented in Table 5.5. The complete results, including all the tested configurations and their joint values, are presented in Table B.1 in Appendix B.

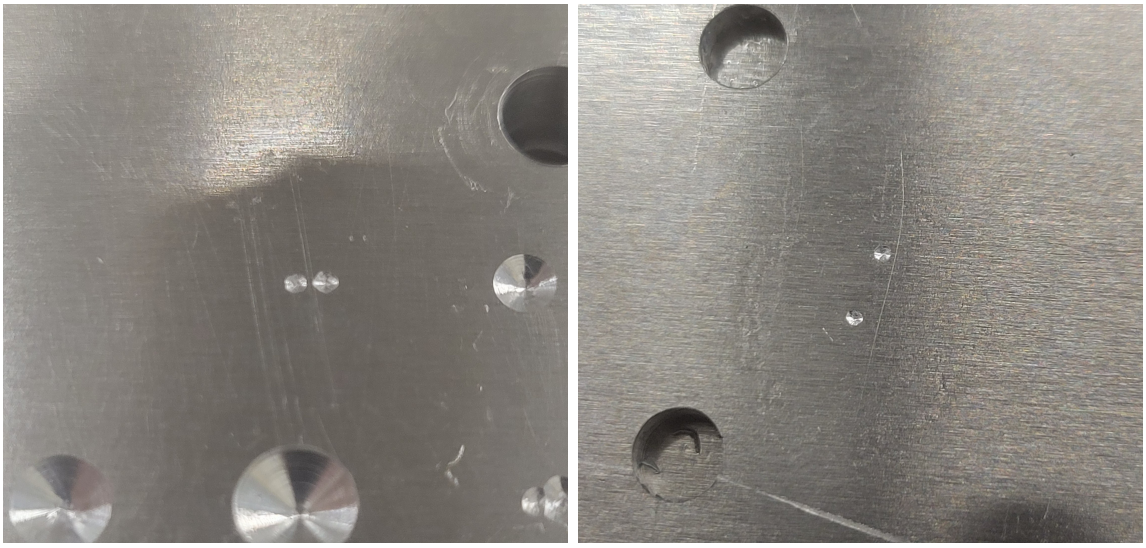
Table 5.5 Relative slippage results of the 5 best robot configurations

Config.	Elbow/Wrist orientation	Section	s [mm]
26	u/u	5	0.262
33	u/u	6	0.484
14	u/d	3	0.489
21	d/u	4	0.551
31	d/d	6	0.971

Results show that configuration 26 in Section 5 exhibits the smallest relative slippage of 0.262 mm. Following the relative slippage evaluation, the absolute slippage evaluation was conducted by comparing configuration 26 with a configuration that has not been adjusted for normal drilling or minimal slippage, using peak drilling. Figure 5.18 shows the before and after dots of both configurations.

The absolute slippage measurements for the non-optimal and the optimal configuration are respectively 1.2 mm and 3.6 mm, meaning that the robot configuration has a significant impact on the drilling accuracy of Exobrake. The slippage during clamping can be attributed to the cobot's high structural compliance, as we consider the end-effector to be rigid. Additionally, this experiment confirmed that the robot configuration impacts slippage, meaning that a characterization of this behavior can potentially lead to a general optimal configuration over the full configuration space delimited by the operational zone that minimizes slippage.

After finding the optimal robot configuration, we proceeded to the drilling tests. The first,



(a) (b)

Figure 5.18 Peak feed dots drilled for absolute slippage evaluation. (a) Optimal configuration (b) Non-Optimal configuration.

third, and fifth steps are assessments of the effect of tool wear on drilling quality, conducted using the rigid setup. Between these assessment steps, we drill holes to build the dataset on a total of 2 workpieces per tool, using Exobrake. Figure 5.19 shows an example of a workpiece after drilling a full set of holes.

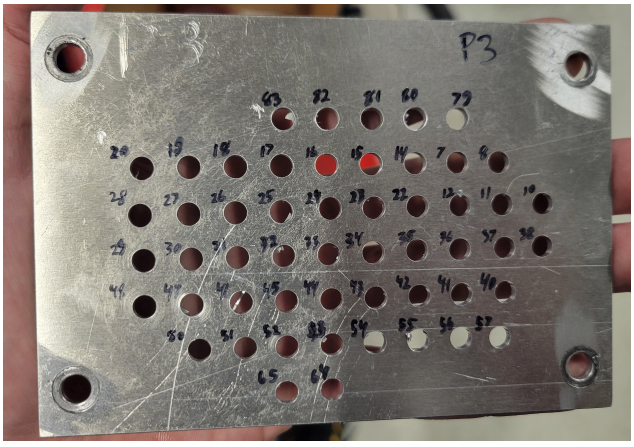
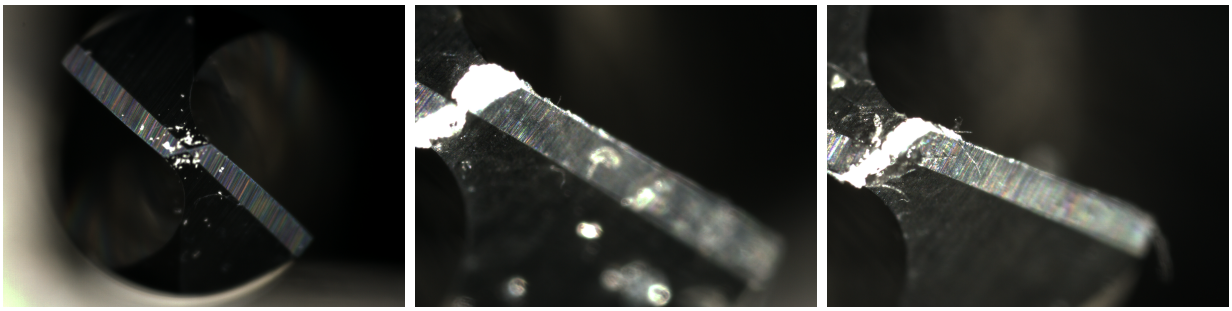


Figure 5.19 Workpiece #3 top sheet, fully drilled by Exobreak with drill bit #2. Each hole is labelled for CMM measurements

Before each drilling session on the rigid setup, the tool was placed under a microscope to measure the wear area. Figure 5.20 shows examples of microscope measurements throughout

the life of the tool.



(a) (b) (c)

Figure 5.20 Tool wear seen on a microscope and measurements. (a) Unused tool (b) after 100 holes: 18.1μ m (c) after 200 holes: 59.4μ m

The measurement method employed here is not perfect, as it is based on the user's subjective manual measurements of the tool wear area. This means that two individuals can have different measurements for the same picture. For example, taking the images from Figure 5.20b, the measurement shown in Figure 5.21 indicates how we measured tool wear.

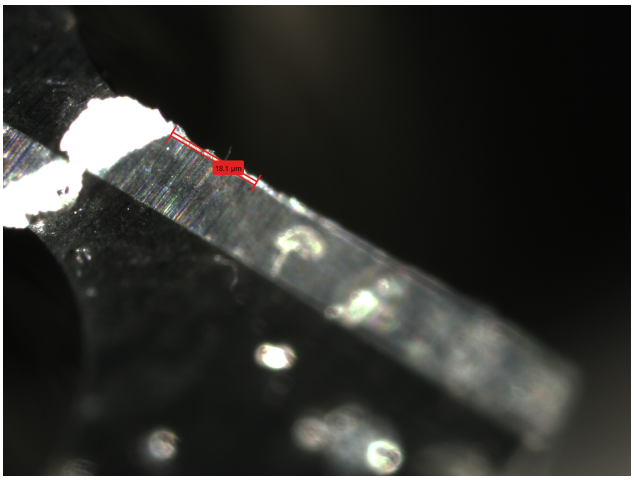


Figure 5.21 Tool wear manually measured on Figure 5.20b directly from the microscope.

5.3.2 Data Pre-processing and Features extraction

Following the drilling tests, we processed the acquired data into a usable format for machine learning. The first step consists of filtering the data; we will reuse the previous example from

Figure 5.16, which displays the raw Fz force data taken on hole 38 in plate 1. Figure 5.22 isolates the axial force sensed by the dynamometer.

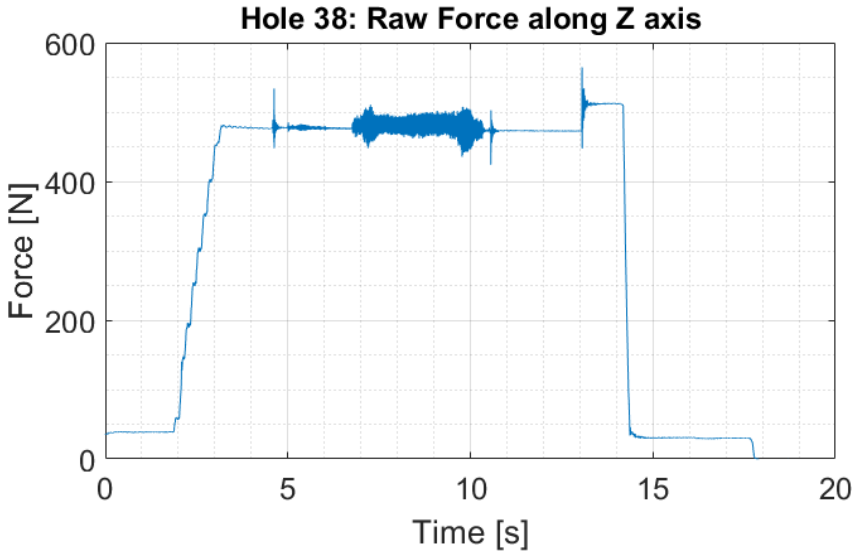


Figure 5.22 Time domain raw axial force recorded by the dynamometer during hole 38 in plate 1

From this time domain signal, we apply an FFT in Matlab to obtain its frequency domain form, as shown in Figure 5.23.

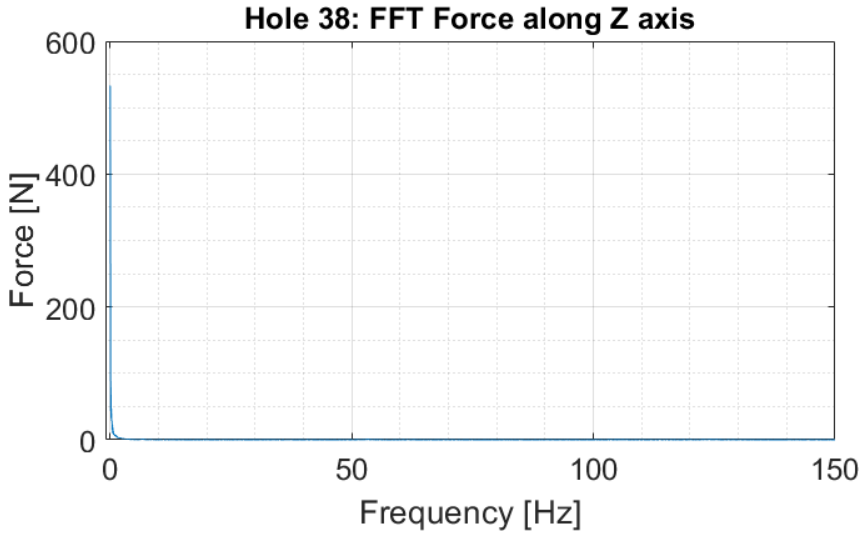


Figure 5.23 Frequency domain raw axial force recorded by the dynamometer during hole 38 in plate 1

We can notice in Figure 5.22 a first bias of 475 N during the clamping phase and a brief bias

of 520 N when removing the clamping force, which is the expected behavior of the pneumatic cylinder. These biases are represented as a 0 Hz peak of 530 N in Figure 5.23 which has a similar amplitude to the highest second plateau caused by the removal of the clamping force. In the same figure, we also notice that the 0 Hz peak decreases until 10 Hz, which is below the high-pass filter cut-off frequency of 40 Hz. We eliminate the clamping biases and high-frequency noise by applying high pass and low pass filters. The filtered time and frequency domain signals are respectively shown in Figures 5.24 and 5.25.

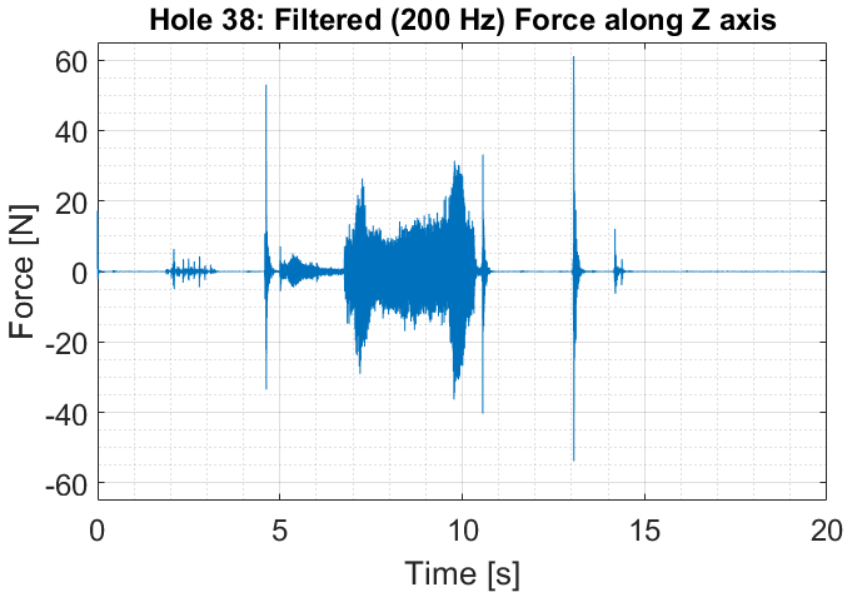


Figure 5.24 Time domain filtered axial force recorded by the dynamometer during hole 38 in plate 1

In Figure 5.24 we see that the biases caused by the clamping forces have been eliminated by the high pass filters. In Figure 5.25 we see that the 0 Hz peak caused by the clamping force has been completely filtered out and that signals above 200 Hz are being gradually reduced, which is expected from a first order low-pass filter. Similarly to F_z, the other force and vibration signals were filtered in the same manner. The example of hole 38 shows that filtering helps to process operational signals into a more usable format.

The feature set of Table 4.4 condenses drilling signals into standard time and frequency descriptors, which are widely used for hole-quality and tool-wear prediction [22–26, 28–30]. Compared to studies using only force and torque or only vibration features [22–26, 28, 29], our multi-sensor design provides a richer basis for the subsequent feature selection and model training stages.

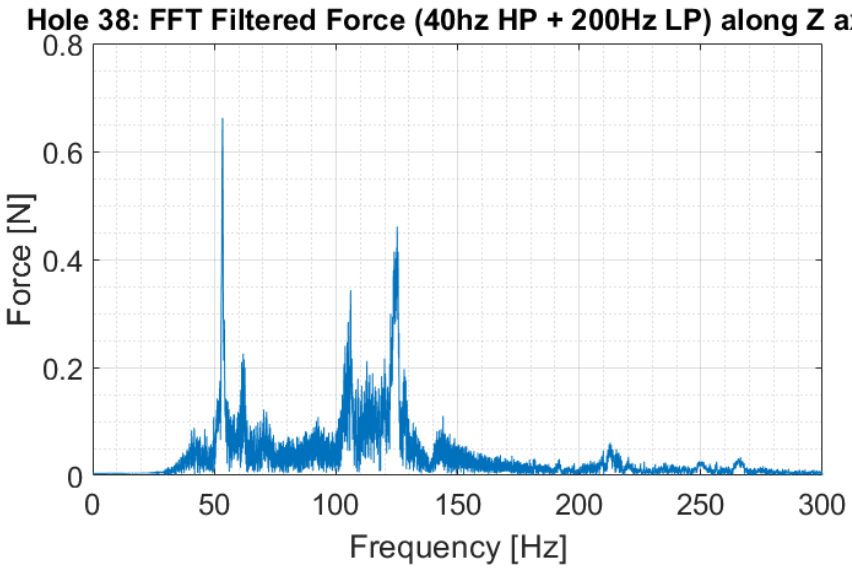


Figure 5.25 Frequency domain filtered axial force recorded by the dynamometer during hole 38 in plate 1

5.3.3 Feature Selection and Model Training

Following the methodology defined in the previous chapter, this section will present the results for both the hole quality regressor and the process parameters classifier. The results are the model's performances linked to each feature selection method for the holes quality regressor and the classification accuracy of both approaches of the process parameter classifier. This section will also present the models performances on the reduced feature set.

Hole quality regressor

The first feature selection method presented was a filter-based method based on ANCOVA analysis. ANCOVA analysis was iteratively used to identify the most significant features based on their p-value. The remaining features were then fed into other regression models using grid-search. Taking the diameter metric as as an example, the 12 most significant features are presented in Table 5.6.

These results show that the hole diameter is mostly affected by the peak amplitude of the signal components.

After keeping only the 8 most significant features, Table 5.7 presents the average performance between the initial set of 12 features and the filtered set of 8 features.

This table shows that filtering the features based on their p-value down to 8 features in-

Table 5.6 Most significant features based on p-value of the diameter metric used as initial feature set in the filter based method

Feature	p (E-05)
PeakAmp_Z	1.15E-03
PeakAmp_Fx	5.0E-03
PeakAmp_X	9.85E-03
PeakAmp_Fy	1.07E-02
PeakFreq_Y	5.32E-02
MeanFrequency_Z	1.04E-01
PeakFreq_Z	1.76E-01
PeakFreq_Fx	2.628E-01
PeakAmp_Y	3.04E-01
PowerBandwidth_X	4.0E-01
PeakAmp_Mz	4.94E-01
PeakFreq_Fz	1.56

Table 5.7 Average R^2 prediction performances when selecting the features based on their p-value

n	Train	CV	Test	Best model
12	0.99	0.49	0.42	ET
8	0.99	0.49	0.55	ET

creases the best model's performance. However, in both situations, we are confronted with overfitting, which can contribute to a drop in performance on the test set. These results show that using the most significant features based on their p-value can increase predictive performance, even when employing a non-linear model such as the Extended Trees.

The next investigated feature selection methods are wrapper methods. The best model's performance was monitored throughout each iteration of the algorithm. The first algorithm used is the feed-forward feature selection; Figures 5.26 and 5.27 show the R^2 progression of both versions, evaluated on the same random seed of run 6.

From these two figures, we notice that, despite a performance drop on the 10th added feature, version 2 keeps exploring, unlike version 1, which stops at this iteration. This results in version 2 performing better as it found a higher peak at feature #29. From this observation, we can conclude that the exploration of the feed-forward feature selection version 2 will always lead to better CV performance than that of version 1. This, however, implies a longer runtime.

Similarly, Figures 5.28 and 5.29 show the performance progression of both versions of the

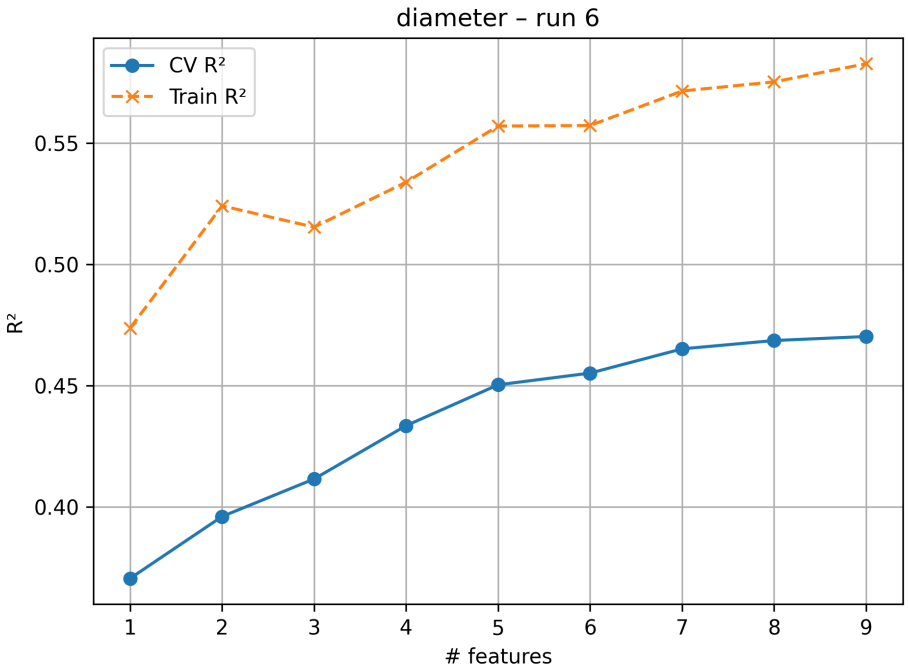


Figure 5.26 Training and CV performances progression of the version 1 forward feature selection algorithm, during run #6

backward feature selection algorithm.

From these two figures, note the same behavior as that observed for the forward feature selection algorithms. Despite a performance drop at the 54th feature, version 1 managed to find a feature set leading to higher cross-validation performance than version 2, which stopped the search at 58 features. This is expected as backward feature selection v1 and forward feature selection v2 are both greedy algorithms.

The last presented wrapper method was the JMI forward feature selection, which follows the JMI criterion of Equation 4.52 for feature selection, taking a forward approach. Keeping the example of run 6, Figure 5.30 shows the performance progression of the JMI algorithm.

JMI feature selection behaves the same way as the other greedy version of the previous methods.

Table 5.8 shows the average performance metrics of the presented wrapper methods on the full and reduced feature sets.

In general, the full feature set provides the best predictive results when compared using the same method. However, some models trained on the reduced feature set can perform better than models trained on the full feature set For example, backward v1 on the reduced feature

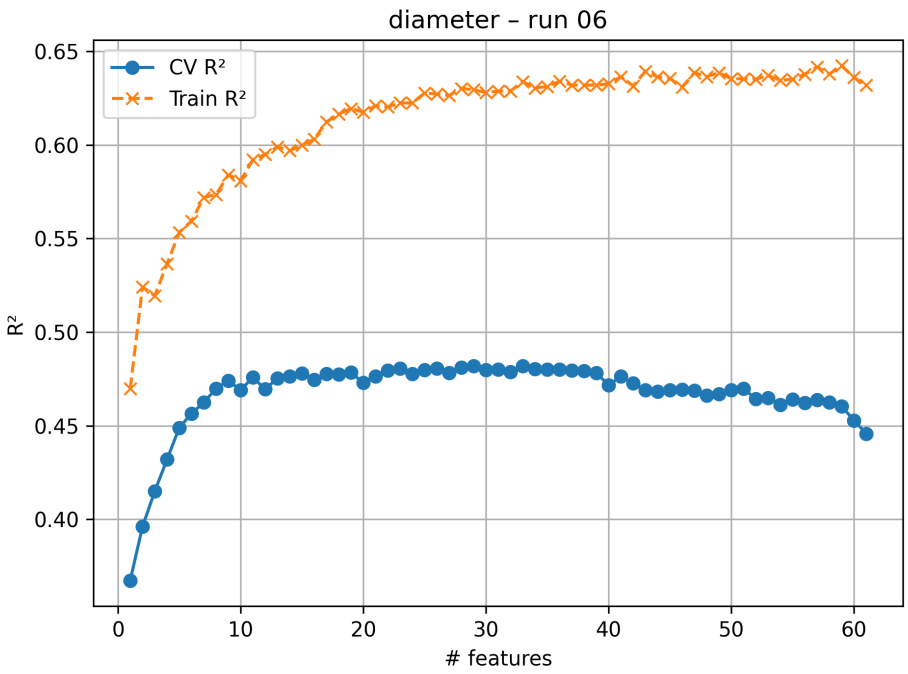


Figure 5.27 Training and CV performances progression of the version 2 forward feature selection algorithm, during run #6

Table 5.8 Average R^2 performances and final features number of the wrapper methods during feature selection, on the full and reduced features sets

	Full feature set				Reduced feature set			
	Train	CV	Test	n	Train	CV	Test	n
Forward v1	0.598	0.483	0.363	7.5	0.590	0.478	0.350	6.7
Forward v2	0.656	0.513	0.432	35.6	0.637	0.502	0.427	24.4
Backward v1	0.635	0.558	0.568	58.7	0.617	0.551	0.564	34.7
Backward v2	0.625	0.572	0.567	31.4	0.608	0.559	0.567	21.1
JMI	0.657	0.494	0.424	51.3	0.639	0.485	0.423	29.9

set performs better than forward v1 on the full feature set. We also notice that backward methods and greedy feature selection algorithms perform better in general. From the results presented in Tables 5.8 and 5.7, we conclude that the backward v1 feature selection algorithm shows the best results when using the full feature set and has been applied to the remaining hole quality metrics to be utilized in the adaptive drilling algorithm. Table 5.9 shows the regression performances of the best method on all 4 hole quality metrics.

The results presented in Table 5.9 show that the diameter metric has the best predictive performances, which is expected since the feature selection method was chosen using this

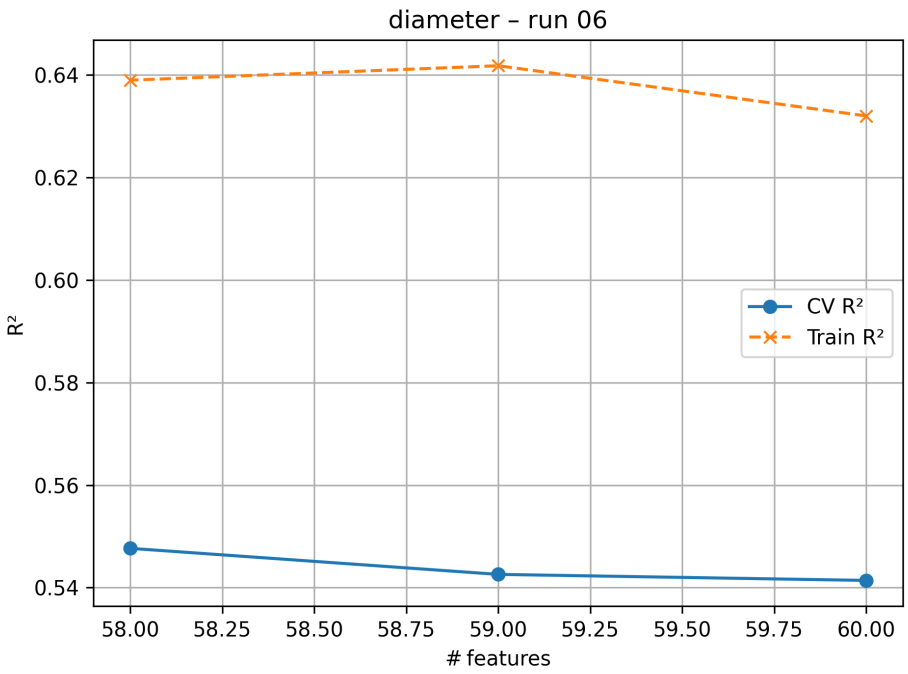


Figure 5.28 Training and CV performances progression of the version 2 backward feature selection algorithm, during run #6

Table 5.9 R^2 hole quality prediction performances all 4 metrics using the backward feature selection v1 method

	Train	CV	Test	$RMSE_{Te}$	n
Diameter	0.61	0.57	0.75	0.0079	27
Concentricity	0.74	0.74	0.62	0.0142	19
Cylindricity	0.38	0.33	0.36	0.0241	7
Runout	0.49	0.33	0.10	0.0153	16

metric. We also notice that the concentricity metric has better test performance than the average performance of the diameter metric using the same feature selection method, as seen in Table 5.8. However, the cylindricity and runout perform poorly compared to the other metrics. It is expected that the diameter yields the best regression results as it was the metric used to select the feature selection method to adopt, meaning the selected method is adapted for the diameter metric, and has been generalized the other metrics, despite possibly not being their most performing features selection method. Once integrated into the adaptive drilling algorithm, models with poor prediction performance can lead to incorrect decisions regarding the adoption of process parameters.

These regression results confirm that hole quality metrics can be predicted from force and

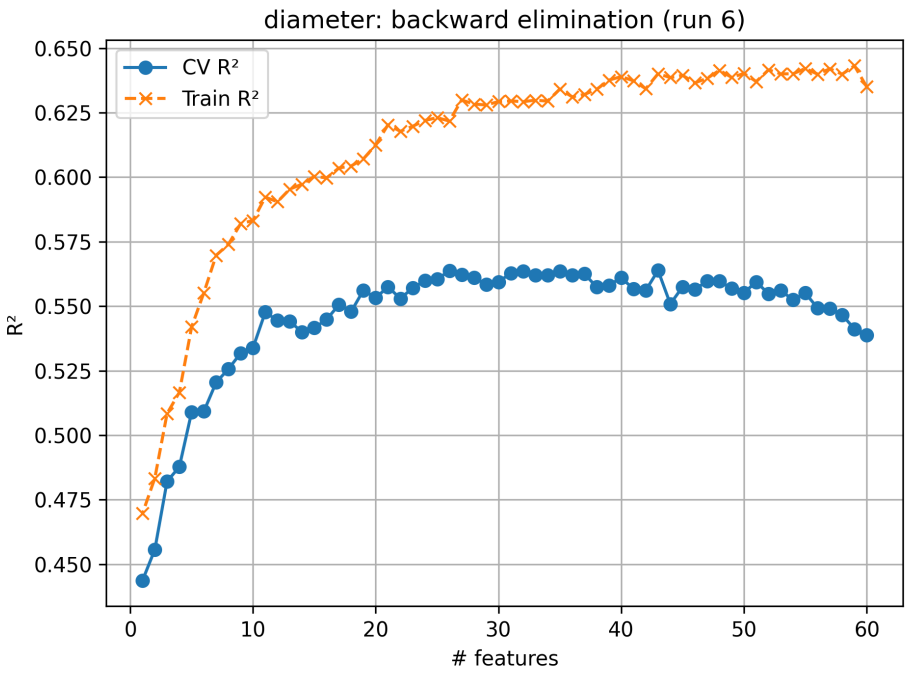


Figure 5.29 Training and CV performances progression of the version 1 backward feature selection algorithm, during run #6

vibration signals on a pose dependant cobotic system. Most drilling studies hand-picked features and target diameter, surface roughness or quality and wear classes on rigid CNC systems [22–26, 28, 30], while the feature selection approach presented automatically selects compact feature sets based on performance on a less rigid cobotic drilling system. Álvarez-Alcón et al. [24] reached an $R^2 = 0.9$ for diameter and $R^2 = 0.8$ for cylindricity, which means the regression performances of our best models are lower than those of models built from data acquired on rigid CNC machines, indicating that robot compliance can be an element preventing accurate hole quality prediction.

Process parameters classifier

The second part of this section will present the results of the process parameters classifier. The performance metric considered is classification accuracy. Grid-search and CV were used to find the best hyperparameters for each candidate classifier, and the experiment was repeated 10 times using different random seeds. Separate and joint classifications were investigated using both the full and reduced feature sets. Table 5.10 summarizes the process parameters classification results.

These results show that the classification of both process parameters separately results in

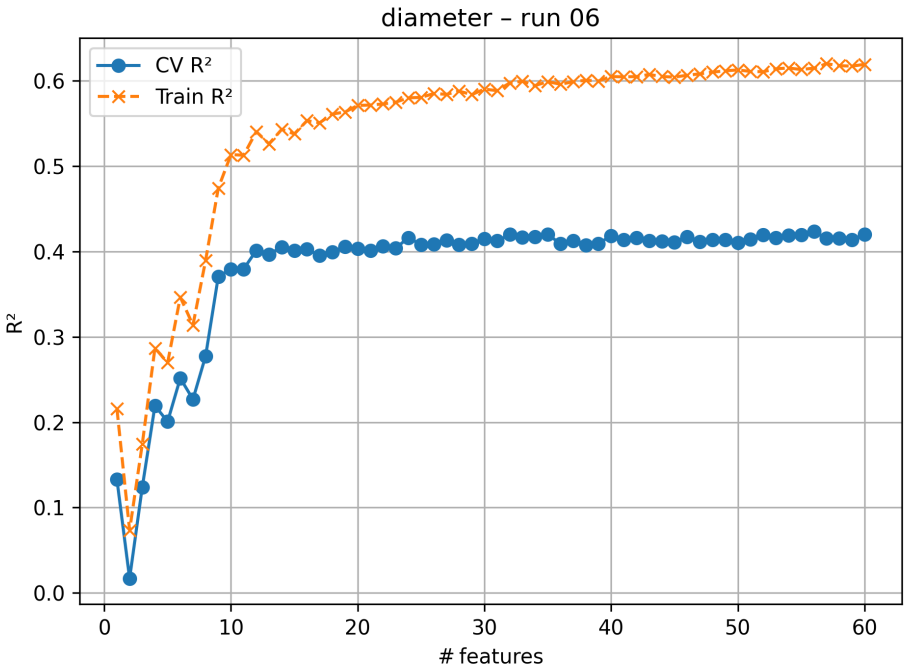


Figure 5.30 Training and CV performances progression of the version 1 JMI forward feature selection algorithm, during run #6

Table 5.10 Average accuracy results of the best classification models for the Clamping, Feed speed and Multiclass scenarios, considering both the full and reduced feature set

	Full feature set				Reduced feature set			
	CV	Train	Test	Model	CV	Train	Test	Model
Clamping	0.81	0.97	0.84	GB	0.79	0.97	0.83	GB
Feed speed	0.98	1.00	0.99	SVM	0.96	1.00	0.97	SVM
Multiclass	0.80	0.97	0.82	RF	0.78	0.97	0.80	RF

better performance compared to the multiclass approaches, with the feed speed classifier being close to perfect. Also, models trained on the reduced feature set are slightly less accurate than those trained on the full feature set. This performance drop is considered negligible, meaning that models using the reduced feature set are equivalent and can be used in a context where only the axial axis of the force is available. From these results, we deploy two separate classifiers for the clamping and feed speed process parameters, using their respective best models. At its current state, it is recommended to use the clamping classifier along with another means of anomaly detection since its average accuracy is not perfect, unlike the feed speed classifier, which can be used independently. The other solution is to improve the clamping classification model.

Most monitoring approaches treat feed speed or clamping as known inputs to predict hole quality or tool wear [22–26, 28, 30], the process parameter classifier presented learns to infer clamping and feed speed levels from measured operationnal signals. The high accuracy of the feed speed classifier and the robust performance of the clamping classifier suggest that these parameters leave a clear signature on force and vibration signals. This is consistent with other ML based studies showing how thrust, torque, and vibration signals are sensitive to feed and clamping [14–17, 22–24, 28].

5.3.4 Adaptive Drilling algorithm

This section will present the results related to the adaptive drilling algorithm. First, the specification limits taken from the rigid setup drilling test are presented in Table 5.11.

Table 5.11 Specification Limits used for the adaptive drilling system

	Spec. Max [mm]
Diameter	5.047
Concentricity	0.0749
Cylindricity	0.0541
Runout	0.0307

When comparing the RMSE values of Table 5.9, we notice that the $RMSE/Spec.Max$ ratios are the following: 0.156%, 18.95%, 44.5% and 49.8% for the diameter, concentricity, cylindricity and runout respectively. The regression error for the diameter is negligible compared to its maximum limit. But it is significant for the other metrics, meaning that the variance introduced by these models could easily go beyond the specification limit.

Next, the success table is presented in Table 5.12.

From this table, we see that a significant proportion of the holes performed by Exobrake respect the specification limits when taken individually. However, only a third of the holes respect all specification limits simultaneously. Recommendations for the most likely successful recipes can be taken from Table 5.12. For example, to maximize the probability of respecting all specification limits, a LOW clamping force and MID feed speed are recommended. These probabilities are lowered as we include the prediction intervals. Figure 5.31 shows an example of a random data point taken into the adaptive drilling algorithm, displaying the true and predicted metrics, the specification limits, and the prediction intervals.

In the example of Figure 5.31, we see that none of the specification limits are respected. The system exhibits a 100% conservative safety rate. It correctly identifies that the high uncer-

Table 5.12 Per-metric and overall success probability given the process parameters recipes

Process parameters		Success probability [%]				
Clamping	Feed speed	Diameter	Concentricity	Cylindricity	Runout	ALL
LOW	LOW	14.3	100	57.1	42.9	7.1
LOW	MID	92.9	100	71.4	57.1	57.1
LOW	HIGH	73.9	73.9	82.6	65.2	34.8
MID	LOW	62.5	100	56.2	12.5	12.5
MID	MID	82.4	100	76.5	70.6	47.1
MID	HIGH	68	100	92	80	52
HIGH	LOW	66.7	100	40	33.3	20
HIGH	MID	85	100	80	75	50
HIGH	HIGH	72	100	80	72	40
TOTAL		69.8	96.4	73.4	59.8	37.3

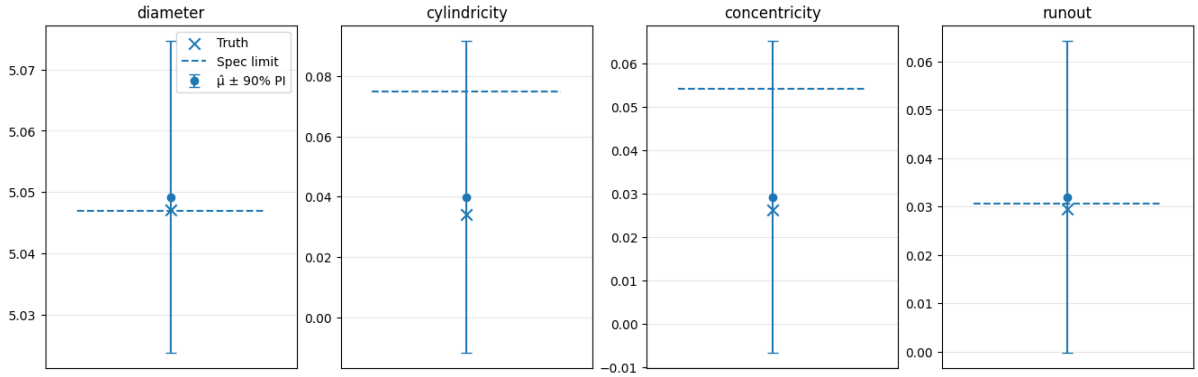


Figure 5.31 Example of a data point fed into the adaptive drilling system. We see the computed prediction interval, the predicted and true metric value and the specification limit for each hole quality metrics

tainty (wide prediction intervals) makes quality prediction impossible. This demonstrates that the system prioritizes "no change" decisions over "bad change" risks, which is the correct behavior for a high-value aerospace context.

After randomly running the adaptive drilling algorithm on 100 points, no iteration achieved a successful outcome. The cause of all these failures was the excessively high PI, which was always above the specification limits due to the poor regression performance of the hole quality prediction models, as exhibited in the example in Figure 5.31. Better performing models lead to narrower prediction intervals, which makes the adaptive drilling system more forgiving, as it is less likely that the prediction intervals will exceed the specification limits. Also, the construction of these prediction intervals assumes that the regression errors are independent, normally distributed and homoscedastic. A more robust alternative would be

to use quantile regression models to directly predict quantiles instead of the mean with Gaussian noise, as currently in use. Another option would be to recast the decision engine as a classification problem that estimates the probability that a given recipe is valid, rather than relying on normal prediction intervals around a continuous hole quality predictions.

CHAPTER 6 CONCLUSION

This work demonstrates that cobots, when equipped with a rigidity-enhancement exoskeleton and sensing capabilities, have the potential to be serious candidates for semi-autonomous aerospace drilling, reducing costs and improving flexibility compared to industrial robots. The presented framework has the potential to bring cobots closer to full integration into aerospace manufacturing by proposing foundational elements related to cobotic smart drilling end-effector design and adaptive drilling systems. Even with partial success, the prototype offers concrete value as an instrumented drilling system and a process monitoring tool, and it clearly identifies the main bottlenecks that must be addressed for a fully functional framework.

6.1 Summary of works

The work presented in this master thesis contributed towards the development of the foundational elements of a cobotic drilling framework. In addition to the rigidity increase solution already developed (Exobrake's exoskeleton), elements of a cobotic drilling framework including drilling end-effector, vision system and an adaptive drilling system design.

This project led to the design of a cobotic drilling end-effector capable of applying a clamping force of up to 600N required for aerospace drilling. It also includes a built in process parameters measurement system that senses feed-speed and clamping force. This cobotic drilling end-effector was used to drill hundreds of holes constituting the drilling dataset without any stoppages or defects.

Mounted on the end-effector, the vision system could acquire images and pointclouds of the workpiece to output relevant informations to the cobot such as pilot holes positions with an accuracy of 3.84mm and surface normal vectors locally computed on drilling site with an average error of 0.75°. Overall, despite some missing features presented in the next section, we can consider **SO1** complete.

The end-effector designed in this project was then used in a drilling test campaign to acquire force and vibration data on a total of 300 holes. In parallel, drilling tests were conducted with the same drilling unit but mounted on a CNC machine to simulate a rigid system in order to acquire benchmark data. Preprocessing of these data involved filtering and feature extraction of meaningful indicators from the measured signals. Automatic feature selection and grid search regression model training algorithms were tested to obtain the optimal feature

set leading to the maximum regression performance expressed by R^2 values of 0.75, 0.62, 0.36 and 0.1 for the diameter, concentricity, cylindricity and runout hole quality metrics respectively. Along with the hole quality prediction task, a process parameter classifier was also designed as an extra protection layer for process anomaly detection, using the force and vibration features. This classifier learned to reconstruct the feed-speed and clamping with average accuracies of 99% and 84% respectively. We conclude that **SO2** is partially complete, as the classifiers can be considered robust, as reflected by their accuracy values; however, the hole quality prediction model predicts new data poorly, as reflected by the low R^2 results, potentially because of the overgeneralized feature selection method usage over the hole quality metrics, as previously explained. However, the research that led to **SO2** successfully identifies the limitations of indirect monitoring for compliant systems, as it characterizes the gap between cobots and their rigid counterparts.

These ML models were then integrated into an adaptive drilling algorithm that models the next process parameters to adopt, given the current process parameters and the quality metric predictions of neighboring process parameters stored in a lookup table.

Using the results in the success Table 5.12 and the conclusion set on **SO2**, we can finally answer the research question defined in Section 3. Results indicate that hardware rigidity remains the primary bottleneck, as the adaptive software layer could not fully compensate for the physical compliance limitation identified in Chapter 5. As concluded for **SO2**, the monitoring models developed can only monitor process parameters but not the hole quality metrics.

Finally, we can conclude that the main objective was partially achieved, providing strong foundations toward a complete cobotic drilling framework.

6.2 Limitations

Exobrake, the cobotic drilling system based on a UR10e collaborative robot, highlights the potential use of collaborative robots for multi-stack aerospace drilling. However, this system is limited in some aspects.

First, the drilling end-effector is subject to contradictory functionalities. As described in its dedicated methodology section, the drilling unit's rotation speed measurement is interfered by the MQL system as both functionalities cannot coexist because the lubricant sprayed on the tool removes the reflector and thus making rotation speed impossible to acquire with the tested solution. To fix this problem, we can acquire tool rotation speed through modal analysis by using a machining accelerometer and online FFT algorithms, as previously

recommended. Modifying the end-effector design in such a way to spray lubricant to not interfere with laser readings is also an alternative. Also, the drilling unit was not operating at full capacities as the air pressure available in the CNRC's robotics lab was not enough to drill at full power. The feed speed being directly linked with the drilling unit's air pressure, we could have explored higher values of feed speed and even add another discretization level if we were not limited in this aspect. The tool air pressure also impacts the drilling unit's maximum rotation speed, meaning that higher input pressure can run the tool at a higher rotation speed, which could have led to different outcomes than those presented. This limitation can be addressed by simply increasing the available air pressure in the lab. Unlike the clamping force, which was controlled by an analog signal sent to a proportionnal valve from the robot controller, feed-speed had to be controlled by manually adjusting the hydraulic damper acting as physical resistance to the self-fed drilling unit's motion. The limitation can be fixed by replacing the manual damper by a MR damper whose resistance is controlled by an analog signal.

Second, the vision system can sometimes be inconsistent, as the data captured are highly affected by the surrounding lighting conditions, sometimes causing over exposition of the workpiece. Some fine tuning was required, such as for the *HoughCircles* parameters to fit the right holes in the 2D image. In a test bench with different lighting conditions, these parameters must be tuned once again. And despite being designed with the intention of being transposable on curved surfaces, the vision system was only validated on a flat surface and would require additional developments and tests to be used on a curved surface.

As for the DOE used to build the drilling dataset, only a single robot pose was used during the tests to reduce the time spent to data collection. Because a collaborative robot is a pose-dependant system, including multiple robot configurations unlocks another process parameter in the adaptive drilling system. This limitation can be addressed by collecting drilling data using different robot configurations and re-training the hole quality prediction models. In the same subject, cockpits are made of stacks including CFRP, aluminum and titanium; work done in this thesis only considered aluminum sheets. Stack composition is also a process parameter impacting the hole quality and limiting the work on aluminum stacks exclusively is not fully representative of industrial scenarios. This limitation can be addressed by applying the same methodology presented, but on stacks of different compositions. Not being able to control and measure the rotation speed, the presented methodology did not include this process parameter, despite its crucial importance in drilling. Acquiring control over the rotation speed is considered future work. Once available, it can be included in the DOE following the same methodology as clamping and feed-speed.

6.3 Future research

In addition to the recommendations regarding the limitations presented in the previous section, this project has opened the door to multiple possibilities for future research. Robot modal analysis for manufacturing was not covered in this project, being able to predict Exobrake's FRF signature from its joint values, as in the research conducted by Chen et al. [20, 20], could lead to improvements in hole quality by finding robot configurations with minimal vibrations.

In the vision system, CNN and deep learning can be used to train a model to detect holes from example images taken under different lighting conditions, instead of using procedural algorithms such as Hough Circles that are sensitive to external conditions and require parameters fine tunning. Following the developments made on the vision system, the algorithm to generate the hole pattern and guide the robots towards it still needs to be implemented; this can either be done in ROS or by interfacing the camera with the robot by using TCP/IP communication protocol, as described in Section 4.2.

Also, non ML methods could be investigated for hole quality predictions, as presented in this project's literature review. More advanced machine learning models, such as deep learning models, can be investigated. Work on more robust feature selection methods can be developed, for example, by considering each hole quality metrics to find their best feature selection method instead of just considering the diameter, as done here. Because **SO2** was partially achieved due to poor regression performance, future work should first address model limitations before attempting more advanced adaptive drilling strategies

REFERENCES

[1] "A Kuka AG robotic arm, fitted with an Alema Automation SAS unit,..." Oct. 2019. [Online]. Available: https://www.gettyimages.co.uk/detail/news-photo/kuka-ag-robotic-arm-fitted-with-an-alema-automation-sas-news-photo/1172719564

[2] L. Rice, "Project Highlight: Flexible Drilling System (FDS)," Aug. 2022. [Online]. Available: https://arminstitute.org/news/project-highlight-flexible-drilling-system-fds/

[3] L. Muijs and M. Snijders, "Collaborative Robot Applications at GKN Aerospace's Fokker Business," Sep. 2017, pp. 2017–01–2091. [Online]. Available: https://www.sae.org/content/2017-01-2091/

[4] "The cobotic OUTPORT project," May 2025. [Online]. Available: https://www.safran-group.com/videos/cobotic-outport-project

[5] C. Tsao and H. Hocheng, "Effects of exit back-up on delamination in drilling composite materials using a saw drill and a core drill," *International Journal of Machine Tools and Manufacture*, vol. 45, no. 11, pp. 1261–1270, Sep. 2005. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0890695505000192

[6] "ur10e_e-series_datasheets_web." [Online]. Available: https://www.universal-robots.com/media/1807466/ur10e_e-series_datasheets_web.pdf

[7] S. Laporte, E. Gueydon, A. Auffret, and C. De Castelbajac, "A New Robotic Vibration-Drilling Process," *SAE International Journal of Materials and Manufacturing*, vol. 8, no. 1, pp. 56–62, Sep. 2014. [Online]. Available: https://www.sae.org/content/2014-01-2258/

[8] Y. Zhendi, Z. Yuanyuan, L. Ying, Z. Weixuan, Y. Xueqin, C. Qunlin, and X. Aijie, "Automatic drilling and riveting actuators and methods applicable to rocket hull riveting," WO Patent WO2 021 036 588A1, Mar., 2021. [Online]. Available: https://patents.google.com/patent/WO2021036588A1/en?oq=WO2021036588A1

[9] L. Qian, P. Liu, H. Lu, J. Shi, and X. Zhao, "An End-to-End Inclination State Monitoring Method for Collaborative Robotic Drilling Based on Resnet Neural Network," *Sensors*, vol. 24, no. 4, p. 1095, Jan. 2024, number: 4 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/24/4/1095

[10] N. Doiron and A. Gallant, "Increasing the Payload of a 7DOF Cobot," Apr. 2023, pp. 101–110.

[11] V. Arakelian, "A COBOTIC AND FLEXIBLE SOLUTION FOR HANDLING OF HEAVY LOADS," 2022.

[12] G. Van de Perre, T. Hubert, T. Verstraten, and B. Vanderborght, "Investigating the potential of flexible links for increased payload to mass ratios for collaborative robotics," *IEEE Access*, vol. 11, pp. 15 981–15 995, Feb. 2023. [Online]. Available: http://www.scopus.com/inward/record.url?scp=85149178218&partnerID=8YFLogxK

[13] H. Jin, M. Luo, S. Lu, Q. He, and Y. Lin, "Design and Analysis of a Novel Variable Stiffness Joint for Robot," *Actuators*, vol. 12, no. 1, p. 10, Jan. 2023, number: 1 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2076-0825/12/1/10

[14] B. Luo, K. Zhang, S. Liu, H. Cheng, and R. Wang, "Investigation on the interface damage in drilling low-stiffness CFRP/Ti stacks," *Chinese Journal of Aeronautics*, vol. 32, no. 9, pp. 2211–2221, Sep. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1000936119302006

[15] M. Aamir, "Simultaneous machining of aerospace structural materials using poly-drills."

[16] F. Bañon, A. Sambruno, M. Batista, S. R. Fernandez-Vidal, and J. Salguero, "Study of the one-shot drilling of CFRP/Ti6Al4V stacks with a double tip angle cutting-tool geometry," Vitoria-Gasteiz, Spain, 2019, p. 080010. [Online]. Available: https://pubs.aip.org/aip/acp/article/889612

[17] C. Lei, C. Li, Y. Bi, and J. Li, "The optimal clamping force option for robotic drilling of stacked aluminum sheets based on shell theory," *Advances in Mechanical Engineering*, vol. 9, no. 3, p. 168781401769504, Mar. 2017. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1687814017695048

[18] A. Pardo, "Fundamental investigation of the drilling of multi- material aerospace stacks to aid adaptive drilling."

[19] C. Chen, F. Peng, R. Yan, X. Tang, Y. Li, and Z. Fan, "Rapid prediction of posture-dependent FRF of the tool tip in robotic milling," *Robotics and Computer-Integrated Manufacturing*, vol. 64, p. 101906, Aug. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0736584519302443

[20] H. Chen and K. Ahmadi, "Estimating pose-dependent FRF in machining robots using multibody dynamics and Gaussian Process Regression," *Robotics and Computer-Integrated Manufacturing*, vol. 77, p. 102354, Oct. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0736584522000424

[21] J. R. Flachs, M. Salahshoor, and S. N. Melkote, "Mechanistic models of thrust force and torque in step-drilling of Al7075-T651," *Production Engineering*, vol. 8, no. 3, pp. 319–333, jun 2014. [Online]. Available: http://link.springer.com/10.1007/s11740-014-0531-5

[22] J. P. Kumar and P. Packiaraj, "EFFECT OF DRILLING PARAMETERS ON SURFACE ROUGHNESS, TOOL WEAR, MATERIAL REMOVAL RATE AND HOLE DIAMETER ERROR IN DRILLING OF OHNS."

[23] S. S. Babu and C. Dhanasekaran, "Comparative analysis of thrust force, roughness and roundness error in drilling of aluminium composites using RSM, ANN and fuzzy logic," *Materials Today: Proceedings*, vol. 69, pp. 908–917, 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2214785322050076

[24] M. Álvarez Alcón, L. N. López De Lacalle, and F. Fernández-Zacarías, "Multiple Sensor Monitoring of CFRP Drilling to Define Cutting Parameters Sensitivity on Surface Roughness, Cylindricity and Diameter," *Materials*, vol. 13, no. 12, p. 2796, Jun. 2020. [Online]. Available: https://www.mdpi.com/1996-1944/13/12/2796

[25] Y. Shaban, S. Yacout, M. Balazinski, M. Meshreki, and H. Attia, "Diagnosis of machining outcomes based on machine learning with Logical Analysis of Data," in *2015 International Conference on Industrial Engineering and Operations Management (IEOM)*. Dubai: IEEE, Mar. 2015, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/document/7093752/

[26] C. Domínguez-Monferrer, J. Fernández-Pérez, R. De Santos, M. Miguélez, and J. Cantero, "Machine learning approach in non-intrusive monitoring of tool wear evolution in massive CFRP automatic drilling processes in the aircraft industry," *Journal of Manufacturing Systems*, vol. 65, pp. 622–639, Oct. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0278612522001893

[27] B. Denkena, B. Bergmann, and M. Witt, "Material identification based on machine-learning algorithms for hybrid workpieces during cylindrical operations," *Journal of Intelligent Manufacturing*, vol. 30, no. 6, pp. 2449–2456, Aug. 2019. [Online]. Available: http://link.springer.com/10.1007/s10845-018-1404-0

[28] J. Susai Mary, D. Dinakaran, M. A. Sai Balaji, S. Satishkumar, and A. S. Arockia Doss, "Monitoring and Modeling of Cylindricity Error Using Vibration Signals in Drilling," in *Handbook of Smart Materials, Technologies, and Devices*, C. M. Hussain and P. Di Sia, Eds. Cham: Springer International Publishing, 2022, pp. 117–125. [Online]. Available: https://link.springer.com/10.1007/978-3-030-84205-5_149

[29] D. A. Dornfeld and M. DeVries, "Neural Network Sensor Fusion for Tool Condition Monitoring," *CIRP Annals*, vol. 39, no. 1, pp. 101–105, 1990. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0007850607610129

[30] S. K. H. Lee, P. G. Mongan, A. Farhadi, E. P. Hinchy, N. P. O'Dowd, and C. T. McCarthy, "In-situ evaluation of hole quality and cutting tool condition in robotic drilling of composite materials using machine learning," *Journal of Intelligent Manufacturing*, Jan. 2025, publisher: Springer Science and Business Media LLC. [Online]. Available: https://link.springer.com/10.1007/s10845-024-02528-7

[31] "OpenCV: Camera Calibration and 3D Reconstruction." [Online]. Available: https://docs.opencv.org/4.x/d9/d0c/group___calib3d.html

[32] M. Bennasar, Y. Hicks, and R. Setchi, "Feature selection using Joint Mutual Information Maximisation," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8520–8532, Dec. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417415004674

APPENDIX A DRILLING STRUCTURE FEA

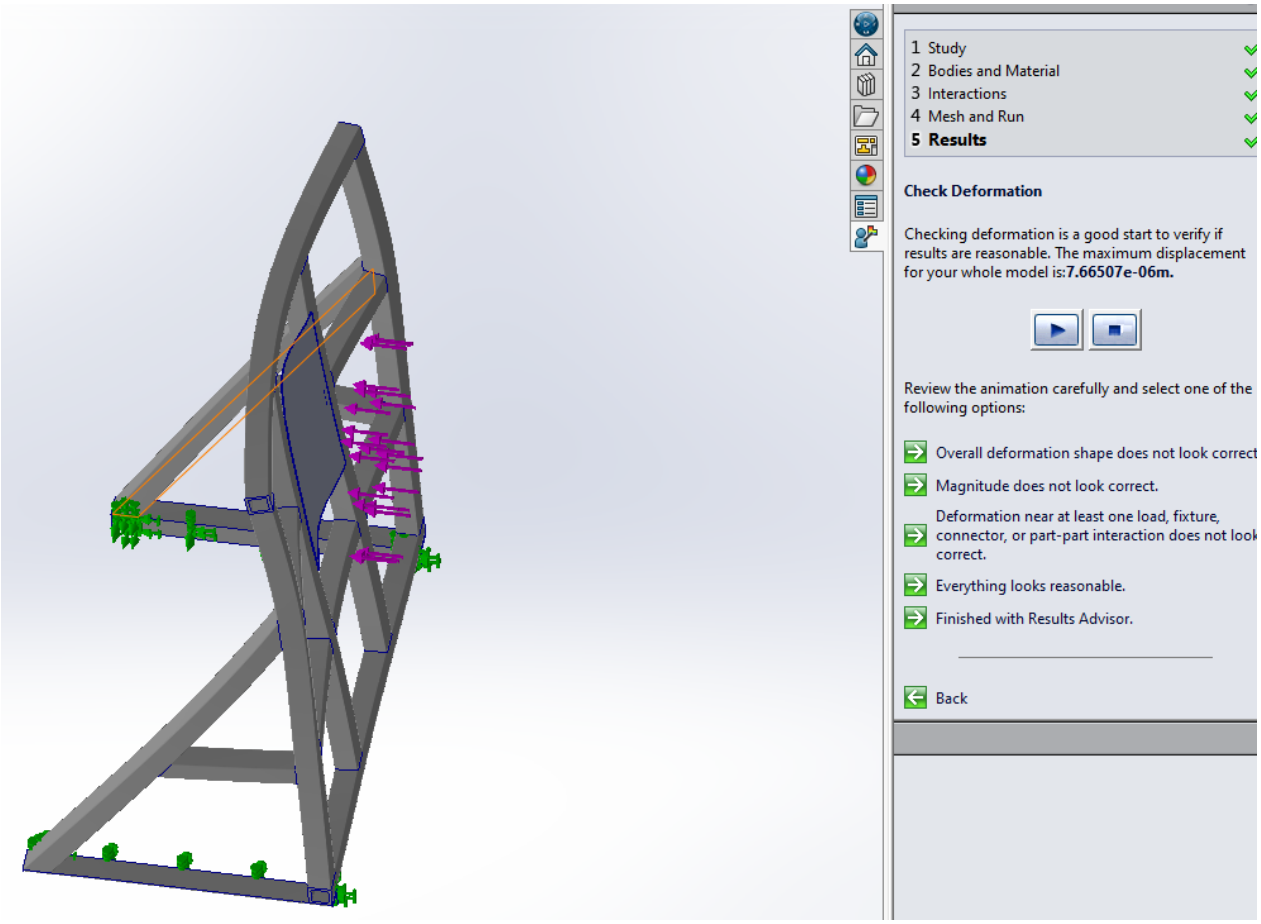


Figure A.1 FEA analysis of the drilling structure with 600N of applied force

APPENDIX B DRILLING TESTS

Table B.1 Complete slippage results with robot configurations

Config. #	Section	Elbow/Wrist orient.	Robot joints value [°]						s [mm]
			q_0	q_1	q_2	q_3	q_4	q_5	
1	1	d/d	-8.4	-159.9	71.2	-49.0	-43.9	-142.6	2.785
2	1	u/d	351.5	269.3	287.4	22.9	316.3	217.7	2.901
3	1	d/d	-11.0	-150.5	85.1	-156.4	-42.5	-125.6	1.422
4	1	u/d	-11.1	-68.9	-84.3	-71.5	-42.8	-125.0	2.270
5	2	d/u	-15.3	-132.5	43.9	32.7	-53.7	-152.2	3.318
6	2	u/u	344.0	274.7	300.9	67.7	311.2	206.1	3.169
7	2	u/d	341.4	283.5	280.0	45.6	320.0	206.7	3.050
8	2	d/u	344.8	229.9	40.4	38.4	305.2	208.3	2.993
9	2	u/u	344.3	273.1	305.8	71.5	309.5	206.5	2.637
10	3	u/d	28.9	235.3	82.2	290.8	304.1	259.7	2.760
11	3	d/d	28.9	314.0	277.8	16.5	304.1	259.7	2.601
12	3	u/d	21.6	235.2	96.9	240.1	292.0	239.5	2.824
13	3	d/d	21.6	327.6	263.1	341.5	292.0	239.5	3.223
14	3	u/d	13.7	255.0	82.9	171.7	297.4	216.2	0.490
15	3	d/d	13.7	334.3	277.1	258.1	297.4	216.2	1.261
16	3	d/d	29.8	239.8	70.6	315.2	314.6	263.6	3.117
17	3	u/d	29.8	307.5	289.4	28.7	314.6	263.6	2.316
18	4	u/d	351.8	268.8	263.2	55.7	315.5	216.7	2.206
19	4	d/d	-11.2	190.9	113.8	-175.0	-43.4	-123.9	1.113
20	4	u/d	349.5	292.7	243.5	-23.3	318.6	231.2	2.590
21	4	d/u	-10.8	-149.7	87.3	129.9	-55.0	-123.0	0.550
22	4	u/u	349.2	293.8	-87.3	221.0	305.0	-123.0	1.549
23	5	d/u	344.5	195.5	94.6	-7.4	307.9	204.2	2.183
24	5	u/d	340.4	292.6	245.9	59.7	322.3	202.9	2.254
25	5	d/u	342.9	217.1	73.7	53.0	292.8	217.4	0.825
26	5	u/u	339.1	293.0	286.9	148.8	288.1	227.5	0.262
27	6	d/u	29.8	224.3	106.7	294.6	314.6	263.6	2.510
28	6	u/u	28.4	332.4	243.0	29.4	302.1	258.1	1.885
29	6	d/d	20.8	229.0	126.8	212.5	291.8	237.3	2.509
30	6	u/d	20.8	347.8	233.2	347.3	291.8	237.3	2.849
31	6	d/d	13.6	251.5	111.3	144.5	297.7	215.7	0.671
32	6	u/d	13.6	356.9	248.7	261.6	297.7	215.7	1.402
33	6	u/u	13.9	336.0	279.7	183.3	312.5	206.8	0.484
34	6	d/u	22.8	244.2	76.3	24.2	331.3	231.0	1.178

APPENDIX C WRAPPER-BASED FEATURE SELECTION ALGORITHM PSEUDOCODES

Algorithm 1 Evaluate($D, y, S; \mathcal{M}, \text{opts}$)

Require: Subset S, model family $\mathcal{M}$, and evaluation options:

- preprocessing (scaling for numeric, one-hot for categoricals),
- optional PCA components $\alpha \in \{\text{off}\} \cup \{\alpha_1, \dots\}$,
- cross-validation scheme (e.g., KFold),
- scoring metric (e.g., CV R^2).

Build feature matrix $X \leftarrow D[S]$.
For each model $m \in \mathcal{M}$ and each PCA setting α:
  Fit m using CV; record the mean validation score.
return the best CV score among all tested pipelines.

Algorithm 2 Forward Feature Selection v1 (tolerance, early stop)

Input: $F, D, y, \mathcal{M}, \text{opts}$ with ϵ
$S \leftarrow \emptyset,\ R \leftarrow F,\ S^\star \leftarrow \emptyset,\ s^\star \leftarrow -\infty$
while $R \neq \emptyset$ **do**
    $s_{\text{base}} \leftarrow \text{Evaluate}(D, y, S; \mathcal{M}, \text{opts})$
    if $s_{\text{base}} > s^\star$ **then**
        $s^\star \leftarrow s_{\text{base}},\ S^\star \leftarrow S$
    end if
    $f^* \leftarrow \varnothing,\ s_{\text{best}} \leftarrow s_{\text{base}}$
    for each $f \in R$ **do**
        $s \leftarrow \text{Evaluate}(D, y, S \cup \{f\}; \mathcal{M}, \text{opts})$
        if $s > s_{\text{best}}$ **then**
            $s_{\text{best}} \leftarrow s,\ f^* \leftarrow f$
        end if
    end for
    if $f^* = \varnothing$ **or** $s_{\text{best}} \leq s_{\text{base}} + \epsilon$ **then**
        break
    end if
    $S \leftarrow S \cup \{f^*\},\ R \leftarrow R \setminus \{f^*\}$
end while
Return $S^\star$

Algorithm 3 Forward Feature Selection v2 (always add one, best-so-far)

Input: $F, D, y, \mathcal{M}, \text{opts}$
$S \leftarrow \emptyset,\ R \leftarrow F,\ S^\star \leftarrow \emptyset,\ s^\star \leftarrow -\infty$
while $R \neq \emptyset$ **do**
    $f^* \leftarrow \varnothing,\ s_{\text{step}} \leftarrow -\infty$
    for each $f \in R$ **do**
        $s \leftarrow \text{Evaluate}(D, y, S \cup \{f\}; \mathcal{M}, \text{opts})$
        if $s > s_{\text{step}}$ **then**
            $s_{\text{step}} \leftarrow s,\ f^* \leftarrow f$
        end if
    end for
    $S \leftarrow S \cup \{f^*\},\ R \leftarrow R \setminus \{f^*\}$
    if $s_{\text{step}} > s^\star$ **then**
        $s^\star \leftarrow s_{\text{step}},\ S^\star \leftarrow S$
    end if
end while
Return $S^\star$

Algorithm 4 Backward Feature Selection v1 (always drop one, best-so-far)

Input: $F, D, y, \mathcal{M}, \text{opts}$
$S \leftarrow F,\ S^\star \leftarrow S,\ s^\star \leftarrow \text{Evaluate}(D, y, S; \mathcal{M}, \text{opts})$
while $|S| > 1$ **do**
    $f_{\text{drop}} \leftarrow \varnothing,\ s_{\text{step}} \leftarrow -\infty$
    for each $f \in S$ **do**
        $S' \leftarrow S \setminus \{f\}$
        $s \leftarrow \text{Evaluate}(D, y, S'; \mathcal{M}, \text{opts})$
        if $s > s_{\text{step}}$ **then**
            $s_{\text{step}} \leftarrow s,\ f_{\text{drop}} \leftarrow f$
        end if
    end for
    $S \leftarrow S \setminus \{f_{\text{drop}}\}$
    if $s_{\text{step}} > s^\star$ **then**
        $s^\star \leftarrow s_{\text{step}},\ S^\star \leftarrow S$
    end if
end while
Return $S^\star$

Algorithm 5 Backward Feature Selection v2 (tolerance, early stop)

Input: $F, D, y, \mathcal{M}$, opts with ϵ
$S \leftarrow F$
while True **do**
    $s_{\text{base}} \leftarrow \text{Evaluate}(D, y, S; \mathcal{M}, \text{opts})$
    $f_{\text{drop}} \leftarrow \varnothing,\ s_{\text{best}} \leftarrow s_{\text{base}}$
    for each $f \in S$ **do**
        $S' \leftarrow S \setminus \{f\}$
        $s \leftarrow \text{Evaluate}(D, y, S'; \mathcal{M}, \text{opts})$
        if $s > s_{\text{best}}$ **then**
            $s_{\text{best}} \leftarrow s,\ f_{\text{drop}} \leftarrow f$
        end if
    end for
    if $s_{\text{best}} > s_{\text{base}} + \epsilon$ **then**
        $S \leftarrow S \setminus \{f_{\text{drop}}\}$
    else
        break
    end if
end while
Return S

Algorithm 6 JMI Forward Feature Selection

Input: Feature set F, target y, number of features k
$S \leftarrow \emptyset$
Compute univariate MI: $I(f; y)$ for all $f \in F$.
Optionally compute pairwise redundancy: $I(f; g)$ for all $f, g \in F$.
for $i = 1$ to k **do**
    for each $f \in F \setminus S$ **do**
        if $S = \emptyset$ **then**
            $JMI(f) \leftarrow I(f; y)$
        else
            $JMI(f) \leftarrow I(f; y) - \frac{1}{|S|} \sum_{g \in S} I(f; g)$
        end if
    end for
    $f^* \leftarrow \arg\max_{f \in F \setminus S} JMI(f)$
    $S \leftarrow S \cup \{f^*\}$
end for
Return S